One-Time Anonymous Certificateless Signcryption Scheme Based on Blockchain

Yan Jin[1,2], Chunxiao Ye[1,2(✉)], Mengqing Yang[1,2], and Chunming Ye[1,2]

[1] College of Computer Science, Chongqing University, Chongqing, China
{jiny,yecx}@cqu.edu.cn
[2] Key Laboratory of CPS-DSC, MoE, Chongqing University, Chongqing, China
yangmengqing@pku.edu.cn

Abstract. The rapid increase of users in the blockchain makes it possible to exchange a large amount of data every moment. Since the information published on blockchain is recorded and cannot be tampered with, there is the problem of leaking the real identity of users under the big data clustering attack. Meanwhile, the existing key generation center (KGC) needs a secure channel to transmit partial private keys, which makes partial private keys depending on the channel of interaction and creates a private key security problem. In this paper, we propose a one-time anonymous certificateless signcryption (OTACLSC) scheme based on blockchain. We securely use the public channel to improve the security of the private key. By constructing a one-time pseudonym public key in the blockchain to achieve anti-identity leakage, the communication initiator constructs the pseudonym public key of both communication parties to avoid the reuse of the pseudonym public key. Then we prove the security of the scheme under the random oracle model and compare it with other schemes. Our scheme has less computation cost and shorter ciphertext length while maintaining more reliable security.

Keywords: Blockchain · Big data clustering attack · Certificateless signcryption · One-time pseudonym · Random oracle model

1 Introduction

As an emerging technology, blockchain has attracted extensive attention in the industry due to its decentralization, anonymity, tamper-proof, and other characteristics [25]. To ensure the security of communication, the communicating parties need to authenticate each other's identity and often need to go to verify the user's certificate through public key infrastructure (PKI). In order to simplify the certificate management process, Al-Riyami and Paterson [1] proposed the concept of certificateless public key cryptography on the basis of the identity-based encryption system (IBE) [20]. Based on the private key generator (PKG), the certificateless public key cryptosystem retains the process of incorporating user ID into the generation of public-private key pairs and uses a trusted third-party key generation center (KGC).

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 460, pp. 533–551, 2022.
https://doi.org/10.1007/978-3-031-24383-7_29

Using the certificateless algorithm, the generation of the public key requires a partial private key, and most certificateless schemes use a centralized KGC to generate the partial private key and send it to the user through a secure channel. There is no doubt that centralized KGC is convenient and available, but it cannot avoid the possibility of some common security flaws, such as man-in-the-middle attacks, and KGC compromised attacks. It is important to note that the use of secure channels makes the privacy of partial private keys dependent on secure channels. Once a secure channel is controlled by an attacker, the user's partial private key may be compromised, which is a serious security problem for both the cryptographic system and the communication system. In addition, maintaining a secure channel increases the complexity of communication systems and requires additional costs.

At the same time, blockchain technology, as a trust solution in the digital era, has a certain contradiction between its decentralized transparency and the reality of the required privacy. Posting transaction information in blockchain requires the use of user accounts, and data such as transactions are traceable and cannot be tampered with, which means vulnerability to big data clustering attacks. When we interact in the real world and the virtual world, the public key will be recorded and saved all the time, and the user's transaction records will be collected in large quantities, inevitably leaving traces that can be traced from transactions and other information to the real identity information [24]. The missing privacy protection is unacceptable in the real world, because not only individuals want to protect the privacy of their property information and other private information, but also any business or organization wants to keep its sensitive and valuable data confidential.

1.1 Related Work

Li et al. [13] built the certificateless scheme on the Internet of Things and divided it into the online/offline signcryption scheme. To reduce the tedious operations of online signcryption, the scheme eliminated all the tedious operations in the online stage and transferred them to the offline stage. To realize identity authentication on the chain, Fromknecht et al. [10] proposed the combination of PKI and blockchain. Under the characteristics of blockchain, Fromknecht et al. build a distributed PKI, which is responsible for managing the certificates disclosed by users on the chain. Based on the research of Fromknecht et al., Axon [4] proposed PKI privacy protection based on blockchain to reduce the risk of user privacy disclosure. The scheme uses online and offline double keys to protect user privacy. However, the solution does not solve the inherent problem of maintaining certificates at a high cost by building PKI on a blockchain. Therefore, Ao et al. [3] to solve the certificate management problem of PKI on the blockchain, constructed PKG on Ethereum, and constructed a blockchain-based IBE signature scheme to solve the certificate management problem. At the same time, the scheme combines PKG and smart contract on the blockchain to jointly undertake the key generation and applies the blockchain to the key generation process. Li et al. [14] considered that the disadvantage of certificateless cryptography is

that verification of public key requires pre-broadcast, while blockchain provides a platform for sharing public information. This means that a user's public key can be shared via blockchain, enabling certificateless encryption in a blockchain-based Internet of Things system. Gervais et al. [11] then proposed a certificateless authenticated key agreement for wireless body area networks on the blockchain.

Tseng et al. [21] proposed an anonymous certificateless multi-receiver encryption scheme, which can verify the sender's identity while ensuring anonymity. Pang et al. [18] proposed an anonymous certificateless multi-message and multi-receiver signcryption scheme, which realized the anonymity of the receiver by the sender authorizing the decryptable receiver. Although the scheme protects the privacy of the receiver in ad-hoc networks, it does not consider the anonymity of the sender. Further, considering that the security of the partial private key depends on interactive channels, Pang et al. [17] proposed an anonymous certificateless multi-receiver signcryption scheme, which allows KGC to send a pseudo-partial private key to the user in the key extraction algorithm using only public channels. The specified user can extract the real partial private key from the pseudo-partial private key, while other users cannot. Mandal et al. [16] implemented secure access control over the Internet of Things through certificateless signcryption and communicated securely by establishing session keys. Guo et al. [12] proposed a controllable lightweight secure certificateless signature algorithm based on the federated blockchain security architecture of Hyperledger Fabric and edge computing. KGC saves all seeds for the user and restores the pseudonym and public key that belong to a particular ID. In low security, a single pseudonym and multiple keys are used. In high security, each pseudonym and key can only be used once. Although anonymity is guaranteed, KGC still maintains a list of pseudonyms required, and once KGC is in the hands of an attacker, the pseudonym list is also exposed. Cheng et al. [8] proposed a lightweight key negotiation mechanism using blockchain, certificateless encryption, and elliptic curve encryption, choosing the smart contract to act as KGC to provide session keys. Wang et al. [22] proposed a certificateless scheme using a smart contract on Ethereum to replace the role of KGC. Xu et al. [23] replaced the pseudonym form of user identity with a public key to achieve a certain certificateless anonymity and considered queries on blockchain to construct a hash table with user identity hash value and public key as key-value pairs.

1.2 Our Contribution

The main contributions of this paper are summarized as follows:

- We design a new anonymous certificateless signcryption scheme on blockchain for mitigating the aforementioned possible attacks and the interaction process does not use a secure channel.
- We design a one-time anonymous algorithm, which further disguises the public key based on the public key. The sender actively generates the pseudonym public keys of both communicating parties and does not need to maintain the list.

– We give anonymous authentication on blockchain and prove the security of the scheme through the random oracle model. Compared with other schemes, this scheme has better performance through experiments.

1.3 Organization

This paper is organized as follows. Section 2 gives the corresponding background, algorithm model, and security model of the one-time anonymous certificateless signcryption (OTACLSC) scheme. Section 3 presents a specific OTACLSC scheme. After that, Sect. 4 gives the security analysis of the scheme. Section 5 simulates the scheme and performs performance analysis. Finally, the paper is summarized in Sect. 6.

2 Preliminaries

2.1 Bilinear Mapping

Suppose there are two cyclic groups G_1, G_2, where G_1 is an additive group of prime order p, G_2 is a multiplicative group of prime order p, P is a random generator of G_1, and there exists a mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$, satisfying the following properties:

Bilinear: Any $P, Q \in G_1$, $a, b \in Z_p^*$, there is $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
Nondegenerate: There is $P, Q \in G_1$, such that $\hat{e}(P, Q) \neq 1$.
Computable: Any $P, Q \in G_1$, $\hat{e}(P, Q)$ can be calculated effectively.

2.2 Hard Problems

The security of this paper depends on the difficulty of the following hard problems.

Definition 1. *Given groups G_1 and G_2 of the same prime order p, P is a generator of G_1, there exists the mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$. The q-bilinear Diffie-Hellman inversion (q-BDHI) problem is given $(P, \alpha P, \alpha^2 P, ..., \alpha^q P)$ to compute $\hat{e}(P, P)^{\frac{1}{\alpha}}$, where $\alpha \in Z_p^*$.*

Definition 2. *Given groups G_1 and G_2 of the same prime order p, P is a generator of G_1, there exists the mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$. The q-strong Diffie-Hellman (q-SDH) problem is given $(P, \alpha P, \alpha^2 P, ..., \alpha^q P)$ to compute $\hat{e}(w, \frac{1}{\alpha+w} P)$, where $\alpha \in Z_p^*$.*

2.3 Smart Contract

The smart contract is a computerized transaction protocol that can be verified digitally and automatically execute the terms of the contract and can conduct trusted transactions without a third party, without the need for an agency [9]. In this paper, the smart contract SC_{KGC} is used to generate keys instead of KGC and distribute them to the corresponding users, and any legitimate user can query the public parameters of this system. Meanwhile, the receiver can look up the passively generated pseudonym public key.

2.4 Algorithm Model

There are seven steps for our OTACLSC scheme.

Setup. The algorithm is run by smart contract SC_{KGC} on the blockchain system, inputs the security parameter k, generates the master secret key s and system parameter $params$. System parameters have been omitted in the following steps for simplicity.

Set Partial Private Key. The algorithm is run by smart contract SC_{KGC}, inputs a user's identity $ID \in \{0,1\}^*$ and a master secret key s, outputs a partial private key D_{ID}.

Set Public/Private Key. The algorithm is run by the user, inputs a partial private key D_{ID}, outputs public key PK_{ID}, private key SK_{ID}, and user identity tag M_{ID}.

Set Pseudonym Public Key. The algorithm is run by the user, inputs the public key PK_A of the sender and the public key PK_B of the receiver, and outputs the pseudonym public key PPK_A and PPK_B of both parties.

Signcryption. The algorithm is run by the user, inputs a message m, a sender's private key SK_A and identity tag M_A, and the pseudonym public key PPK_A of the sender and the pseudonym public key PPK_B of the receiver, and outputs a ciphertext δ.

Check. The algorithm is run by smart contract SC_{KGC}, inputs the pseudonym public key PPK_B submitted by the sender and the private key SK_B submitted by the receiver, and outputs the pseudonym public key PPK_B of the receiver.

Verification. The algorithm is run by the user, inputs the ciphertext δ, the sender's pseudonym public key PPK_A, and the receiver's pseudonym public key PPK_B and the private key SK_B. If the verification is successful, message m is output; otherwise, failure symbol $\perp$ is output.

2.5 Security Model

In this paper, we need to consider two types of adversaries [1]. Adversary $\Im_I$ does not possess the master secret key but can legally replace the pseudonym public key. $\Im_{II}$ knows the master secret key, but it cannot replace the user's pseudonym public key. In addition, a signcryption scheme should satisfy confidentiality (i.e. indistinguishability against adaptive chosen ciphertext attack) and unforgeability (i.e. existential unforgeability against adaptive chosen messages attack) [5].

Definition 3. *The OTACLSC scheme is IND-CCA2 secure if adversary $\Im$ does not have a polynomial probability time t has at least a non-negligible probability advantage $\epsilon \geq Adv^{IND-CCA2}(\Im)$ in the game.*

The first game (Game-I) is a confidentiality game played between adversary $\Im_I$ and challenger C.

Initial: Challenger C generates system parameters and sends them to adversary $\Im_I$.

Phase 1: Adversary $\Im_I$ performs adaptive queries.

- Partial private key queries: $\Im_I$ submits an ID to C, C generates partial private key D_{ID} and sends it to $\Im_I$.
- Private key queries: $\Im_I$ submits an ID to C, C generates private key SK_{ID} and sends it to $\Im_I$.
- Pseudonym public key queries: $\Im_I$ submits an ID to C, C generates pseudonym public key PPK_{ID} and sends it to $\Im_I$.
- Pseudonym public key replacement queries: $\Im_I$ can replace the pseudonym public key PPK_{ID} with a selected value, and save it as a new pseudonym public key from C.
- Signcryption queries: $\Im_I$ selects a message m, a sender's ID_i and a receiver's ID_j to send to C. C generates the sender's private key SK_i, pseudonym public key PPK_i and receiver's pseudonym public key PPK_j. Then, C generates the ciphertext δ and sends it to $\Im_I$.
- Unsigncryption queries: $\Im_I$ sends the ciphertext δ, a sender's ID_i and a receiver's ID_j to C. C generates the receiver's private key SK_j, pseudonym public key PPK_i and receiver's pseudonym public key PPK_j. Then, C generates the message m and sends it to $\Im_I$.

Challenge: $\Im_I$ decides when phase 1 ends. $\Im_I$ selects two message (m_0, m_1) of the same length, a sender's identity ID_A and a receiver's identity ID_B, and sends them to C, on which it wishes to be challenged. Among them, ID_B cannot be submitted to the private key query in phase 1. ID_B also cannot be used for both the partial private key query and the pseudonym public key replacement query. C chooses a random bit $\beta \in \{0, 1\}$ and generates the ciphertext δ^* from the *Signcryption* algorithm and sends it to $\Im_I$.

Phase 2: Adversary $\Im_I$ performs adaptive queries, as in phase 1. But $\Im_I$ cannot perform a private key query on the target identity ID_B. If the pseudonym public key PPK_B of the target identity ID_B has been replaced before the challenge phase, $\Im_I$ cannot query its partial private key. $\Im_I$ cannot ask an unsigncryption query on the target ciphertext δ^* unless the pseudonym public key PPK_A or PPK_B has been replaced after the challenge phase.

Guess: $\Im_I$ guesses β^*, if $\beta^* = \beta$, then game wins. $\Im_I$'s probability of winning this game is $Adv^{IND-CCA2-I}(\Im_I) = |2Pr[\beta^* = \beta] - 1|$.

The second game (Game-II) is a confidentiality game played between adversary $\Im_{II}$ and challenger C.

Adversary $\Im_{II}$ performs queries as in Game-I. Among them, there is no need to perform partial private key queries. Because $\Im_{II}$ can get master secret key s.

$\Im_{II}$'s probability of winning is $Adv^{IND-CCA2-II}(\Im_{II}) = |2Pr[\beta^* = \beta] - 1|$.

Since the adversary knows the private keys of all senders, the Game-I and Game-II have the insider security in terms of confidentiality [2]. The insider security assures the forward security of the signcryption scheme. It means that if the sender's private key is disclosed, confidentiality is still maintained.

Definition 4. *The OTACLSC scheme is EUF-CMA secure if adversary* $\Im$ *does not have a polynomial probability time* t *has at least a non-negligible probability advantage* ϵ *in the game.*

The third game (Game-III) is an unforgeability game played between adversary $\Im_I$ and challenger C.

Adversary $\Im_I$ performs adaptive queries, as in the Game-I.

Forgery: $\Im_I$ outputs the ciphertext δ^*, a sender's identity ID_A and a receiver's identity ID_B. $\Im_I$ will win the game if the target ciphertext δ^* can be used to obtain the target message m^* by an unsigncryption query, $\Im_I$ cannot ask a private key query, both the partial private key query and the pseudonym public key replacement query for ID_A, and a signcryption query on the target message m^*. The advantage of $\Im_I$ is defined as the probability that it wins.

The fourth game (Game-IV) is an unforgeability game played between adversary $\Im_{II}$ and challenger C.

Adversary $\Im_{II}$ performs queries as in Game-II and Game-III.

In the Game-III and Game-IV, the receiver's private key SK_B is allowed to be known to the adversary. For unforgeability, the insider security can be obtained [2].

3 One-Time Anonymous Certificateless Signcryption Scheme

In this section, we assume that Alice is the sender A and Bob is the receiver B. In the following steps, they are abbreviated as A and B.

Setup. Input the security parameter k, select two cyclic groups $<G_1, G_2>$, where G_1 is the additive group of a large prime order $p > 2^k$, G_2 is the multiplicative group of the same prime order p, and there exists the mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$. P is a random generator in G_1, $g = \hat{e}(P, P)$. There are four hash functions $H_1 : \{0,1\}^* \rightarrow Z_p^*$, $H_2 : G_1 \rightarrow Z_p^*$, $H_3 : G_2 \rightarrow \{0,1\}^n$ and $H_4 : \{0,1\}^n \times G_2 \times G_1 \times G_2 \rightarrow Z_p^*$. Here n is the number of bits of a message to be sent. SC_{KGC} randomly selects the master secret key $s \in Z_p^*$, and set $P_{Pub} = sP$. Publish system parameters $params = <G_1, G_2, \hat{e}, n, p, P, P_{Pub}, g, H_1, H_2, H_3, H_4>$ on blockchain and secretly store the master secret key s.

Set Partial Private Key. The sender A inputs the identity ID_A used to construct $C_A = ID_A + ID_A P_{Pub}$ and $E_A = ID_A P$. Since the master secret key s is known only to SC_{KGC}, the sender can transmit the identity ID_A over the public channel. The smart contract SC_{KGC} is able to get the ID_A by means of $C_A - sE_A = ID_A + ID_A sP - sID_A P = ID_A$. Then, SC_{KGC} calculates

$D_A = sH_1(ID_A)$ and returns $CK_A = D_A + sE_A$ to A through the public channel, where D_A is the partial private key.

Set Public/Private Key. The sender A receives the CK_A and calculates the partial private key D_A by $CK_A - ID_A P_{Pub} = D_A + sE_A - ID_A sP = D_A + sID_A P - sID_A P = D_A$, and then confirms that the partial private key D_A is valid by whether the equation $\hat{e}(D_A, P) = \hat{e}(sH_1(ID_A), P) = \hat{e}(H_1(ID_A), sP) = \hat{e}(H_1(ID_A), P_{Pub})$ holds. If it holds, the partial private key D_A is correct.

The sender A generates $V_A = D_A + u_A$ and constructs the public key $PK_A = V_A P \in G_1$ according to the system parameters, where $u_A \in Z_p^*$. Then A randomly selects the secret value $x_A \in Z_p^*$, makes $Y_A = x_A + H_2(PK_A)$, generates user identity tag $M_A = g^{Y_A^{-1}} \in G_2$ and constructs private key $SK_A = Y_A^{-1} V_A^{-1} P$. The secret values u_A and x_A and the private key SK_A require local secret storage, while the public key PK_A and user identity tag M_A will be publicly published on blockchain.

Set Pseudonym Public Key. Input the public key PK_A of sender A and the public key PK_B of receiver B, and randomly select $\theta \in Z_p^*$ to generate the pseudonym public key $PPK_A = (\theta + H_2(\theta PK_B))PK_A$ of A and the pseudonym public key $PPK_B = (\theta + H_2(\theta PK_A))PK_B$ of B. The random number θ selected in the pseudonym public key is not public and does not need to be saved. A randomly generates the one-time pseudonym public key of the communication parties. The pseudonym public keys PPK_A and PPK_B are embedded in the sender and receiver of the transaction and published on blockchain.

Signcryption. The sender A randomly selects $\gamma \in Z_p^*$ and computes $\alpha = \gamma(\theta + H_2(\theta PK_A))$, $r = g^\alpha$ and $c = m \oplus H_3(r)$. Then get PPK_B to compute $T = \gamma PPK_B$ and $Z = (\theta + H_2(\theta PK_B))^{-1} SK_A + H_3(r)P$. A makes $h = H_4(m, M_A, PPK_B, r)$ and $S = (\alpha + h)(\theta + H_2(\theta PK_B))^{-1} V_A^{-1} P$. When publishing the ciphertext $\delta = (c, T, Z, S, t_1)$ on blockchain through the transaction, add the current timestamp t_1, where t_i is the timestamp.

Check. The sender A constructs $L_A = PPK_B + (\theta + H_2(\theta PK_A))P_{pub}$, $K_A = (\theta + H_2(\theta PK_A))P$, and submits $\{L_A, K_A, t_2\}$. Receiver B randomly selects $\vartheta, \tau \in Z_p^*$, generates $L_B = \vartheta Y_B + \tau P_{pub}$ and $K_B = \tau P$, and searches for the passively generated pseudonym public key PPK_B by submitting $\{L_B, K_B, \vartheta^{-1} SK_B, t_3\}$ to SC_{KGC}. SC_{KGC} verifies the timestamp $t_3 - t_2 < \triangle t_1$ and receives the input that meets the preset time threshold $\triangle t_1$. The pseudonym public key PPK_B can be transmitted over a public channel by way of $L_A - sK_A = PPK_B + (\theta + H_2(\theta PK_A))P_{pub} - (\theta + H_2(\theta PK_A))P_{pub} = PPK_B$. Similarly, the input submitted by receiver B can also be transmitted publicly according to $L_B - sK_B = \vartheta Y_B + \tau P_{pub} - \tau P_{pub} = \vartheta Y_B$. After calculating PPK_B and ϑY_B, determine if the equation holds by using Eq. 1.

$$\begin{aligned} \hat{e}(\vartheta Y_B \cdot PPK_B, \vartheta^{-1} SK_B) =& \hat{e}(Y_B(\theta + H_2(\theta PK_A))PK_B, SK_B) \\ =& \hat{e}(Y_B(\theta + H_2(\theta PK_A))V_B P, Y_B^{-1} V_B^{-1} P) \\ =& \hat{e}((\theta + H_2(\theta PK_A))P, P) \\ =& \hat{e}(K_A, P). \end{aligned} \tag{1}$$

If it holds, then the passively generated pseudonym public key PPK_B is found. Otherwise, the output is failure symbol $\perp$. After that, SC_{KGC} passes $PPK_B + sK_B$ through the public channel. Based on the random value τ, the receiver B can construct τP_{pub} and recover the pseudonym public key PPK_B according to $PPK_B + sK_B - \tau P_{pub} = PPK_B + sK_B - sK_B = PPK_B$.

Verification. The receiver B searches for the corresponding transaction information based on the obtained pseudonym public key PPK_B and retrieves the ciphertext δ. B verify whether $t_4 - t_1$ is within the time threshold $\triangle t_2$, where t_4 is the current timestamp. B can pass V_B^{-1} and T in the ciphertext δ to get r.

$$\begin{aligned}\hat{e}(T, V_B^{-1}P) =& \hat{e}(\gamma(\theta + H_2(\theta PK_A))V_B P, V_B^{-1}P) \\ =& \hat{e}(\gamma(\theta + H_2(\theta PK_A))P, P) \\ =& \hat{e}(\alpha P, P) \\ =& r.\end{aligned} \tag{2}$$

After calculating the r, it can compute $m = c \oplus H_3(r)$. By using the pseudonym public key PPK_A of the sender A and the Z in the ciphertext δ, B can get the identity tag M_A of A.

$$\begin{aligned}\hat{e}(PPK_A, Z - H_3(r)P) =& \hat{e}((\theta + H_2(\theta PK_B))PK_A, \\ & (\theta + H_2(\theta PK_B))^{-1} SK_A) \\ =& \hat{e}(V_A P, Y_A^{-1} V_A^{-1} P) \\ =& \hat{e}(P, Y_A^{-1} P) \\ =& M_A.\end{aligned} \tag{3}$$

Then B obtains $h = H_4(m, M_A, PPK_B, r)$ and verifies whether $r = \hat{e}(PPK_A, S)g^{-h}$ holds according to Eq. 4.

$$\begin{aligned}\hat{e}(PPK_A, S)g^{-h} =& \hat{e}((\theta + H_2(\theta PK_B))PK_A, (\alpha + h) \\ & (\theta + H_2(\theta PK_B))^{-1} V_A^{-1} P)g^{-h} \\ =& \hat{e}(V_A P, (\alpha + h)V_A^{-1} P)g^{-h} \\ =& \hat{e}(P, (\alpha + h)P)g^{-h} \\ =& g^{(\alpha+h)} g^{-h} \\ =& r.\end{aligned} \tag{4}$$

If it holds, it means that the r calculated in Eq. 2 is correct, thus confirming that the message m is correct, and also detecting and confirming the identity tag M_A. Otherwise, the output is failure symbol $\perp$.

By implementing the above steps, the receiver B completes the OTACLSC verification process (Fig. 1).

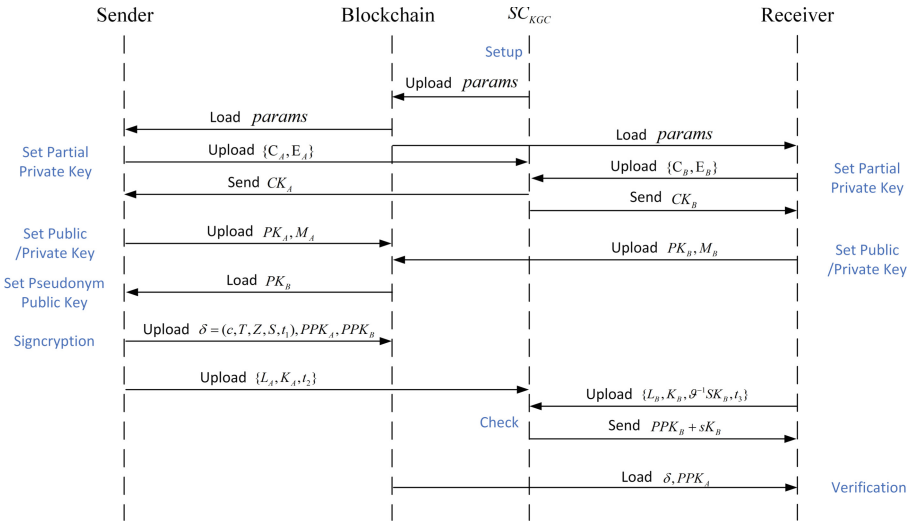

Fig. 1. One-time anonymous certificateless signcryption scheme process.

4 Security Analysis

This section presents the formal and informal security analysis of our scheme to show its resilience against various attacks.

4.1 Formal Security

In this subsection, we use the security model to prove the security of our OTA-CLSC scheme under the random oracle model.

Lemma 1. *Under the random oracle model, if there exists an adversary* $\Im_I$ *with a non-negligible probability advantage* ϵ *that wins the IND-CCA2-I security model game. Then* C *can solve the q-BDHI problem.*

Proof. Given a random instance $(P, \alpha P, \alpha^2 P, ..., \alpha^q P)$ of the q-BDHI problem, C solves the q-BDHI problem using $\Im_I$ and generates the following interaction procedure.

Initial: In Game-I, challenger C runs the *Setup* algorithm, selects the target $l \in \{1, .., q_{H_1}\}$, and generates $e_l \in Z_p^*$ and $w_1, .., w_{l-1}, w_{l+1}, w_q \in Z_p^*$ randomly. C sets $e_i = e_l - w_i$, $X = \alpha Q$, where $i \in \{1, .., l-1, l+1, q\}$, and generates elements $Q \in G_1$ such that there are q-1 pairs $(w_i, \frac{1}{\alpha + w_i} Q)$.

C chooses $c_0, .., c_{q-1} \in Z_p^*$ to expand $f(z) = \prod_{i=1, i \neq l}^{q} (z + w_i)$ into $f(z) = \sum_{j=0}^{q-1} c_j z^j$. Calculate the generating elements $Q = \sum_{j=0}^{q-1} c_j (\alpha^j P) = f(\alpha) P$ and $X = \sum_{j=1}^{q} c_{j-1} (\alpha^j P) = \alpha f(\alpha) P = \alpha Q$. For $(w_i, \frac{1}{\alpha + w_i} Q)$, this can be computed [7] by expanding $f_i(z) = \frac{f(z)}{z + w_i} = \sum_{j=0}^{q-2} d_j z^j$ into $\sum_{j=0}^{q-2} d_j (\alpha^j P) = f_i(\alpha) P = \frac{f(\alpha)}{\alpha + w_i} P = \frac{1}{\alpha + w_i} Q$.

Phase 1: Adversary $\Im_I$ sends a series of queries to C. C maintains the lists corresponding to the queries.

- H_1 queries: For the $H_1(ID_i)$ queries, C returns a result e_i and inserts (ID_i, e_i) in the list L_1.
- H_2 queries: These queries first check if the corresponding H_2 values are set for $H_2(\theta PK_i)$ and returns the value if it exists. Otherwise, C randomly selects $h_{2,i} \in Z_p^*$, sends it to $\Im_I$, and inserts $(\theta PK_i, h_{2,i})$ in the list L_2.
- H_3 queries: These queries first check if the corresponding H_3 values are set for $H_3(r_i)$ and returns the value if it exists. Otherwise, C randomly selects $h_{3,i} \in \{0,1\}^n$, sends it to $\Im_I$, and inserts $(r_i, h_{3,i})$ in the list L_3.
- H_4 queries: These queries first check if the corresponding H_4 values are set for $H_4(m_i, M_i, PPK_i, r_i)$ and returns the value if it exists. Otherwise, C randomly selects $h_{4,i} \in Z_p^*$, sends it to $\Im_I$. In response to the next phase of the queries, C queries $H_3(r_i)$, gets the ciphertext $c_i = m_i \oplus h_{3,i}$ and $\xi_i = r_i \hat{e}(Q,Q)^{h_{4,i}}$, and inserts $(m_i, M_i, PPK_i, r_i, h_{4,i}, c_i, \xi_i)$ in the list L_4.
- Partial private key queries: $\Im_I$ send the identity ID_i to C for partial private key queries, if the target identity $i = l$, then C fails and stops. Otherwise, C knows that $H_1(ID_i) = e_i$ and returns the partial private key se_i to $\Im_I$.
- Private key queries: $\Im_I$ send the identity ID_i to C for private key queries, if the target identity $i = l$, then C fails and stops. Otherwise, C knows the partial private key se_i. Then C checks if there is $(ID_i, PPK_i, u_i, x_i, \theta_i)$ in the list L_{fk}. If not, C randomly selects $u_i, x_i \in Z_p^*$, generates a new information in the list L_{fk} and saves it, and returns the private key $SK_i = (x_i + h_{2,i})^{-1}(se_i + u_i)^{-1}Q$.
- Pseudonym public key queries: $\Im_I$ sends the identity ID_i to C, C checks if there is $(ID_i, PPK_i, u_i, x_i, \theta)$ information in the list L_{fk}, and if it exists, returns the pseudonym public key. Otherwise C selects $\theta \in Z_p^*$ randomly, makes the pseudonym public key $PPK_i = (\theta + h_{2,i})PK_i$, where $PK_i = (se_i + u_i)Q$, and inserts $(ID_i, PPK_i, u_i, x_i, \theta)$ in the list L_{fk} and return the pseudonym public key.
- Pseudonym public key replacement queries: $\Im_I$ sends the replaced pseudonym public key PPK_i of the identity ID_i to C, C updates the list L_{fk} and reset the information $(ID_i, PPK_i, \perp, \perp, \perp)$ of the ID_i.
- Signcryption queries: $\Im_I$ sends the message m, the sender's ID_i and the receiver's ID_j to C. If the target identity $l \neq i$, then C knows the private key SK_i of the sender and returns the result according to the *Signcryption* algorithm. If the target identity $l = i$ but $l \neq j$, then C knows the private key SK_j of the receiver. C randomly selects $\eta, \theta, h \in Z_p^*$ and computes $S = \eta\theta^{-1}(se_j + u_j)^{-1}Q$, $T = \eta PPK_l - hPPK_j = \eta(\theta + h_{2,l})(se_l + u_l)Q - h(\theta + h_{2,j})(se_j + u_j)Q$, $r = \hat{e}(T, (se_j + u_j)^{-1}Q)$, and $Z = \theta^{-1}SK_j + H_3(r)Q$, where h is the value of $H_4(m, M_l, PPK_l, r)$. Finally, C outputs $c = m \oplus H_3(r)$ and sends the ciphertext $\delta = (c, T, Z, S)$ to $\Im_I$.
- Unsigncryption queries: $\Im_I$ sends the ciphertext $\delta = (c, T, Z, S)$, the sender's ID_i and the receiver's ID_j to C. If the target identity $l \neq j$, then C knows the private key SK_j of the receiver and returns the result according to the

Verification algorithm. If the target identity $l = j$ but $l \neq i$, then C knows the private key SK_i of the sender. For a valid ciphertext, $\log_{(se_i+u_i)^{-1}Q}(\theta S - h(se_i + u_i)^{-1}Q) = \log_{(\theta+h_{2,l})(se_l+u_l)Q}T$ can be calculated, where h is the value of $H_4(m, M_i, PPK_i, r)$. So the equation $\hat{e}(T, (se_i + u_i)^{-1}Q) = \hat{e}((\theta + h_{2,l})(se_l + u_l)Q, \theta S - h(se_i + u_i)^{-1}Q)$ holds. C computes $\xi = \hat{e}(S, (\theta + h_{2,i})(se_i + u_i)Q)$ and searches for $(m_i, M_i, PPK_i, r_i, h_{4,i}, c, \xi)$ in the list L_4. If it does not exist, then C rejects the ciphertext δ, otherwise, C computes whether $\frac{\hat{e}(T,(se_i+u_i)^{-1}Q)}{\hat{e}((\theta+h_{2,l})(se_l+u_l)Q,\theta S)} = \hat{e}((\theta + h_{2,l})(se_l + u_l)Q,(se_i + u_i)^{-1}Q)^{-h_{4,i}}$ holds. If holds, C returns the corresponding message m_i. Otherwise, C rejects the ciphertext δ.

Challenge: $\Im_I$ sends two message (m_0, m_1) of the same length, the sender's ID_A and the receiver's ID_B to C. If the target identity $ID_l \neq ID_B$, then C fails. Otherwise, C randomly selects $c^* \in \{0,1\}^n, \lambda \in Z_p^*, S^* \in G_1$, computes $T^* = -\lambda\theta Q - \lambda h_{2,B}Q$, and sends the ciphertext $\delta^* = (c^*, T^*, Z^*, S^*)$ to $\Im_I$. If we define $\rho = \frac{\lambda}{\alpha}$ and $u_l = -\alpha - se_l$, we can expand the equation as $T^* = -\lambda\theta Q - \lambda h_{2,B}Q = -\alpha\rho\theta Q - \alpha\rho h_{2,B}Q = (se_B + u_B)\rho\theta Q + (se_B + u_B)\rho h_{2,B}Q = \rho(\theta + h_{2,B})(se_B + u_B)Q = \rho PPK_B$.

Phase 2: Adversary $\Im_I$ performs adaptive queries as in phase 1. But $\Im_I$ cannot perform a private key query and a partial private key query on the target identity ID_B if the pseudonym public key PPK_B has been replaced. $\Im_I$ cannot ask an unsigncryption query on the target ciphertext δ^*.

Guess: $\Im_I$ submits the β^* to determine whether $\beta^* = \beta$ holds. If equal, $\Im_I$ wins the game and C can solve the q-BDHI problem.

C from the list L_3 to get $(r_i, h_{3,i})$ randomly or from the list L_4 to get $(m_i, M_i, PPK_i, r_i, h_{4,i}, c_i, \xi_i)$ randomly. The correct element is $r_i = \hat{e}(Q, Q)^{\rho} = \hat{e}(P, P)^{f(\alpha)^2 \frac{\lambda}{\alpha}}$. Suppose $\xi^* = \hat{e}(P, P)^{\frac{1}{\alpha}}$, the q-BDHI problem can be solved [6] by equation $\hat{e}(Q, Q)^{\frac{1}{\alpha}} = \xi^{*^{(c_0^2)}} \hat{e}(\sum_{j=0}^{q-2} c_{j+1}(\alpha^j P), c_0 P)\hat{e}(Q, \sum_{j=0}^{q-2} c_{j+1}(\alpha^j)P)$.

In order to calculate the probability advantage of C, define the probability events. E_1: $\Im_I$ has not chosen to challenge the target identity ID_l as the receiver's identity. E_2: $\Im_I$ has asked a private key query on ID_l. E_3: $\Im_I$ replaces the pseudonym public key of the target identity ID_l and asks a partial private key query on ID_l before the challenge phase. E_4: C finds a conflict in the value of H_4 in the signcryption queries, and terminates the queries. E_5: C rejects a valid ciphertext in the unsigncryption queries, thus terminating the queries.

According to above analysis, we know that $Pr[\neg E_1] = \frac{1}{q_{H_1}}$, $Pr[E_4] = \frac{q_s(q_s+q_{H_4})}{2^k}$, and $Pr[E_5] = \frac{q_u}{2^k}$. Since the event E_1 contains the event E_2 and E_3, the probability that C not terminate the game is $Pr[\neg abort] = Pr[\neg E_1 \wedge \neg E_4 \wedge \neg E_5] \geq \frac{1}{q_{H_1}}(1 - \frac{q_s(q_s+q_{H_4})}{2^k})(1 - \frac{q_u}{2^k})$. Also, the probability that C obtains the record with the correct element from the list L_3 or list L_4 is $\frac{1}{q_{H_3}+2q_{H_4}}$. Therefore, the probability advantage is $\epsilon' \geq \frac{\epsilon}{q_{H_1}(q_{H_3}+2q_{H_4})}(1 - \frac{q_s(q_s+q_{H_4})}{2^k})(1 - \frac{q_u}{2^k})$. C takes time to complete the preparation phase as $O(q_{H_1}^2)$ point multiplication

operations and time to complete the signcryption and unsigncryption queries as $O(q_s + q_u)$ bilinear pairing operations and $O(q_u q_{H_4})$ exponentiation operations.

Lemma 2. *Under the random oracle model, if there exists an adversary $\Im_{II}$ with a non-negligible probability advantage ϵ that wins the IND-CCA2-II security model game. Then C can solve the q-BDHI problem.*

Proof. Given a random instance $(P, \alpha P, \alpha^2 P, ..., \alpha^q P)$ of the q-BDHI problem, C solves the q-BDHI problem using $\Im_{II}$ and generates the following interaction procedure.

In Game-II, $\Im_{II}$ is similar to Game-I, expect pseudonym public key replacement queries.

Nonetheless, $\Im_{II}$ has the capability to steal the master key s and so has the possibility to crack the ciphertext of the target identity ID_l. When $\Im_{II}$ tries to crack a ciphertext, it must generate D_l beforehand, which is accessible to everyone via the smart contract SC_{KGC}. If successful, this means that 51% of the attacks are carried out in the blockchain system. That is, 51% of the attack requires $\frac{H_{\Im_{II}}}{H_t} \leq 51\%$, where $H_{\Im_{II}}$ denotes the available computational power of $\Im_{II}$ and H_t denotes the total computational power in the system. Due to the tamper-proof specification of smart contract and blockchain, it is almost impossible for $\Im_{II}$ to launch such an attack with sufficient hash power [22]. Therefore, the success probability ϵ' of adversary $\Im_{II}$ can be calculated as $\epsilon' \geq \frac{\epsilon}{q_{H_1}(q_{H_3}+2q_{H_4})}(1 - \frac{q_s(q_s+q_{H_4})}{2^k})(1 - \frac{q_u}{2^k}) + P(\frac{H_{\Im_{II}}}{H_t} \leq 51\%)$, where $P(\frac{H_{\Im_{II}}}{H_t} \leq 51\%)$ is the probability of occurrence of a 51% attack performed by $\Im_{II}$. C takes time to complete the preparation phase as $O(q_{H_1}^2)$ point multiplication operations and time to complete the signcryption and unsigncryption queries as $O(q_s + q_u)$ bilinear pairing operations and $O(q_u q_{H_4})$ exponentiation operations.

Lemma 3. *Under the random oracle model, if there exists an adversary $\Im_I$ with a non-negligible probability advantage ϵ that wins the EUF-CMA-I security model game. Then C can solve the q-SDH problem.*

Proof. In Game-III, the proof is similar to the proof of Game-I, and its intermediate procedure will not be recapitulated. Using the forking lemma [19], assume that $\Im_I$ forges a ciphertext with probability $\epsilon' \geq \frac{10(q_s+1)(q_s+q_{H_4})}{2^k}$, and then there exists time $t' \leq 120686 q_{H_1} \frac{t}{\epsilon}$ within which two valid ciphertext signatures (m, r, h_1, S_1), (m, r, h_2, S_2), where $h_1 \neq h_2$.

$\Im_I$ after inputting (X, ID_l) with sufficient time, two appropriate forged ciphertexts will be obtained (m_l, r, h_1, S_1), (m_l, r, h_2, S_2), where $h_1 \neq h_2$, but with common m_l and r. C recovers (ID_l, w_l) from the list L_1, where $w_l \neq w_1, .., w_{l-1}, w_{l+1}, w_q$ has a probability of at least $1 - \frac{q_{H_1}}{2^k}$.

If both forged ciphertexts satisfy the *Verification* algorithm, we can obtain the relation $\hat{e}(S_1, PPK)\hat{e}(Q, Q)^{-h_1} = \hat{e}(S_2, PPK)\hat{e}(Q, Q)^{-h_2}$, where $PPK = (\theta + H_2(\theta PK))(D + u)Q = (\theta + H_2(\theta PK))(w_l + \alpha)Q$, extracting the common part $\frac{1}{\theta + H_2(\theta PK)}$ of S_1 and S_2 and the relation becomes $\hat{e}(\frac{1}{\theta+H_2(\theta PK)}S_1', (\theta + H_2(\theta PK))(w_l + \alpha)Q)\hat{e}(Q, Q)^{-h_1} = \hat{e}(\frac{1}{\theta+H_2(\theta PK)}S_2', (\theta +$

$H_2(\theta PK))(w_l+\alpha)Q)\hat{e}(Q,Q)^{-h_2}$. The simplification gives $\hat{e}((h_1-h_2)^{-1}(S_1^{'}-S_2^{'}),(w_l+\alpha)Q)=\hat{e}(Q,Q)$, hence $T_l=(h_1-h_2)^{-1}(S_1^{'}-S_2^{'})=\frac{1}{w_l+\alpha}Q$.

Before returning the result (w_l,δ_l), use the long division expansion $f(z)=\gamma(z)(z+w_l)+\gamma_{-1}$ [7], where $\gamma(z)=\sum_{i=0}^{q-2}\gamma_i z^i$, $\gamma_{-1}\in Z_p^*$. Since $f(z)=\prod_{i=1}^{q-1}(z+w_i)$, which is not divisible by $(z+w_l)$, can be computed $\frac{f(z)}{z+w_l}=\frac{\gamma_{-1}}{z+w_l}+\sum_{i=0}^{q-2}\gamma_i z^i$. Finally, compute $\delta_l=\frac{1}{\gamma_{-1}}(T_l-\sum_{i=0}^{q-2}\gamma_i(\alpha^i P))=\frac{1}{\gamma_{-1}}(\frac{f(\alpha)}{\alpha+w_l}P+\frac{\gamma_{-1}}{\alpha+w_l}P-\frac{f(\alpha)}{\alpha+w_l}P)=\frac{1}{\alpha+w_l}P$.

If $\Im_I$ may forge a ciphertext with probability $\epsilon^{'}\geq\frac{10(q_s+1)(q_s+q_{H_4})}{2^k}$ in time t, then C can solve the q-SDH problem in the desired time $t^{'}\leq 120686q_{H_1}q_{H_4}\frac{t+O(q_s t_p)}{\epsilon(1-\frac{q_{H_1}}{2^k})}+O(q_{H_1}^2 t_m)$, where $O(q_{H_1}^2 t_m)$ is the time cost required for the point multiplication operation in the preparation phase.

Lemma 4. *Under the random oracle model, if there exists an adversary $\Im_{II}$ with a non-negligible probability advantage ϵ that wins the EUF-CMA-II security model game. Then C can solve the q-SDH problem.*

Proof. In Game-IV, the proof is similar to the proof of Game-II, and its intermediate procedure will not be recapitulated. Therefore, the success probability ϵ' of adversary $\Im_{II}$ can be calculated as $\epsilon'\geq\frac{10(q_s+1)(q_s+q_{H_4})}{2^k}+P(\frac{H_{\Im_{II}}}{H_t}\leq 51\%)$. C takes time to complete the signcryption queries as $O(q_s)$ bilinear pairing operations.

4.2 Informal Security

In this subsection, we will discuss some possible features of the scheme.

Anonymity: The purpose of anonymity is to hide the true identity, and various techniques are usually used to achieve this purpose. Assuming that an adversary captures the transmitted messages if the identity is not hidden, the adversary can easily know the true identity of the associated. Since we use the identity tag to indicate the origin of the pseudonym public key, the receiver can retrieve the corresponding sender's public key on the blockchain through the identity tag. Only the receiver can know the identity tag of the sender, but it is still not enough to know the true identity of the sender. Other users on the blockchain can only know that two anonymous users have made a transaction, but the real public keys of both are not known, only the pseudonym public keys of the transactions. Therefore, the OTACLSC scheme in this paper achieves anonymity.

KGC Compromised Attack: In the traditional anonymous certificateless scheme, there is always a centralized KGC. Although it is possible to build the KGC on the blockchain, there is a risk of the KGC being compromised, which can bring security vulnerabilities. To solve this problem, we use the smart contract instead of the original KGC. If there exists an adversary trying to control the whole blockchain system, this requires a 51% attack. From the perspective of the current blockchain system, this would undoubtedly spend a huge amount

of cost, and it is unlikely that an adversary could obtain some secret values or even crack the blockchain system. Therefore, the OTACLSC scheme proposed in this paper can successfully defend against the KGC compromised attack.

Replay Attack: An adversary intercepts a received signature and resends it for spoofing purposes, most likely during the verification process. It is assumed that all signatures generated or transmitted by the sender can be intercepted by the adversary. If the actual timestamp is not used, the adversary can replay the received signature to the verifier to perform fake verification and gain the trust of the verifier. Therefore, in this paper, timestamps are added as an important element in the querying and signcryption process, and then each verifier needs to check the validity of the timestamp. Finally, the OTACLSC scheme in this paper can prevent potential replay attacks.

Man-in-the-Middle Attack: Most certificateless schemes assume that the user interacts with the KGC through a secure channel, thus ignoring the high probability of the man-in-the-middle attack. However, in a real-world environment, this type of attack occurs from time to time. To prevent an adversary from intercepting and altering all messages transmitted in this channel, we construct secure encryption with elliptic curves using a random value τ submitted by the receiver so that the pseudonym public key is transmitted implicitly. The returned pseudonym public key can be decrypted only by the receiver who knows the random value, i.e., $PPK_B + \tau sP$. Therefore, the OTACLSC scheme in this paper excludes man-in-the-middle attacks.

Forgery Attack: Based on the security definition mentioned in the previous subsection, we can know that the advantage of an adversary forging a valid signature is negligible. Since the receiver can correctly verify the result using the pseudonym public key provided by SC_{KGC} and the corresponding ciphertext, the OTACLSC scheme proposed in this paper is secure against forgery attacks.

Big Data Clustering Attack: The information of all transactions can be viewed on the blockchain, and the real information of users can be inferred from the transaction records. If only a single pseudonym is used, the adversary can infer the corresponding transaction rule by obtaining the user's transaction records, thus exposing the real information. To resist such attacks, we adopt the one-time pseudonym approach and do not repeatedly use a pseudonym to prevent adversaries from associating pseudonyms. Therefore, the OTACLSC scheme in this paper can resist the big data clustering attack.

5 Performance Analysis

In this paper, we compare the computation and communication cost of the anonymous certificateless signcryption scheme with those of Tseng et al. [21], Mandal et al. [16], and Xu et al. [23].

The experiments are based on Intel(R) Core(TM) i5-11400 CPU @ 2.60 GHz, 16.00 GB RAM and PBC library [15] on Windows 10. The results are shown in

Table 1. Where h denotes the hash function operation, mul denotes the point multiplication operation in G_1, exp denotes the exponentiation operation in G_2, and $pair$ denotes the bilinear pairing operation in G_1. $|m|$ denotes the length of the message, $\left|Z_p^*\right|$ denotes the length of the element in Z_p^*, $|G_1|$ denotes the length of the element in G_1, and $|G_2|$ denotes the length of the element in G_2.

Table 1. Performance comparison of certificateless signcryption schemes

Schemes	Signcryption/encryption	Unsigncryption/decryption	Ciphertext size
Tseng et al. [21]	$2h + 5mul + 2pair + 1exp$	$2h + 2mul + 2pair + 1exp$	$2\left\|Z_p^*\right\| + 3\|G_1\| + \|G_2\| = 363$ *bytes*
Mandal et al. [16]	$12h + 11mul$	$9h + 3mul$	$1\|m\| + 3\left\|Z_p^*\right\| + 4\|G_1\| = 340$ *bytes*
Xu et al. [23]	$3h + 3mul + 3pair$	$3h + 2mul + 1pair$	$1\|m\| + 1\left\|Z_p^*\right\| + 1\|G_1\| = 105$ *bytes*
Ours	$2h + 4mul + 1exp$	$1h + 1mul + 3pair + 1exp$	$1\|m\| + 3\|G_1\| = 215$ *bytes*

5.1 Computation Cost

To show the performance difference between the schemes more intuitively, the code can be reproduced according to the algorithm design in each literature, and the results are shown in Fig. 2. The experimental results are taken as the average value after 100 times of the same operation.

From Table 1 and Fig. 2, it can be seen that our scheme has less computation cost than the other schemes in the signcryption process, and although the schemes of Mandal et al. [16] and Xu et al. [23] are better than our scheme in the unsigncryption process, the overall computation cost is still the smallest in our scheme. Therefore, the computation cost of this paper's OTACLSC scheme is better than the above schemes.

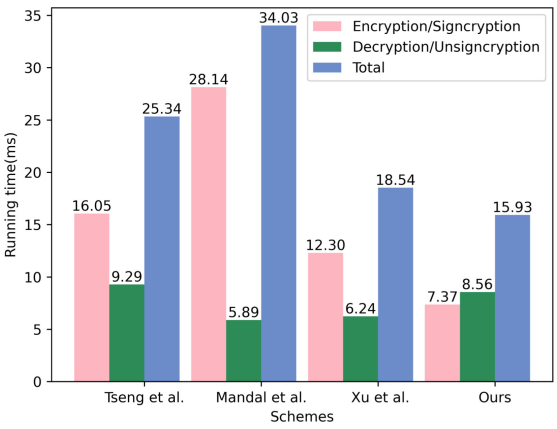

Fig. 2. Comparison of computation cost.

5.2 Communication Cost

For comparison, we assume that the message $|m|$ length is 160 bits, $\left|Z_p^*\right|$ length is 160 bits, $|G_1|$ length is 520 bits, and $|G_2|$ length is 1024 bits.

From Table 1, we can see that the ciphertext length used in our scheme is more advantageous than the schemes of Tseng et al. [21] and Mandal et al. [16], but longer than Xu et al. [23], because we add anonymity processing to protect our identity, while the scheme of Xu et al. [23] does not implement anonymity. The extra ciphertext length is acceptable considering the big data clustering attack. Therefore, the communication cost of the OTACLSC scheme in this paper is better than the above scheme.

5.3 Security Features Comparison

In this paper, we study the above security features and compare the anonymous certificateless signcryption scheme with other schemes in the literature, and the results are shown in Table 2.

Among the anonymity features, only Xu et al. [23] did not achieve anonymity. Due to the tamper-proof characteristic of blockchain, we use the smart contract to replace the function of KGC so that no one can falsify the data, not even the owner who issues the smart contract itself. At the same time, malicious nodes in the blockchain cannot access any private information of the smart contract, thus resisting the KGC compromise attack, which is not considered in other schemes. Considering the threat of the big data clustering attack, this paper uses a one-time pseudonym public key to circumvent the possible strong relationship between identity and public key, and can securely transmit relevant messages without passing through a secure channel, which is not reflected in other schemes. Therefore, the anonymous certificateless signcryption scheme in this paper can well solve the above attacks and achieve higher security.

Table 2. Comparison of security features

Features	Tseng et al. [21]	Mandal et al. [16]	Xu et al. [23]	Ours
Anonymity	Y	Y	N	Y
KGC compromised attack	N	N	N	Y
Replay attack	Y	Y	Y	Y
Man-in-the-middle attack	Y	Y	Y	Y
Forgery attack	Y	Y	Y	Y
Big data clustering attack	N	N	N	Y

Y: "a scheme is secure or it provides a functionality feature"; N: "a scheme is insecure or it does not provide a functionality feature".

6 Conclusion

In this paper, we propose a new OTACLSC scheme based on blockchain, which optimizes the computation cost and ciphertext length compared to other schemes and proves the security of this scheme under the random oracle model. This scheme is resistant to the possible attacks mentioned above and does not require a secure channel, so it requires less cost to maintain secure communication compared to other schemes. Due to the higher level of anonymity achieved, one does not have to worry about revealing one's identity in blockchain by the big data clustering attack. At the same time, this paper is one-time anonymity, which does not need to save the related data and reduces the overhead required to maintain the anonymity list. In the future, reducing the scheme cost and proposing more generalized anonymization methods will be the next research direction.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_29
2. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_6
3. Ao, W., Fu, S., Zhang, C., Huang, Y., Xia, F.: A secure identity authentication scheme based on blockchain and identity-based cryptography. In: 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), pp. 90–95. IEEE (2019)
4. Axon, L.: Privacy-awareness in blockchain-based PKI. CDT technical paper series **21**, 15 (2015)
5. Barbosa, M., Farshim, P.: Certificateless signcryption. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, pp. 369–372 (2008)
6. Barreto, P.S.L.M., Libert, B., McCullagh, N., Quisquater, J.-J.: Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 515–532. Springer, Heidelberg (2005). https://doi.org/10.1007/11593447_28
7. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
8. Cheng, G., Chen, Y., Deng, S., Gao, H., Yin, J.: A blockchain-based mutual authentication scheme for collaborative edge computing. IEEE Trans. Comput. Soc. Syst. **9**(1), 146–158 (2021)
9. Clack, C.D., Bakshi, V.A., Braine, L.: Smart contract templates: essential requirements and design options. arXiv preprint arXiv:1612.04496 (2016)
10. Fromknecht, C., Velicanu, D., Yakoubov, S.: A decentralized public key infrastructure with identity retention. IACR Cryptology ePrint Archive 2014/803 (2014)
11. Gervais, M., Sun, L., Wang, K., Li, F.: Certificateless authenticated key agreement for decentralized WBANs. In: Shen, B., Wang, B., Han, J., Yu, Y. (eds.) FCS 2019. CCIS, vol. 1105, pp. 268–290. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-0818-9_18

12. Guo, X., Guo, Q., Liu, M., Wang, Y., Ma, Y., Yang, B.: A certificateless consortium blockchain for IoTs. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp. 496–506. IEEE (2020)
13. Li, F., Han, Y., Jin, C.: Certificateless online/offline signcryption for the Internet of Things. Wireless Netw. **23**(1), 145–158 (2017). https://doi.org/10.1007/s11276-015-1145-3
14. Li, R., Song, T., Mei, B., Li, H., Cheng, X., Sun, L.: Blockchain for large-scale Internet of Things data storage and protection. IEEE Trans. Serv. Comput. **12**(5), 762–771 (2018)
15. Lynn, B.: PBC library: the pairing-based cryptography library (2013). https://crypto.stanford.edu/pbc/. Accessed 1 May 2022
16. Mandal, S., Bera, B., Sutrala, A.K., Das, A.K., Choo, K.K.R., Park, Y.: Certificateless-signcryption-based three-factor user access control scheme for IoT environment. IEEE Internet Things J. **7**(4), 3184–3197 (2020)
17. Pang, L., Kou, M., Wei, M., Li, H.: Anonymous certificateless multi-receiver signcryption scheme without secure channel. IEEE Access **7**, 84091–84106 (2019)
18. Pang, L., Wei, M., Li, H.: Efficient and anonymous certificateless multi-message and multi-receiver signcryption scheme based on ECC. IEEE Access **7**, 24511–24526 (2019)
19. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**(3), 361–396 (2000). https://doi.org/10.1007/s001450010003
20. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
21. Tseng, Y.F., Fan, C.I.: Provably CCA-secure anonymous multi-receiver certificateless authenticated encryption. J. Inf. Sci. Eng. **34**(6), 1517–1541 (2018)
22. Wang, W., Xu, H., Alazab, M., Gadekallu, T.R., Han, Z., Su, C.: Blockchain-based reliable and efficient certificateless signature for IIoT devices. IEEE Trans. Ind. Inform. **18**(10), 7059–7067 (2022)
23. Xu, G., Dong, J., Ma, C.: A certificateless encryption scheme based on blockchain. Peer-to-Peer Netw. Appl. **14**, 2952–2960 (2021). https://doi.org/10.1007/s12083-021-01147-w
24. Xu, S., Chen, X., He, Y.: EVchain: an anonymous blockchain-based system for charging-connected electric vehicles. Tsinghua Sci. Technol. **26**(6), 845–856 (2021)
25. Yuan, Y., Wang, F.: Current status and prospects of blockchain technology development. Acta Automatica Sinica **42**(4), 481–494 (2016)