Mathematical Biosciences and Engineering

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

MBE, 20(10): 18146–18172.
DOI: 10.3934/mbe.2023806
Received: 10 July 2023
Revised: 05 September 2023
Accepted: 12 September 2023
Published: 21 September 2023

Research article

Multi-message multi-receiver signcryption scheme based on blockchain

Xiao Dong Yang[1,*], Wen Jia Wang[1], Bin Shu[2], Mei Juan Li[1], Rui Xia Liu[1] and Cai Fen Wang[3]

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

[2] China Telecom WanWei Information Technology Co., LTD, Lanzhou 730030, China

[3] Department of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

* **Correspondence:** Email: y200888@163.com.

Abstract: In conventional message communication systems, the practice of multi-message multi-receiver signcryption communication encounters several challenges, including the vulnerability to Key Generation Center (KGC) attacks, privacy breaches and excessive communication data volume. The KGC necessitates a secure channel to transmit partial private keys, thereby rendering the security of these partial private keys reliant on the integrity of the interaction channel. This dependence introduces concerns regarding the confidentiality of the private keys. Our proposal advocates for the substitution of the KGC in traditional certificateless schemes with blockchain and smart contract technology. Parameters are publicly disclosed on the blockchain, leveraging its tamper-proof property to ensure security. Furthermore, this scheme introduces conventional encryption techniques to achieve user identity privacy in the absence of a secure channel, effectively resolving the issue of user identity disclosure inherent in blockchain-based schemes and enhancing communication privacy. Moreover, users utilize smart contract algorithms to generate a portion of the encrypted private key, thereby minimizing the possibility of third-party attacks. In this paper, the scheme exhibits resilience against various attacks, including KGC leakage attacks, internal privilege attacks, replay attacks, distributed denial of service attacks and Man-in-the-Middle (MITM) attacks. Additionally, it possesses desirable security attributes such as key escrow security and non-repudiation. The proposed scheme has been theoretically and experimentally analyzed under the random oracle model, based on the computational Diffie-Hellman problem and the discrete logarithm problem. It has been proven to possess confidentiality and unforgeability. Compared with similar schemes, our scheme has lower computational cost and shorter ciphertext length. It has obvious advantages in communication and time overhead.

Keywords: signcryption; blockchain; multi-receiver; discrete logarithm; confidentiality

1. Introduction

In the era of resource sharing and information exchange, it is common for a single user to send messages to multiple users. However, these messages are often different, and this type of communication is referred to as multi-message multi-receiver communication. Multi-message multi-receiver signcryption is a technique designed to achieve secure multicast communication, allowing a sender to simultaneously transmit identical messages to multiple recipients [1]. In contrast to traditional one-to-one communication methods [2], multi-receiver signcryption not only enhances computational efficiency, but also addresses the issue of receivers receiving inconsistent messages due to intentional deception or transmission errors. Widely regarded as a highly promising approach, multi-receiver encryption enables secure one-to-many communication and finds applications in various domains, including remote education, cloud data sharing and network conferences [3]. In recent years, scholars have proposed solutions that combine machine learning algorithms for processing large amounts of data and parallel computing [4, 5]. By selecting appropriate machine learning algorithms and data analysis routines, computational efficiency can be significantly improved. However, in the research on multi-message multi-receiver communication data, how to enhance communication quality and adapt to the requirements of high-speed information transmission remains an important issue.

In multi-party communications, information security issues, such as confidentiality and integrity during transmission , need to be considered. Typically, cryptographic techniques are used to maintain these two properties. In the course of this research, the first two approaches to combining encryption and digital signature technology were signing before encryption and encrypting before signing. Subsequently, a complete cryptographic scheme was proposed by combining encryption with digital signature technology. Unfortunately, such schemes have relatively low efficiency in practice. Based on the aforementioned research, a new primitive called signcryption [6] was proposed in 1997. It can complete signing and encryption in a single logical step, thereby improving the efficiency of message processing. Recently, the integration of signatures with various cryptographic systems such as Public Key Infrastructure (PKI) [7], Identity-Based Cryptograph (IBC) [8] and Certificateless Cryptograph (CLC) [9] has seen significant advancements. References [10–12] are all signcryption schemes proposed based on CLC. Liu et al. [13] proposed the first certificateless signcryption scheme based on RSA. They designed an efficient data access control scheme for Wireless body area networks using the proposed signcryption scheme. Zhang et al. [14] proposed a certificateless signcryption scheme where they employed a pseudonym mechanism to address privacy protection issues in Vehicular Sensor Networks. They proposed a self-generation mechanism for vehicle pseudonyms and their corresponding keys, eliminating the need for additional tamper-resistant devices. Reference [15] proposed a signcryption scheme based on cloud storage. Notably, they introduced a security model for the non-transferability and sender identity privacy of the signcryption scheme, which is a first in the field. They also provided concrete simulation proofs to support their proposed model. However, a practical challenge arises when considering existing CLC, where the KGC utilizes system master key information to facilitate the generation of user-specific public and private key information. This raises concerns regarding the security of the KGC. Wang et al. [16] have proposed the existence of an Advanced Persistent Threat (APT) attack on the KGC, wherein an attacker persistently targets the KGC with the goal of obtaining the system master key. Such an attack directly jeopardizes the security of the certificateless signcryption system. Additionally, apart from external attacks like APT on the KGC, there are inherent security issues associated with the KGC itself. According to the security model presented by Zhang et al. [17], the most formidable type of attacker is a malicious and active KGC. Although it may not replace the signer's public key information, this attacker can introduce trapdoor information during the system initialization phase, thereby adapting the system parameters in a computationally indistinguishable manner to undermine the system's security. To mitigate this potential security problem within the KGC, two approaches are commonly employed. First, enhancing the security of the algorithm itself is crucial to break existing linear relationships. Such attacks often exploit linear relationships present in the scheme's design, enabling a malicious KGC to manipulate certain controlled parameters and bypass the user's secret value, ultimately leading to the forgery of ciphertext information. Second, leveraging the tamper-evident nature of blockchain technology proves beneficial in preventing these security issues. By uploading the system parameters of the uncertificated signature system onto the blockchain [18], the integrity of the data can be maintained since the blockchain is resistant to tampering and modifications once the data has been uploaded.

In order to avoid the occurrence of the strongest type of attacker attacks, this research proposes a certificateless multi-message multi-receiver signcryption scheme based on blockchain (CMMSB). CMMSB incorporates traditional cryptography and elliptic curve technology to establish a secure and trustworthy KGC, which generates system-related parameters through a smart contract and publicly stores all parameters except the system master key on the blockchain [19–21]. Additionally, the smart contract serves as the KGC and generates an encrypted partial private key for users upon their request, effectively addressing the security channel issue present in traditional schemes. Furthermore, the combination of on-chain and off-chain approaches enhances both the security and efficiency of the system. The proposed scheme is proven to be confidential and unforgeable based on the complexity of the Discrete Logarithm (DL) and Computational Diffie-Hellman (CDH) difficulty problems under the random oracle model. It also demonstrates resilience against Distributed Denial of Service (DDoS) attacks and MITM attacks, while possessing security attributes such as key escrow security and non-repudiation. Compared to existing approaches, the CMMSB scheme offers several notable advantages.

1) In the proposed certificateless cryptographic scheme, the responsibilities of the traditional KGC are transferred to a smart contract on the blockchain. Key generation and management is decentralized across multiple nodes in the network. This eliminates the vulnerability of the second-class attacks present in traditional certificateless cryptographic schemes and makes the system more robust and secure.
2) By inputting a security parameter, the smart contract administers the generation of system-related parameters for the certificateless cryptographic system. To ensure system security, the smart contract securely stores these parameters on the blockchain while maintaining the tamper-evident nature of the technology.
3) Smart contracts are automatically executed and follow predefined logic and rules. Using smart contracts to replace traditional KGC, it means that there is no need for human intervention or reliance on third parties to perform key generation-related operations. This reduces the potential risks of errors and improper behavior.
4) In order to address the issue of user identity leakage in blockchain-based solutions, we leverage

traditional encryption techniques and elliptic curve technology to achieve user identity concealment. By employing this approach, blockchain nodes can generate the corresponding portion of a user's private key while ensuring the concealment of their identity. This solution effectively mitigates the risk of user identity exposure inherent in existing blockchain-based approaches. In this scheme, users encrypt their identity using a combination of their chosen private key and the master public key of the system. This process serves to safeguard their privacy, minimize the potential for hidden channel exposure and enhance communication concealment.

5) During the phase of partial private key generation, the smart contract functions as the KGC and generates an encrypted partial private key. Importantly, even if a third-party attacker manages to acquire the partial private key, they are unable to exploit it for malicious purposes. Only the rightful owner, i.e., the corresponding user, possesses the capability to utilize the partial private key to derive the correct private key.
6) Theoretical analysis confirms that the CMMSB scheme accomplishes secure communication with unforgeability and confidentiality, while imposing no additional communication overhead or computation time overhead. In comparison to similar schemes, the proposed scheme in this research achieves secure communication with unforgeability and confidentiality, demonstrates a shorter ciphertext length and offers notable advantages in terms of communication overhead and time overhead.

Section 2 of this paper introduces the current state of the art of certificateless signing and encryption. Section 3 introduces the basics of this paper, including the hard problem, system model and security model. Section 4 presents the multi-message multi-receiver signing scheme designed in this paper. Section 5 gives a proof of the security of the proposed scheme. Section 6 analyses the security features and performance of the scheme. Finally, the paper concludes with a summary of the entire study.

2. Related work

Due to the high cost associated with managing certificates in traditional public key cryptography, the reliance on fully trusted certificate authorities poses significant challenges. As a result, contemporary multi-message multi-receiver signing schemes tend to favor alternatives such as IBC and CLC. These advanced cryptographic approaches provide more efficient and practical solutions by eliminating the need for traditional certificate authorities. In 2007, Baek et al. [22] introduced the pioneering identity-based multi-receiver encryption (IBME) scheme, which requires only one pairing computation to encrypt a single message for multiple recipients. This scheme by Baek et al. departs from the traditional approach of repetitive encryption in multi-receiver encryption schemes, resulting in significantly enhanced encryption efficiency. Subsequently, scholars from both academia and industry have begun to devote themselves to the research of identity-based signcryption schemes. Scheme [23] proposed a multi-receiver signcryption scheme with decryption fairness to address the problems of receiver identity information leakage and signature unfairness in existing signature schemes. However, this scheme does not check the ciphertext legitimacy when unsigncrypt, and there is anonymity unfairness, ignoring the anonymity problem of the sender. The signcryption scheme proposed in literature [24] combines the identity information required by the receiver with the ciphertext information and achieves public verifiability based on the ciphertext information.

Scheme [25] is a single communication mode, which is not efficient if multiple messages are sent. In 2014, Zhou et al. [26] proposed a scheme that utilizes random re-encryption techniques. This scheme only requires one pairing computation to signcrypt multiple messages, resulting in high efficiency. Scheme [27] achieves a multi-message signcryption scheme based on bilinear pairing, which meets the security attributes of non-forgeability, confidentiality and non-repudiation; however, in order to meet the requirement of identity anonymity, the length of the ciphertext generated by this scheme is too long, resulting in inefficiency. Scheme [28] proposes an id-based multi-message and multi-recipient signcryption scheme with low computational and communication overheads to achieve authenticity and confidentiality. Scheme [29] makes use of existing signcryption techniques to generate a verifiable public ciphertext with a fixed length, and at the same time uses a broadcasting function to deliver the ciphertext to multiple recipients at the same time, with high transmission and computational efficiencies. However, it is important to note that these schemes primarily address the challenge of certificate management. One limitation of these schemes is that they heavily rely on the Private Key Generator (PKG) for private key generation. The trustworthiness of the PKG is crucial for ensuring the security of the system. In cases where the PKG is deemed fully trusted, the system can maintain its integrity. However, it is important to acknowledge that if the PKG lacks trustworthiness, it gains complete control over the user's private key, resulting in a key escrow problem. To address this issue, a CLC system has been proposed [9]. This system aims to resolve both the certificate management problem associated with the PKI system and the key escrow problem typically encountered in Identity-Based cryptosystems. In a CLC system, the user's private key comes from two parts, one is the secret value generated by the user's own choice, and the other is part of the private key generated by the KGC, and since the KGC does not have access to the user's complete private key, there is no key escrow problem. The advent of CLC has led to a significant surge in research on certificateless signatures within the field.

The field of multi-message multi-receiver signatures has also experienced a wave of research on certificateless signatures. Among the proposed schemes, certain ones [19, 20, 30] are built upon bilinear pairwise operations. It is worth noting that such operations impose a significant computational burden due to the extensive time required for their execution. To address this concern, researchers have begun investigating uncertificated multi-message multi-receiver signatures that eliminate the need for bilinear pairwise operations. Instead, they have focused on leveraging elliptic curves as the foundation for their research. Pang et al. [31] proposed an anonymous certificateless signcryption scheme that does not rely on bilinear pairings. However, the communication efficiency of the scheme is not optimal. Then, Pang et al. [32] introduced an efficient anonymous multi-message multi-receiver signcryption scheme based on Elliptic Curve Cryptography (ECC). They substantiated its security under a random prediction model. However, Peng et al. [33] subsequently discovered a vulnerability in the certificateless multi-message multi-receiver signing scheme proposed by Pang et al. [32]. This vulnerability allows an attacker to forge a user's legitimate private and public keys, posing a significant threat to the overall security of the scheme. Peng et al. [33] proposed an efficient and provably secure multi-receiver encryption scheme for multicast communication in edge computing, aiming to achieve secure communication. Fu et al. [34] proposed a practical anonymous multi-receiver certificateless signcryption scheme that can satisfy message confidentiality, source authentication and anonymity simultaneously and efficiently. Wang et al. [35] designed a certificateless multi-receiver signature and encryption scheme based on elliptic curves, specifically tailored for multicast communication in smart grids. This scheme exhibits excellent timing performance and high computational efficiency. Zhou et al. [36] presented a certificateless multi-receiver multi-message signing mechanism that does not rely on bilinear mapping. This innovative approach improves user anonymity and fulfills the sender's requirement for multi-message transmission. Notably, the scheme achieves high computational efficiency by eliminating the need for bilinear mapping and exponential operations in both the signing and declassification algorithms. Wang et al. [16] introduced an elliptic curve-based multi-message multi-recipient signing scheme. This scheme employs multiple KGCs to oversee the system master key using a threshold secret sharing protocol. Additionally, the researchers periodically update the individual KGC sub-secret information to enhance its resilience against persistent attacks. Despite the numerous proposed schemes, the issue of key distribution remains unresolved. In many cases, a secure channel is established during the design phase to transmit a portion of the private key generated by the KGC to the intended user. However, finding a solution to establishing a secure or secret channel during the design phase has become a crucial problem that researchers need to address. The papers [37] and [38] discuss a method that addresses the key distribution problem by encrypting a portion of the key generated by the KGC using heterogeneous or traditional calculations. The encrypted key is then transmitted to the user through the public channel.

3. Preliminaries

3.1. Difficult problems

Discrete logarithm (DL) problem: Given a binary group $(P, aP) \in G_p^2$, known to $a \in Z_p^*$ be unknown, solve for a , where the large prime p is the order of the cyclic group G_p and P is the generating element of G_p.

Computational Diffie-Hellman (CDH) problem: Given a ternary group $(P, aP, bP) \in G_p^3$, known to $a, b \in Z_p^*$ be unknown, solve for abP, where the large prime p is the order of the cyclic group G_p and the generating element of G_p.

3.2. System model

The multi-receiver multi-message signcryption scheme based on CLC system consists of the following six algorithms :

1) Setup

Given security parameters k, KGC generates the master key s and system parameters $params$. This algorithm can be defined as $Setup(k) \rightarrow (s, params)$.

2) PartKeyGen

Given the user identity ID and some related information, KGC uses the known user information and the master key s to generate a partial private key psk. This algorithm can be defined as PartKeyGen $(params, \mathrm{ID}, s) \rightarrow psk$.

3) KeyGen

Given system parameters $params$ and partial private key psk, users generate their own public key pk and private key sk. This algorithm can be defined as KeyGen $(params, \mathrm{ID}, psk) \rightarrow (pk, sk)$.

4) SignCrypt

The ciphertext sender ID_S needs to send n corresponding plaintext messages $m_i(m_i \in M, 1 \leq i \leq n)$ to n ciphertext receivers $\mathrm{ID}_{R_i}(1 \leq i \leq n)$. Given the system parameters, the identity of the ciphertext sender ID_S, the receiver identity set ID_R=$\{\mathrm{ID}_{R_1}$,$\mathrm{ID}_{R_2}, \cdots$,$\mathrm{ID}_{R_n}\}$ and the plaintext message set $M = \{m_1, m_2, \cdots, m_n\}$, the corresponding ciphertext set $C = \{c_{R_1}, c_{R_2}, \cdots, c_{R_n}\}$ is generated. This algorithm can be defined as SignCrypt $(params, \mathrm{ID}_S, \mathrm{ID}_R, M) \rightarrow C$.

5) UnSignCrypt

Given the system parameters *params* , the identity of the ciphertext sender ID_S , the private key information sk_{R_i} of the ciphertext receiver and the ciphertext message $c_{R_i}(c_{R_i} \in C, 1 \leq i \leq n)$, the receiver calculates the corresponding plaintext message $m'_{R_i}(1 \leq i \leq n)$. This algorithm can be defined as UnSignCrypt $(params, \mathrm{ID}_S, \mathrm{ID}_{R_i}, sk_{R_i}, c_{R_i}) \rightarrow m'_{R_i}$.

6) VerifySign

According to the known system parameters *params* and the public key information pk_{R_i} of the ciphertext receiver, the receiver ID_{R_i} verifies the result of the decryption, that is, the decrypted plaintext message m'_{R_i}. If the verification is successful, return True; otherwise, return False. This algorithm can be defined as VerifySign $(params, \mathrm{ID}_{R_i}, pk_{R_i}, m'_{R_i}) \rightarrow$ (True|False).

Correctness: The correctness of multi-receiver multi-message signcryption schemes based on certificateless cryptography mandates that the probability of these schemes failing to pass the verification equation, when generated in strict accordance with the given definition, can be considered negligible.

$$\Pr\left[\text{VerifySign } (params, \mathrm{ID}_{R_i}, pk_{R_i}, m'_{R_i}) \rightarrow (\text{True}|\text{False}) \;\middle|\; \begin{array}{c} \text{Setup } (k) \rightarrow (s, params) \\ \text{PartKeyGen } (params, \mathrm{ID}, s) \rightarrow psk \\ \text{KeyGen } (params, \mathrm{ID}, psk) \rightarrow (pk, sk) \\ \text{SignCrypt } (params, \mathrm{ID}_S, \mathrm{ID}_R, M) \rightarrow C \\ \text{UnSignCrypt } (params, \mathrm{ID}_S, \mathrm{ID}_{R_i}, sk_{R_i}, c_{R_i}) \rightarrow m'_{R_i} \end{array}\right] = \text{VALID}$$

3.3. Security model

The security model defined in references [12] and [13] involves two types of attackers, including Type I attackers and Type Π attackers.

1) The Type I attackers are represented by A_1, which mainly refers to the malicious signcryptor (ciphertext sender in this paper). The strongest Type I attacker possesses the capability to substitute the public key without providing the secret value. Consequently, they can obtain the plaintext message from the manipulated public key, even when the public and private keys have been replaced.

2) A category of attackers, referred to as "honest-but-curious" Type Π, possesses knowledge of the system master key but lacks the capability to obtain the secret value of the targeted user or manipulate the user's public key. In contrast, another type of attacker, known as "malicious-but-passive", possesses all the capabilities of the "honest-but-curious" Type Π attackers. In addition, they have the capability to dynamically initialize system parameters by incorporating trapdoor information into the primary key and system parameters utilizing computational indistinguishable methods.

Set the ciphertext sender to be ID_S, the ciphertext receiver's number is n, respectively, $\mathrm{ID}_R = \{\mathrm{ID}_{R_1}, \mathrm{ID}_{R_2}, \cdots, \mathrm{ID}_{R_n}\}$, and the plaintext message set to be signed is $M = \{m_1, m_2, \cdots, m_n\}$. The precise definitions and corresponding game descriptions for the aforementioned two types of attackers, namely "confidentiality under adaptive chosen ciphertext attack" and "unforgeability under adaptive chosen message attack", are outlined as follows. In order not to lose generality, this part will be based on the receiver of the i-th ciphertext, ID_{R_i}, how does he handle the message m_i. The attackers that select the confidentiality of ciphertext attack are Type I attacker A_{1-1} and Type Π attacker A_{2-1} and the attackers that select the unforgeability of message attack are Type I attacker A_{1-2} and Type Π attacker A_{2-2}.

Definition 1 Type I attacker A_{1-1} Confidentiality under chosen ciphertext attack.

If the Type I attacker A_{1-1} wins the following game with a negligible probability $Adv^{T}_{A_{1-1}}(k)$ in polynomial time by correlation calculation, the scheme satisfies confidentiality under A_{1-1} adaptive chosen ciphertext attack.

Game 1 The interaction process between the attackers and the challengers in the game can be described as follows:

1) System initialization phase k, challenger $\mathcal{T}$ executes the system initialization Setup algorithm, outputs the system master key s and system parameter *params*, and passes *params* to attacker A_{1-1}. Furthermore, the master key s is kept secretly.

2) Inquiry phase

i) Hash queries: Attacker A_{1-1} can complete the query of any Hash.

ii) Key generation queries: Attacker A_{1-1} selects the identity of a target user ID and asks challenger $\mathcal{T}$ with ID as input. Challenger $\mathcal{T}$ executes algorithm KeyGen $(params, \mathrm{ID}, psk) \rightarrow (pk, sk)$, then responds to A_{1-1} queries with final result (pk, sk) . In the process of security proof, user key query includes both private key generation query and public key generation query, which are described separately. Then, we respond with sk and pk.

iii) Public key replacement queries: Attacker A_{1-1} can replace the public key pk of target user ID at any time.

iv) Signcryption queries: Attacker A_{1-1} selects ciphertext sender ID_S .Then, as the sender's identity ID_S , plaintext message set M and receiver identity set $\mathrm{ID}_R = \{\mathrm{ID}_{R_1}, \mathrm{ID}_{R_2}, \cdots, \mathrm{ID}_{R_n}\}$ .Take the above parameters as input and ask the challenger. The challenger starts executing the algorithm , The final calculation result is ciphertext message $\mathcal{T}$ in response to SignCrypt $(params, \mathrm{ID}_S, \mathrm{ID}_R, M) \rightarrow C$ query.

v) Unsigncryption queries: Attacker A_{1-1} asks Challenger $\mathcal{T}$ with the input of sender's identity ID_S, receiver's identity ID_{R_i} and ciphertext message $(c_{R_i} \in C, 1 \leq i \leq n)$. The challenger $\mathcal{T}$ first executes the algorithm KeyGen $(params, \mathrm{ID}_{R_i}, psk_{R_i}) \rightarrow (pk_{R_i}, sk_{R_i})$ for the receiver ID_{R_i}. Through the above operations, the corresponding public key pk_{R_i} and private key sk_{R_i} are obtained. Then execute the algorithms UnSignCrypt $(params, \mathrm{ID}_S, \mathrm{ID}_{R_i}, sk_{R_i}, c_{R_i}) \rightarrow m'_{R_i}$ and VerifySign $(params, \mathrm{ID}_{R_i}, pk_{R_i}, m'_{R_i}) \rightarrow$ (True|False). If the verification result is True, then m'_{R_i} responds to query of A_{1-1}; otherwise, respond to A_{1-1} queries with "⊥".

3) Challenge phase

Attacker A_{1-1} selects ciphertext sender ID_S and receiver ID_R, two equal-length messages m_0 and m_1 and pass these messages to the challenger $\mathcal{T}$.The challenger $\mathcal{T}$ selects a random number $i \in \{0, 1\}$ and executes the algorithm SignCrypt $(params, \mathrm{ID}_S, \mathrm{ID}_R, m_i) \rightarrow c_i$ to return the calculation results c_i to the attacker A_{1-1}.

4) Conjecture phase

The attacker A_{1-1} can perform various operations in the query phase multiple times. However, in this process, it is not possible to initiate a private key generation query about the receiver ID_{R_i}. Moreover, it is not possible to initiate a decryption query on the ciphertext $c_i \in C$. If the attacker A_{1-1} successfully guessed $i' = i$, that is, A_{1-1} won the game. The probability is set to $Adv^{T}_{A_{1-1}}(k) = \left|\Pr[i = i'] - \frac{1}{2}\right|$.

Definition 2 Type Π attacker A_{2-1} chooses the confidentiality under ciphertext attack

If the attacker A_{2-1} passes the correlation calculation in a polynomial time range and wins the following games with a negligible probability of $Adv^{T}_{A_{2-1}}(k)$, the scheme satisfies confidentiality under the attacker A_{2-1} adaptive chosen ciphertext attack.

Game 2 Both sides of the game are the attacker A_{2-1} and the challenger $\mathcal{T}$, and the following is the interaction process between the two.

1) System initialization phase

Given a security parameter k, Challenger $\mathcal{T}$ executes the system initialization Setup algorithm, outputs the system master key s and system parameter *params* and passes s and *params* to attacker A_{2-1}.

2) Inquiry phase

Attacker A_{2-1} selects the identity ID of a target user and it can perform the same operation as the Definition 1 inquiry phase. However, since the type Π attacker cannot perform public key replacement, the public key replacement query is excluded.

3) Challenge phase Attacker A_{2-1} performs the same operation as the Definition 1 challenge phase.

4) Conjecture phase

Attacker A_{2-1} performs the same operation as the Definition 1 challenge phase. If the attacker A_{2-1} guesses $i' = i$, it means that A_{2-1} wins the game. The probability is set to $Adv^{T}_{A_{2-1}}(k) = \left|\Pr[i = i'] - \frac{1}{2}\right|$.

Definition 3 The unforgeability of Type I attacker A_{1-2} under selective message attack

If the attacker A_{1-2} passes the correlation calculation in a polynomial time range and wins the following games with a negligible probability of $Adv^{T}_{A_{1-2}}(k)$, the scheme satisfies unforgeability under A_{1-2} adaptive selection message attack.

Game 3 Both sides of the game are the attacker A_{1-2} and the challenger $\mathcal{T}$, and the following is the interaction process between the two.

1) System initialization phase

Given a security parameter k, Challenger $\mathcal{T}$ executes the system initialization Setup algorithm, outputs the system master key s and system parameter *params* and passes *params* to attacker A_{1-2}. At the same time, the secret custodian master key s.

2) Inquiry phase

Attacker A_{1-2} selects the identity ID of a target user and performs the same operation as the Definition 1 query phase.

3) Challenge phase

Attacker A_{1-2} forges ciphertext message C' , and then performs various query operations in the query phase. In this process, the attacker cannot initiate a private key generation query about the receiver ID_{R_i}, nor can he initiate a signcryption query about the ciphertext and a reconciliation signcryption query. If the final result of the decryption is True when the signature verification is performed. It shows that A_{1-2} won the game. The probability is set to $Adv^{T}_{A_{1-2}}(k) = \left|\Pr[\text{Forged signature can be verified}] - \frac{1}{2}\right|$.

Definition 4 The unforgeability of Type Π attacker A_{2-2} under chosen ciphertext attack

If the attacker A_{2-2} in the polynomial time range, through the relevant calculation with a probability of $Adv^{T}_{A_{2-2}}(k)$ can be ignored to win the following game, then the scheme satisfies unforgeability under A_{2-2} adaptive selection message attack.

Game 4 Both sides of the game are the attacker A_{2-2} and the challenger $\mathcal{T}$, and the following is the interaction process between the two.

1) System initialization phase

Given a security parameter k, Challenger $\mathcal{T}$ executes the system initialization Setup algorithm, outputs the system master key s and system parameter *params* and passes s and *params* to attacker A_{2-2}.

2) Inquiry phase

Attacker A_{2-2} selects the identity ID of a target user and performs the same operation as the Definition 2 query phase.

3) Challenge phase

Attacker A_{2-2} forges ciphertext message C' and then performs the same operation as the Definition 3 forgery phase. If the final result of the decryption is True when the signature verification is performed. It shows that A_{2-2} won the game. The probability is set to $Adv^{T}_{A_{2-2}}(k) = \left|\Pr[\text{Forged signature can be verified}] - \frac{1}{2}\right|$.

4. Scheme of this article

4.1. System model

As shown in Figure 1, the certificateless multi-message multi-receiver signcryption scheme based on blockchain includes the following entities.

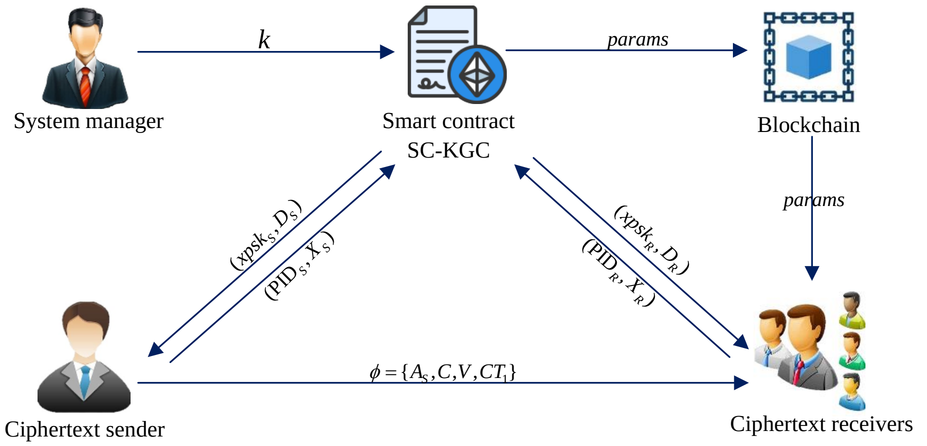

Figure 1. System model diagram.

1) System manager: Given a security parameter and initialize the smart contract.

2) Smart contract (SC-KGC) : According to the relevant information given by the user, it generates the corresponding partial key of the user.

3) Blockchain: Store the system parameters of the certificateless signcryption scheme to ensure the security of the certificateless system.

4) Ciphertext sender: The sender uploads the pseudonym identity and part of the public key information to SC-KGC to obtain the corresponding part of the private key. After having the necessary system parameters, signcryption generates the corresponding ciphertext information for the ciphertext receiver.

5) Ciphertext receivers: The receivers upload the pseudonym identity and part of the public key information to SC-KGC to obtain the corresponding part of the private key. For the received ciphertext information, decrypt the ciphertext information and verify whether the message is valid and legal.

To enhance the readability and comprehensibility of the proposed scheme in this paper, Table 1 provides an overview of the symbols used in the scheme. The contents are as follows.

Table 1. Description of relevant symbols.

Symbols	Description	Symbols	Description
Admin	System administrator	$xpsk_i$	Partial private key generated by user U_i
$SC - KGC$	Acting as a KGC smart contract	P_{pub}	System public key
Z_p^*	Non-zero Multiplicative Group based on a large prime number p	s	System master key
U_i	Users (including ciphertext sender and ciphertext receiver)	sk_i	The private key of user U_i
ID_i	The identity corresponding to the user U_i	pk_i	The public key of user U_i
k	Safety parameter	$Addr_{ID}$	The address of user U_i
p	The big prime number	${CTR^i}_{Addr_{ID}}$	A counter maintained by $SC - KGC$
G	Additive cyclic group		
P	Generators of G	C	Store the encrypted ciphertext set of message M
$params$	System parameter	M	Collection of clear text information
PID_i	The pseudonym identity of user U_i	V	The ciphertext set after storing the message M signature
X_i	Partial public key generated by user U_i	ϕ	Transmitted encrypted ciphertext information
D_i	The partial public key of user U_i is generated by $SC - KGC$	n	Number of ciphertext receivers

4.2. Scheme description

The proposed scheme encompasses five distinct stages: Initialization, partial private key generation, user key generation, signing and encryption and decryption and signature verification. Each stage serves a specific purpose in the overall functioning of the scheme. The following section provides a brief description of each stage.

1) Setup

i) Given a security parameter k, we define a additive cyclic group G of order p, where p is a prime number, and P is its generator.

ii) Define 6 secure Hash functions.: $H_0 : G \to \{0,1\}^{L_{ID}}$, $H_1 : \{0,1\}^{L_{ID}} \times G \times G \to Z_p^*$, $H_2 : \{0,1\}^{L_{ID}} \times G \to Z_p^*$, $H_3 : \{0,1\}^{L_{ID}} \times G \to \{0,1\}^{L_m}$, $H_4 : \{0,1\}^{L_{ID}} \times \{0,1\}^{L_{ID}} \times \{0,1\}^{L_m} \times G \times \{0,1\}^{L_{ID}} \to Z_p^*$, $H_5 : \{0,1\}^{L_{ID}} \times \{0,1\}^{L_{ID}} \times \{0,1\}^{L_m} \times G \times Z_p^* \times \{0,1\}^{L_{ID}} \to Z_p^*$, The length of the user identity ID is represented by L_{ID}, the length of the plaintext message m is represented by L_m and the length of the elements in Z_p^* represented by $|Z_p^*|$.

iii) Define the index function $F_{index}^n(\mathrm{ID}) \to Z_p^*$. This function can take n given user identification and return an index. This function can map the given n user identities to the domain $\{1, 2, \cdots, n\}$ one by one.

iv) The system randomly generates the master key $s \in Z_p^*$, and then calculates the system public key $P_{pub} = sP$ for future public use.

v) The visibility of system parameters $params = \{G, p, P, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, F_{index}^n()\}$ and function *PartKeyGen* () is set to publicly visible, enabling the users to complete the scheme under these initial parameters. The system master key is set to self-visible. Finally, the system master key is set to be visible only to itself.

The details of the smart contract are shown in Algorithm 1.

Algorithm 1 $SC - KGC$ Contract

```
Require: safety parameter $k$
Ensure: encrypted partial private key
Given an additive group $G$ of order $p$ and a generator $P$.
Select a random number $s \in Z_p^*$.
Calculate $P_{pub} = sP$.
Define 6 hash functions: $H_0 : G \to \{0,1\}^{L_{\mathrm{ID}}}$, $H_1 : \{0,1\}^{L_{\mathrm{ID}}} \times G \times G \to Z_p^*$, $H_2 : \{0,1\}^{L_{\mathrm{ID}}} \times G \to Z_p^*$,
  $H_3 : \{0,1\}^{L_{\mathrm{ID}}} \times G \to \{0,1\}^{L_m}$, $H_4 : \{0,1\}^{L_{\mathrm{ID}}} \times \{0,1\}^{L_{\mathrm{ID}}} \times \{0,1\}^{L_m} \times G \times \{0,1\}^{L_{\mathrm{ID}}} \to Z_p^*$, $H_5 :$
  $\{0,1\}^{L_{\mathrm{ID}}} \times \{0,1\}^{L_{\mathrm{ID}}} \times \{0,1\}^{L_m} \times G \times Z_p^* \times \{0,1\}^{L_{\mathrm{ID}}} \to Z_p^*$.
Set the visibility of the parameters: $params = \{G, p, P, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, F_{index}^n()\}$ to be
  publicly visible.
Function PartKeyGen ($PID_i, X_i$):
if $CTR^{i}{}_{Addr_{ID}} < n$ then
    $\mathrm{ID}_i = \mathrm{PID}_i \oplus H_0(s \cdot X_i)$
    $d_i \in Z_p^*$
    $D_i = d_i P$
    $h_1^i = H_1(\mathrm{ID}_i, X_i + D_i, P_{pub})$
    $xpsk_i = d_i + s \cdot h_1^i + H_2(\mathrm{ID}_i, s \cdot X_i) \bmod p$
    return $xpsk_i$
else
    end
end if
```

2) PartKeyGen

When the user U_i invokes the function *PartKeyGen* (), it will automatically generate a partial private key $xpsk_i$ for the user U_i.

i) U_i randomly selects $x_i \in Z_p^*$, then calculates $X_i = x_iP$ and $\mathrm{PID}_i = \mathrm{ID}_i \oplus H_0(x_i \cdot P_{pub})$.

ii) U_i inputs $\{\mathrm{PID}_i, X_i\}$ and evokes the function *PartKeyGen* ().

iii) The user U_i sends $\{\mathrm{PID}_i, X_i\}$ to the KGC for calculating their partial private key. First, $SC - KGC$ checks if there is an inequality $CTR^i{}_{Addr_{ID}} < n$, where $CTR^i{}_{Addr_{ID}}$ is a counter maintained by $SC - KGC$, $Addr_{ID}$ is the address of the user on the Ethereum platform, n is the reset threshold. If the access time is less than the threshold, $SC - KGC$ can calculate $\mathrm{ID}_i = \mathrm{PID}_i \oplus H_0(s \cdot X_i)$ to get the real identity of U_i, otherwise, $SC - KGC$ will reject the request immediately.

iv) $SC - KGC$ randomly selects $d_i \in Z_p^*$, then calculates $D_i = d_iP$ and $h_1^i = H_1(\mathrm{ID}_i, X_i + D_i, P_{pub})$.

v) $SC - KGC$ calculates $xpsk_i = d_i + s \cdot h_1^i + H_2(\mathrm{ID}_i, s \cdot X_i) \bmod p$, and returns the verification parameter D_i and the partial private key $xpsk_i$ back to U_i.

3) KeyGen

When the user U_i receives the partial private key $xpsk_i$ generated by the $SC - KGC$, the user generates their public-private key pair (pk_i, sk_i) for subsequent SignCrypt.

i) U_i verifies whether the equality $xpsk_iP = D_i + H_1(\mathrm{ID}_i, X_i + D_i, P_{pub}) \cdot P_{pub} + H_2(\mathrm{ID}_i, x_i \cdot P_{pub})P$ holds. If the verification is successful, calculates $y_i = xpsk_i - H_2(\mathrm{ID}_i, x_i \cdot P_{pub})$ and private key $sk_i = x_i + y_i$. Otherwise, U_i rejects the response D_i and $xpsk_i$ from $SC - KGC$.

ii) U_i computes and exposes the public key $pk_i = X_i + D_i$.

4) SignCrypt

It is assumed that the ciphertext sender is the user U_S, with an identity of ID_S. He needs to send the ciphertext messages ID_S to n ciphertext receivers U_{R_i}, whose identity set is $\mathrm{ID}_R = \{\mathrm{ID}_{R_1}, \mathrm{ID}_{R_2}, \cdots, \mathrm{ID}_{R_n}\}$. The ciphertext received by each receiver is different. The ciphertext ϕ is related to the identity ID_{R_i} of the ciphertext receiver and the plaintext message m_i to be signed. It is worth noting that the set of plaintext messages is $m = \{m_1, m_2, \cdots, m_n\}$. The ciphertext sender U_S randomly selects $\alpha_S \in Z_p^*$ and calculates $A_S = \alpha_S P$. Then, According to the following steps, signcryption generates ciphertext information corresponding to the receiver.

i) The position of the ciphertext in the domain is calculated based on the user's identity ID. Calculate the index location $J_{R_i} = F_{index}^n(\mathrm{ID}_{R_i})$, where $1 \leq J_{R_i} \leq n$.

ii) Calculate equations $h_1^{R_i} = H_1(\mathrm{ID}_{R_i}, pk_{R_i}, P_{pub})$, $W_{R_i} = \alpha_S(P_{pub}h_1^{R_i} + pk_{R_i})$ and $K_{R_i} = H_3(\mathrm{ID}_{R_i}, W_{R_i})$.

iii) Calculate the equation $\tau_{R_i} = H_4(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m_i, A_S, CT_1)$ and $\mu_{R_i} = H_5(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m_i, W_{R_i}, \tau_{R_i}, CT_1)$, where CT_1 is current timestamp,accompanied by time verification.

iv) Calculate the equation $v_{R_i} = \tau_{R_i} \cdot sk_S + \mu_{R_i} \cdot \alpha_S$. The v_{R_i} is stored at the J_{R_i} position of the set J_{R_i}, where $V[J_{R_i}] \leftarrow v_{R_i}$.

v) Calculate the equation $c_{R_i} = (m_{R_i} \parallel \mathrm{ID}_S) \oplus K_{R_i}$. The c_{R_i} is stored at the J_{R_i} position of the set J_{R_i}, where $C[J_{R_i}] \leftarrow c_{R_i}$.

5) UnSignCrypt

After receiving the ciphertext message $\phi = \{A_S, C, V, CT_1\}$, the ciphertext receiver U_{R_i} obtains the system parameters $params = \{G, p, P, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, F_{index}^n()\}$ from the smart contract. Then, receiver U_{R_i} calculates $J_{R_i} = F_{index}^n(\mathrm{ID}_{R_i})$ and utilizes this value to determine the storage location of the corresponding ciphertext, and extracts $c_{J_{R_i}}$ and $v_{J_{R_i}}$ from the sets C and V respectively, where $1 \leq J_{R_i} \leq n$. Finally, decrypt the ciphertext message according to the following steps.

i) Check if there is an inequality $CTR_{\mathrm{ID}_{R_i}} < n$, where $CTR_{\mathrm{ID}_{R_i}}$ is a counter maintained by $CTR_{\mathrm{ID}_{R_i}}$. If the above relationship equation exists, then U_{R_i} continues to judge the freshness of the received ciphertext message $\phi = \{\mathrm{ID}_S, A_S, C, V, CT_1\}$. If $\phi = \{\mathrm{ID}_S, A_S, C, V, CT_1\}$, it indicates that $\phi = \{\mathrm{ID}_S, A_S, C, V, CT_1\}$ does not expire and can decrypt the ciphertext message, where CT_2 is the current timestamp and Δt is the predefined maximum transmission delay. Otherwise, U_{R_i} discards the received ciphertext.

ii) Calculate $W'_{R_i} = sk_{R_i} \cdot A_S$ and $K'_{R_i} = H_3(\mathrm{ID}_{R_i}, W'_{R_i})$.

iii) Calculate $m'_{R_i} \parallel \mathrm{ID}_S = c_{J_{R_i}} \oplus K'_{R_i}$.

iv) Calculate $\tau'_{R_i} = H_4(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, A_S, CT_1)$ and $\mu'_{R_i} = H_5(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, W'_{R_i}, \tau'_{R_i}, CT_1)$.

v) Verify whether the equation $v_{J_{R_i}} P = \tau'_{R_i} \cdot (pk_S + P_{pub} h_1^S) + \mu'_{R_i} \cdot A_S$ exists. If not, it indicates that the message m'_{R_i} is illegal and refuses to receive.

4.3. Correctness analysis

Theorem 1. The message decrypted by the receiver is a correct and valid message.

Proof

1) Decryption correctness

Because there exists the following equation.

$$\begin{aligned}
W'_{R_i} &= sk_{R_i} \cdot A_S = (x_{R_i} + y_{R_i}) \cdot \alpha_S P \\
&= \alpha_S (x_{R_i} + (xpsk_{R_i} - H_2(\mathrm{ID}_{R_i}, x_{R_i} \cdot P_{pub})))P \\
&= \alpha_S (x_{R_i} + (d_{R_i} + s \cdot h_1^{R_i} + H_2(\mathrm{ID}_{R_i}, s \cdot X_{R_i}) - H_2(\mathrm{ID}_{R_i}, x_{R_i} \cdot P_{pub})))P \\
&= \alpha_S (x_{R_i} + (d_{R_i} + s \cdot h_1^{R_i}))P \\
&= \alpha_S (X_{R_i} + D_{R_i} + P_{pub} \cdot h_1^{R_i}) \\
&= \alpha_S (pk_{R_i} + P_{pub} h_1^{R_i}) \\
&= W_{R_i}
\end{aligned}$$

Note that in the above formula, $h_1^{R_i} = H_1(\mathrm{ID}_{R_i}, pk_{R_i}, P_{pub})$. Therefore $W'_{R_i} = W_{R_i}$ holds, then $K'_{R_i} = H_3(\mathrm{ID}_{R_i}, W'_{R_i}) = H_3(\mathrm{ID}_{R_i}, W_{R_i}) = K_{R_i}$ holds. Eventually, we can get $K'_{R_i} = H_3(\mathrm{ID}_{R_i}, W'_{R_i}) = H_3(\mathrm{ID}_{R_i}, W_{R_i}) = K_{R_i}$.

2) Signature verification correctness

Because of $\tau'_{R_i} = H_4(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, A_S, CT_1) = \tau_{R_i}$ and $\mu'_{R_i} = H_5(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, W'_{R_i}, \tau'_{R_i}, CT_1) = \mu_{R_i}$, the following equation holds.

$$\begin{aligned}
v'_{R_i} P &= (\tau'_{R_i} \cdot sk_S + \mu'_{R_i} \cdot \alpha_S)P \\
&= \tau'_{R_i} \cdot (x_S + y_S)P + \mu'_{R_i} \cdot \alpha_S P \\
&= \tau'_{R_i} \cdot (x_S + d_S + s \cdot h_1^S)P + \mu'_{R_i} \cdot \alpha_S P \\
&= \tau'_{R_i} \cdot (X_S + D_S + P_{pub} h_1^S) + \mu'_{R_i} \cdot A_S \\
&= \tau'_{R_i} \cdot (pk_S + P_{pub} h_1^S) + \mu'_{R_i} \cdot A_S
\end{aligned}$$

5. Security proof

5.1. Confidentiality

Theorem 2. The confidentiality of Type I adversary A_{1-1} under chosen ciphertext attack. Under the random oracle model, if there is an adversary A_{1-2} who wins the game in Definition 1 with a non-negligible probability ε in a polynomial time frame with at most q_S signed secret queries and

q_{SK} private key generation queries. Then, there exist a challenger $\mathcal{T}$ can solve CDH problem with probability $Adv^{T}_{A_{1-1}}(k) \geq (1 - \frac{q_{SK}}{2^k})\frac{\varepsilon}{\mathrm{e}(q_S+1)}$ in a polynomial time. Note that the probability $Adv^{T}_{A_{1-1}}(k) \geq (1 - \frac{q_{SK}}{2^k})\frac{\varepsilon}{\mathrm{e}(q_S+1)}$ is non-negligible, where e is the base of the natural logarithm, k is security parameter.

Proof Assume that the algorithm $\mathcal{R}$ can solve the CDH problem, it works by solving with the input challenge instances (P, aP, bP) to obtain $abP \in G$, where $a, b \in Z_p^*$ and a, b are unknown. $\mathcal{R}$ selects A_{1-1} as a sub-algorithm and acts as challenger $\mathcal{T}$ in the Definition 1 game. ID_S and ID_R in the process of signing and declassifying queries correspond to the identity of the ciphertext ender and the ciphertext receiver, respectively.

1) Init phase

$\mathcal{T}$ performs system initialization, sends $params = \{G, p, P, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, F^n_{index}()\}$ to A_{1-1}, where $P_{pub} = aP$ ($a \in Z_p^*$ is master key and cannot be obtained by $\mathcal{T}$). $\mathcal{T}$ maintains lists $L_0, L_1, L_2, L_3, L_4, L_5, L_{sk}, L_{pk}$ that are initialized to empty, they are used to record the query results for $H_0, H_1, H_2, H_3, H_4, H_5$ private key generation and public key generation respectively.

2) Query phase

i) H_4 query: After the challenger $\mathcal{T}$ receives query $H_4(\mathrm{ID}_i, \mathrm{ID}_j, m_j, A_i, CT_1)$ from A_{1-1}, if $(\mathrm{ID}_i, \mathrm{ID}_j, m_j, A_i, CT_1, \tau_j) \in L_4$, then $\mathcal{T}$ sends τ_j to A_{1-1}. Otherwise, $\mathcal{T}$ randomly chooses $\tau_j \in Z_p^*$ and sends it to A_{1-1}.

ii) H_5 query: After the challenger $\mathcal{T}$ receives query $H_5(\mathrm{ID}_i, \mathrm{ID}_j, m_j, W_j, \tau_j, CT_1)$ from A_{1-1}, if $(\mathrm{ID}_i, \mathrm{ID}_j, m_j, W_j, \tau_j, CT_1, \mu_j) \in L_5$, then $\mathcal{T}$ sends μ_j to A_{1-1} and adds $(\mathrm{ID}_i, \mathrm{ID}_j, m_j, W_j, \tau_j, CT_1, \mu_j)$ to L_5, where $(\mathrm{ID}_i, \mathrm{ID}_j, m_j, W_j, \tau_j, CT_1, \mu_j) \notin L_5$.

iii) H_3 query: After the challenger $\mathcal{T}$ receives query $H_3(\mathrm{ID}_j, W_j)$ from A_{1-1}, if $(\mathrm{ID}_j, W_j, K_j) \in L_3$, then $\mathcal{T}$ sends K_j to A_{1-1}. Otherwise, $\mathcal{T}$ randomly chooses K_j and sends it to A_{1-1}, then adds $(\mathrm{ID}_j, W_j, K_j)$ to L_3, where $(\mathrm{ID}_j, W_j, K_j) \notin L_3$.

iv) Public key generation query: After the challenger $\mathcal{T}$ receives public key generation query, if $(\mathrm{ID}_i, X_i + D_i, v_i) \in L_{pk}$, then $\mathcal{T}$ sends $pk_i = X_i + D_i$ to A_{1-1}, where $v_i \in \{0, 1\}$. Otherwise, $\mathcal{T}$ randomly chooses $v_i \in \{0, 1\}$ and stipulates the existence of $\Pr[v_i = 1] = \delta = \frac{1}{q_S+1}$, then processed according to the value of v_i.

If $v_i = 0$, the challenger $\mathcal{T}$ randomly chooses $h_1, x_i, y_i \in Z_p^*$, calculates $X_i = x_i P$, $sk_i = x_i + y_i$, $D_i = y_i P - h_1 P_{pub}$ and $pk_i = X_i + D_i$. $\mathcal{T}$ sends pk_i to A_{1-1} and adds $(\mathrm{ID}_i, pk_i, v_i)$ to L_{pk}, adds $(\mathrm{ID}_i, sk_i, v_i)$ to L_{sk}, adds $(\mathrm{ID}_i, pk_i, P_{pub}, h_1)$ to L_1.

If $v_i = 1$, $\mathcal{T}$ knows two random numbers ϑ_1 and ϑ_2, calculates $X_i = \vartheta_1 P$, $D_i = \vartheta_2 P$ and $pk_i = X_i + D_i$. $\mathcal{T}$ sends pk_i to A_{1-1} and adds $(\mathrm{ID}_i, pk_i, v_i)$ to L_{pk}. $\mathcal{T}$ randomly chooses $h_1 \in Z_p^*$ and adds $(\mathrm{ID}_i, pk_i, P_{pub}, h_1)$ to L_1 , where $h_1 \notin L_1$.

v) H_0 query: After the challenger $\mathcal{T}$ receives query $H_0(x_i P_{pub}(sX_i))$ from A_{1-1}, if $(x_i P_{pub}(sX_i), h_0) \in L_0$, then sends h_0 to A_{1-1}. Otherwise, $\mathcal{T}$ randomly chooses $h_0 \in Z_p^*$ and sends it to A_{1-1}, $\mathcal{T}$ adds $(x_i P_{pub}(sX_i), h_0)$ to L_0, where $(x_i P_{pub}(sX_i), h_0) \notin L_0$.

vi) H_1 query: After the challenger $\mathcal{T}$ receives query $H_1(\mathrm{ID}_i, pk_i(X_i + D_i), P_{pub})$, if $(\mathrm{ID}_i, pk_i(X_i + D_i), P_{pub}, h_1) \in L_1$, then sends h_1 to A_{1-1}. Otherwise, $\mathcal{T}$ sends h_1 to A_{1-1} after generates public key query of ID_i.

vii) H_2 query: After the challenger $\mathcal{T}$ receives query $H_2(\mathrm{ID}_i, sX_i(x_i P_{pub}))$ from A_{1-1}, if $(\mathrm{ID}_i, sX_i(x_i P_{pub}), h_2) \in L_2$, then sends h_2 to A_{1-1}. Otherwise, $\mathcal{T}$ randomly chooses $h_2 \in Z_p^*$ and sends it to A_{1-1}, $\mathcal{T}$ adds $(\mathrm{ID}_i, sX_i(x_i P_{pub}), h_2)$ to L_2, where $(\mathrm{ID}_i, sX_i(x_i P_{pub}), h_2) \notin L_2$.

viii) Private key generation query: After the challenger $\mathcal{T}$ receives private key generation query from A_{1-1}, if $(\mathrm{ID}_i, sk_i) \in L_{sk}$, then sends sk_i to A_{1-1}. Otherwise, $\mathcal{T}$ will get the information $(\mathrm{ID}_i, pk_i, v_i)$ about the public key of ID_i after generates public key generation query of ID_i . If $v_i = 0$, then $\mathcal{T}$ has generates private key sk_i of ID_i. $\mathcal{T}$ searches $(\mathrm{ID}_i, sk_i, v_i)$ from L_{sk} and sends sk_i to A_{1-1}. If $v_i = 1$, $\mathcal{T}$ ends the game.

ix) Public key replacement query: The adversary A_{1-1} can generate the public key $pk_i{}'$ of the ID at any time and replace the original legal public key pk_i with it.

x) Signcrypt query: After the challenger $\mathcal{T}$ receives signcrypt query $(\mathrm{ID}_S, \mathrm{ID}_R{=}\{\mathrm{ID}_{R_1}, \mathrm{ID}_{R_2}, \cdots, \mathrm{ID}_{R_\rho}\}, M{=}\{m_1{,}m_2, \cdots{,}m_\rho\})$ from A_{1-1}, $\mathcal{T}$ searches $(\mathrm{ID}_S, pk_S, v_S)$ from ID_S. If $v_S = 1$, $\mathcal{T}$ ends the game. If $v_S = 0$, $\mathcal{T}$ has generated private key sk_S of ID_S. $\mathcal{T}$ executes public key query of $\mathrm{ID}_{R_i}(1 \leq i \leq n)$ to obtain $(\mathrm{ID}_{R_i}, pk_{R_i})$, then runs signcrypt algorithm to generate ciphertext $(\mathrm{ID}_{R_i}, C = \{c_{R_1}, c_{R_2}, \cdots, c_{R_n}\})$ and sends it to A_{1-1}.

xi) Unsigncrypt query: After the challenger $\mathcal{T}$ receives $\mathrm{ID}_S, \mathrm{ID}_{R_i}$ from A_{1-1} and ciphertext $c_{J_{R_i}}$ unsigncrypt query, executes public key query of ID_S and proceed as follows.

If $(\mathrm{ID}_S, pk_S, v_S) \in L_{pk}$ and $v_S = 0$, $\mathcal{T}$ executes private key query of ID_{R_i} to obtain private key sk_{R_i}. $\mathcal{T}$ runs unsigncrypt algorithm to get $m'_{R_i} \parallel \mathrm{ID}_S = c_{J_{R_i}} \oplus K'_{R_i}$, $\tau'_{R_i} = H_4(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, A_S, CT_1)$ and $\mu'_{R_i} = H_5(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, W'_{R_i}, \tau'_{R_i}, CT_1)$. If $v_S = 1$, $\mathcal{T}$ executes H_3, H_4 and H_5 queries to obtain $(\mathrm{ID}_{R_i}, W_{R_i}, K_{R_i})$, $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, A_S, CT_1, \tau'_{R_i})$ and $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, W'_{R_i}, \tau'_{R_i}, CT_1, \mu'_{R_i})$ from list L_3, L_4, L_5. Then $\mathcal{T}$ calculates $m'_{R_i} \parallel \mathrm{ID}_S = c_{J_{R_i}} \oplus K'_{R_i}$. If there exists $v'_{R_i}P = \tau'_{R_i} \cdot (pk_S + P_{pub}h_1^S) + \mu'_{R_i} \cdot A_S$, then sends m'_{R_i} to A_{1-1}. Otherwise, the ciphertext is incorrect, $\mathcal{T}$ ends the game.

If $(\mathrm{ID}_S, pk_S, v_S) \notin L_{pk}$, then the public key of ID_S has been replaced before.

Use ID_{R_i} and ID_S as index values to obtain $(\mathrm{ID}_S, pk'_S, P_{pub}, h'_1{}^S) \in L_1$, $(\mathrm{ID}_{R_i}, W_{R_i}, K_{R_i}) \in L_3$, $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, A_S, CT_1, \tau'_{R_i}) \in L_4$ and $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, W'_{R_i}, \tau'_{R_i}, CT_1, \mu'_{R_i}) \in L_5$ from L_1, L_2, L_3 and L_5. Then $\mathcal{T}$ calculates $m'_{R_i} \parallel \mathrm{ID}_S = c_{J_{R_i}} \oplus K'_{R_i}$. If there exists $v'_{R_i}P = \tau'_{R_i} \cdot (pk_S + P_{pub}h_1^S) + \mu'_{R_i} \cdot A_S$, then $\mathcal{T}$ sends m'_{R_i} to A_{1-1}. Otherwise, the ciphertext is incorrect, $\mathcal{T}$ ends the game.

3) Challenge phase

The attacker A_{1-1} selects the identity $(\mathrm{ID}_S, \mathrm{ID}_{R_i})$ of the ciphertext sender and the ciphertext receiver and two plaintexts (m_0, m_1) of the same length. Then, he conveys information about the challenge to the challenger $\mathcal{T}$. When $\mathcal{T}$ receives the challenge information from A_{1-1}, he learns the identity information $(\mathrm{ID}_S, \mathrm{ID}_{R_i})$ and plaintext information (m_0, m_1). First, the public key generation query about ID_S is executed to obtain the corresponding $(\mathrm{ID}_S, pk_S, v_S)$ from L_{pk}. Then, according to the value v_S of the query, different operations are performed. If the result shows $v_S = 0$, game over. Otherwise, $\mathcal{T}$ randomly selects $\tau \in \{0, 1\}$ and $b, v^*_{R_i}, \tau^*_{R_i}, \mu^*_{R_i} \in Z^*_p$. Then, $\mathcal{T}$ calculates $A_S = bP$ and $v^*_{R_i} = \tau^*_{R_i} \cdot sk_S + \mu^*_{R_i} \cdot \alpha_S$. $\mathcal{T}$ selects $W'_{R_i} \in G$ and finds K'_{R_i} from L_3. At last, $\mathcal{T}$ calculates $c_{R_i} = (m_\tau \parallel \mathrm{ID}_S) \oplus K'_{R_i}$ and sends ciphertext information $\{c_{R_i}, pk_{R_i}\}$ to the attacker A_{1-1}.

4) The guessing phase

After receiving the message $\{c_{R_i}, pk_{R_i}\}$ from the challenger $\mathcal{T}$, the attacker A_{1-1} guesses the value of $\tau' \in \{0, 1\}$. If $\tau' = \tau$, the solution $abP = \frac{1}{h_1^S}(\frac{v^*_{R_i} - \mu^*_{R_i} b}{\tau^*_{R_i}} - (\vartheta_1 + \vartheta_2))A_S$ of the CDH problem can be output. Otherwise, $\mathcal{T}$ does not solve the CDH problem. Then, $\mathcal{T}$ simulates the attack in real scene. If the $\mathcal{T}$ never ends the game from beginning to end, and the attacker A_{1-1} breaks the confidentiality of the scheme in this paper with an unnegligible advantage ε, the $\mathcal{T}$ outputs an effective solution to the CDH problem. Assuming that the attacker A_{1-1} never makes a private key generation query about the identity ID_S to be challenged, it is represented by event E. That is, $\Pr(E) = 1 - \frac{q_{\mathrm{SK}}}{2^k}$ exists. The signcryption phase of the algorithm is not ended by the event E'. That is, $\Pr(E') = (1 - \delta)^{q_S}$ exists. The challenge phase algorithm is not ended with event E''. That is, $\Pr(E'') = \delta$ exists. Then, the whole simulation process ends normally, and the probability of non-stop is at least $\Pr(E \wedge E' \wedge E'') = (1 - \frac{q_{\mathrm{SK}}}{2^k})(1-\delta)^{q_\mathrm{S}}\delta$. Because of $\delta = \frac{1}{q_\mathrm{S}+1}$, when q_S is large enough, there exists $(1-\delta)^{q_\mathrm{S}} \to \mathrm{e}^{-1}$. Where e denotes the base of the natural logarithm, So there is $\Pr(E \wedge E' \wedge E'') \geq (1 - \frac{q_{SK}}{2^k})\frac{\varepsilon}{\mathrm{e}(q_\mathrm{S}+1)}$.

From what has been discussed above, $\mathcal{T}$ never terminates in the whole simulation process, and A_{1-1} can break the confidentiality of the scheme in polynomial time with a non-negligible probability ε. Then $\mathcal{T}$ can solve the CDH problem with probability $Adv^{T}_{A_{1-1}}(k) \geq (1 - \frac{q_{SK}}{2^k})\frac{\varepsilon}{\mathrm{e}(q_\mathrm{S}+1)}$.

Theorem 3. Type Π Attacker A_{2-1} selects the confidentiality under ciphertext attack.

In the random oracle model, there is an attacker A_{2-1} in the case of up to q_S signcryption queries and q_{SK} private key generation queries. He can win the game in Definition 2 in polynomial time under a non-negligible probability ε. If the above conditions hold, there is a challenger $\mathcal{T}$ that can successfully solve the CDH problem in polynomial time with the probability $Adv^{T}_{A_{2-1}}(k) \geq (1 - \frac{q_{SK}}{2^k})\frac{\varepsilon}{\mathrm{e}(q_\mathrm{S}+1)}$. Note that the probability $Adv^{T}_{A_{2-1}}(k)$ can not be ignored. The proof idea is the same as Theorem 2, which is no longer explained.

5.2. Unforgeability

Theorem 4. Type I attacker A_{1-2} selects the unforgeability under message attack.

In the random oracle model, there is an attacker A_{1-2}. In the case of at most q_S signature queries, the game in Definition 3 is won in polynomial time with a non-negligible probability ε. If the above situation exists, there is a challenger $\mathcal{T}$ that can successfully solve the DL problem in polynomial time with the probability of $Adv^{T}_{A_{1-2}}(k) \geq \frac{\varepsilon}{\mathrm{e}(q_\mathrm{S}+1)}$. Note that the probability $Adv^{T}_{A_{1-2}}(k)$ can not be ignored. Where e is the base of the natural logarithm, and k is the safety parameter.

Assuming that the solver of the DL problem is an algorithm, its work is to use an input challenge instance.

Proof Assuming that the solver of the DL problem is algorithm $\mathcal{R}$, its work is to obtain the value of a by solving the input challenge instance (P, aP), where $a \in Z_p^*$ and unknown. $\mathcal{R}$ selects A_{1-2} as a subroutine and act as a challenger $\mathcal{T}$ to the game in Definition 3.

1) Init phase

$\mathcal{T}$ performs system initialization, sends $params = \{G, p, P, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, F^n_{index}()\}$ to A_{1-2}, where $P_{pub} = aP$ ($a \in Z_p^*$ is master key and cannot be obtained by $\mathcal{T}$). $\mathcal{T}$ maintains lists $L_0, L_1, L_2, L_3, L_4, L_5, L_{sk}, L_{pk}$ that are initialized to empty, they are used to record the query results for $H_0, H_1, H_2, H_3, H_4, H_5$ private key generation and public key generation respectively.

2) Query phase

$H_0, H_1, H_2, H_3, H_4, H_5$ query, public key generation query, private key generation query and replacement public key query are the same as the query process in Theorem 2.

i) Signature inquiry: When the challenger $\mathcal{T}$ receives a signcryption query (ID_S, ID_R=$\{\mathrm{ID}_{R_1}, \mathrm{ID}_{R_2}, \cdots, \mathrm{ID}_{R_\rho}\}$, M=$\{m_1, m_2, \cdots, m_\rho\}$) from A_{1-2}, the $(\mathrm{ID}_S, pk_S, v_S)$ corresponding to ID_S is viewed from L_{pk}. If there is $v_S = 1$, then challenger $\mathcal{T}$ ends the game. If there is $v_S = 0$, then $\mathcal{T}$ has generated the private key sk_S of ID_S. First, $\mathcal{T}$ performs a public key query on $\mathrm{ID}_{R_i}(i \in [1, n])$ to obtain $(\mathrm{ID}_{R_i}, pk_{R_i})$. In the second place, $\mathcal{T}$ performs a signcryption algorithm to generate ciphertext $(\mathrm{ID}_{R_i}, C = \{c_{R_1}, c_{R_2}, \cdots, c_{Rn}\})$ and sends it to A_{1-2}.

ii) Signature verification inquiry: When the challenger $\mathcal{T}$ receives the signature verification query of ID_S, ID_{R_i}, ciphertext c_{R_i} and message m_{R_i} from A_{1-2}, the public key generation query is performed on ID_S to obtain $(\mathrm{ID}_S, pk_S, v_S)$. Then, the following operations are performed according to the value of v_S.

If there is $(\mathrm{ID}_S, pk_S, v_S) \in L_{pk}$ and $v_S = 0$, $\mathcal{T}$ performs the decryption algorithm and sends the final result to A_{1-2}.

If there is $(\mathrm{ID}_S, pk_S, v_S) \in L_{pk}$ and $v_S = 1$, $\mathcal{T}$ performs a private key generation query on ID_{R_i} to obtain the corresponding private key sk_{R_i}, and performs a public key generation query on ID_S to obtain the corresponding public key $(\mathrm{ID}_S, pk_S, v_S)$. Furthermore, H_1 query operation is performed to get $(\mathrm{ID}_S, pk_S, P_{pub}, h_1^S)$ and $(\mathrm{ID}_{R_i}, pk_{R_i}, P_{pub}, h_1^{R_i})$ from the list L_1. Then $\mathcal{T}$ performs H_3, H_4 and H_5 queries to get $(\mathrm{ID}_{R_i}, W_{R_i}, K_{R_i})$, $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, A_S, \mathrm{s}, CT_1, \tau'_{R_i})$ and $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, W'_{R_i}, \tau'_{R_i}, CT_1, \mu'_{R_i})$ from the list L_3, L_4, L_5 respectively. If $v'_{R_i}P = \tau'_{R_i} \cdot (pk_S + P_{pub}h_1^S) + \mu'_{R_i} \cdot A_S$, $\mathcal{T}$ replies the plaintext m'_{R_i} to A_{1-2}. Otherwise, $\mathcal{T}$ indicates that the ciphertext message is incorrect and ends the game.

If there is $(\mathrm{ID}_S, pk_S, v_S) \notin L_{pk}$, it is shown that the public key of ID_S has been replaced before. At the moment, ID_{R_i} and ID_S are used as index values. $\mathcal{T}$ obtains $(\mathrm{ID}_S, pk'_S, P_{pub}, h_1^S)$, $(\mathrm{ID}_{R_i}, pk_{R_i}, P_{pub}, h_1^{R_i})$, $(\mathrm{ID}_S, pk'_S, v_S)$ and private key sk_{R_i} from L_1, L_{pk} and L_{sk}. Then $\mathcal{T}$ performs H_3, H_4 and H_5 queries and gets $(\mathrm{ID}_{R_i}, W_{R_i}, K_{R_i})$, $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, A_S, CT_1, \tau'_{R_i})$ and $(\mathrm{ID}_S, \mathrm{ID}_{R_i}, m'_i, W'_{R_i}, \tau'_{R_i}, CT_1, \mu'_{R_i})$ from Lists L_3, L_4, L_5 respectively. If $v'_{R_i}P = \tau'_{R_i} \cdot (pk_S + P_{pub}h_1^S) + \mu'_{R_i} \cdot A_S$ is verified, $\mathcal{T}$ will reply plaintext message m'_{R_i} to A_{1-2}. Otherwise, it indicates that the ciphertext message is incorrect, and $\mathcal{T}$ ends th game.

3) Forgery phase

After performing a limited number of the above queries, the attacker A_{1-2} forges the ciphertext sender ID_S, and outputs about the ciphertext receiver ID_R=$\{\mathrm{ID}_{R_1}$,$\mathrm{ID}_{R_2}, \cdots$,$\mathrm{ID}_{R_\rho}\}$ and a forged signature of the plaintext message $M = \{m_1, m_2, \cdots, m_\rho\}$. A_{1-2} randomly selectes $\alpha_S \in Z_p^*$, and get $A_S = \alpha_S P$, and then calculates the $h_1^{R_i} = H_1(\mathrm{ID}_{R_i}, pk_{R_i}, P_{pub})$, $W_{R_i} = \alpha_S(pk_{R_i} + P_{pub}h_1^{R_i})$, $K_{R_i} = H_3(\mathrm{ID}_{R_i}, W_{R_i})$ and $c_{R_i} = (m_{R_i} \parallel \mathrm{ID}_S) \oplus K_{R_i}$.

If the forgery attack of attacker A_{1-2} is successful, then the challenger $\mathcal{T}$ enters the ID_S query the list L_{pk} to get $(\mathrm{ID}_S, pk_S, v_S)$. At the end of the above process, $\mathcal{T}$ calculates $h_1^S = H_1(\mathrm{ID}_S, pk_S, P_{pub})$. If there is $v_S = 1$, then $\mathcal{T}$ obtains the solution of DL problem $a = (h_1^S\tau_{R_i}^*)^{-1}(\upsilon_{R_i}^* - \mu_{R_i}^*\alpha_S - \tau_{R_i}^*(x_S + d_S))$. If there is $v_S = 0$, then $\mathcal{T}$ ends the game. It is shown that $\mathcal{T}$ fails to solve the DL problem.

It is assumed that the challenger $\mathcal{T}$ never ends the game in the signature query phase and is represented by the event E. It indicates the existence of $\Pr(E) = (1-\delta)^{q\mathrm{s}}$. That is to say, the probability of not ending the game throughout the inquiry phase is $(1-\delta)^{qs}$ and the probability of not ending the game in the forgery phase is δ. Therefore, the probability of never ending from beginning to end is at least $(1-\delta)^{q\mathrm{s}}\delta$ in the whole game process. Because of $\delta = \frac{1}{q\mathrm{s}+1}$, when q_{S} is large enough, there exists $(1-\delta)^{qs} \rightarrow \mathrm{e}^{-1}$. Accordingly, the probability that the game will not end until the final step is completed is at least $\frac{1}{\mathrm{e}(q_{\mathrm{S}}+1)}$.

In summary, if $\mathcal{T}$ never terminates in the whole simulation process. A_{1-2} can break the unforgeability of the scheme in polynomial time with a non-negligible probability ε. It shows that $\mathcal{T}$ can solve the DL problem with probability $Adv_{A_{1-2}}^T(k) \geq \frac{\varepsilon}{\mathrm{e}(q_{\mathrm{S}}+1)}$.

Theorem 5. Type Π Attacker A_{2-2} chooses unforgeability under message attack.

In the random oracle model, an attacker A_{2-2} can obtain the victory of the game in Definition 4 in polynomial time with a non-negligible probability ε in the case of at most q_S signature queries. It is pointed out that there is a challenger $\mathcal{T}$ who can successfully solve the DL problem in polynomial time with the probability of $Adv_{A_{2-2}}^T(k) \geq \frac{\varepsilon}{\mathrm{e}(q_{\mathrm{S}}+1)}$. Note that the probability $Adv_{A_{2-2}}^T(k)$ cannot be ignored, where e is the base of the natural logarithm and k is the security parameter.

The proof idea is the same as Theorem 4, which is no longer explained.

6. Programme analysis

6.1. Safety analysis

1) KGC leakage attack

In traditional certificateless signcryption schemes, the presence of a centralized KGC poses inherent vulnerabilities. If the KGC is illicitly compromised, it can have severe and far-reaching consequences. To overcome this limitation, the proposed solution introduces an alternative approach by leveraging smart contracts on the Ethereum blockchain to replace the original centralized and open KGC. This novel design mitigates the risks associated with unauthorized control over the KGC. By employing smart contracts on the Ethereum blockchain, the proposed solution offers enhanced security guarantees. An attacker attempting to manipulate the entire blockchain system would need to carry out a 51% attack, which requires a significant amount of computational power. Statistical data reveals that the overall hash power on the Ethereum blockchain is approximately 314.96 TH/s, making it highly implausible for an attacker to acquire secret values or successfully compromise the blockchain system. Consequently, the proposed solution presented in this article effectively safeguards against attacks on the KGC, significantly improving the overall security of the certificateless signcryption scheme.

2) Internal privilege attack

It is assumed that the deployer of the smart contract acts as an attacker of the scheme and attempts to forge the legitimate signature of the user. However, due to the immutability of the blockchain, once the smart contract is deployed, it can only be destroyed. In addition, there is no way to modify the deployed contract code. Even if the deployer inside the smart contract knows the system master key of the scheme, it still cannot change the deployed smart contract. Therefore, this scheme can successfully defend against internal privilege attacks.

3) Replay attack

The replay attack is most likely to occur during the decryption phase. Assuming that the attacker continuously eavesdrops on the message communication between the ciphertext sender and the ciphertext receiver, all the signcryption ciphertexts generated or transmitted by the ciphertext sender may be intercepted by the attacker. If the current timestamp is not used, the attacker can replay the intercepted signcryption ciphertext to other unwanted ciphertext receivers for false identity decryption or other operations. Therefore, the improved scheme adds CT_1 to the $\tau_{R_i} = H_2(\mathrm{ID}_S, \mathrm{ID}_{R_i}, c_{R_i}, A_S, X_{R_i}, CT_1)$ of the partial ciphertext V_{R_i}. When the ciphertext receiver receives the ciphertext, it is necessary to check the timeliness of CT_1. Therefore, this scheme can resist potential replay attacks.

4) DDoS attack

In order to resist potential DDoS attacks, two counters $CTR^i{}_{Addr_{ID}}$ and $CTR_{\mathrm{ID}_{R_i}}$ are added to the improved scheme. Moreover, each user using Ethereum has an address for trading. The counter $CTR^i{}_{Addr_{ID}}$ is used to check the frequency of the ciphertext sender. If the frequency of the user interface exceeds a predefined threshold, the current user interface will be blocked for some time. In the process of unsigncryption, $CTR_{\mathrm{ID}_{R_i}}$ plays a similar role as $CTR^i{}_{Addr_{ID}}$. Therefore, this scheme can resist potential DDoS attacks.

5) MITM attack

Most of the existing schemes are designed under the assumption that the communication channel between the user interface and SC-KGC is secure, so the possible MITM attack is ignored. However, in the actual environment, such attacks often occur. In order to prevent attackers from intercepting and tampering with the messages transmitted in the channel, the improved scheme performs XOR processing on the user 's real identity ID. It makes ID anonymized in the channel, unless the attacker can obtain the secret value x_{ID} of the ID user or the system's master key s. In addition, some of the returned private key $xpsk_i$ is protected by the correlation operation, where $xpsk_i = d_i + s \cdot h_1^i + H_2(\mathrm{ID}_i, s \cdot X_i)$. As a consequence, the proposed scheme can resist potential MITM attacks.

6) Key escrow security

In this paper, the KGC served by the smart contract SC-KGC is responsible for generating the user's partial key. The user inputs (PID_i, X_i) to call the partial key generation function, and SC-KGC generates the encrypted partial private key $xpsk_i$ and its corresponding partial public key D_i. If SC-KGC wants to get the user's complete private key $sk_i = x_i + y_i$, the only way is to calculate x_i from X_i. However, it is very difficult, similar to solving the DL problem. That is to say, SC-KGC cannot obtain the user 's complete private key through known information. Moreover, the partial private key $xpsk_i$ generated by SC-KGC is only part of the user's partial private key y_i, and the user still needs to process after receiving $xpsk_i$.

7) Non-repudiability

From Theorems 4 and 5, it can be seen that the scheme proposed in this paper is unforgeable for Type I attacker A_{1-2} selection message attack and Type Π attacker A_{2-2} selection message attack. It shows that third-party attackers cannot forge signcryption ciphertexts. After the ciphertext receiver receives the ciphertext information, the ciphertext is decrypted using the private key information and the sender's public information, and the decrypted message is verified. The ciphertext sender cannot deny the ciphertext he has signed, so the scheme in this paper has the characteristics of non-repudiation. The ciphertext sender cannot deny the ciphertext he has signed, so the scheme in this paper has the characteristics of non-repudiation.

8) Anonymity

In actual scenarios, third-party attackers usually use the captured ciphertext information to correlate with user identity information, and then launch Identity-Based traffic analysis attacks. In the scheme proposed in this paper, when the ciphertext sender generates a signcryption ciphertext, it will randomly generate a number $\alpha_S \in Z_p^*$ and calculate $A_S = \alpha_S P$. On this basis, the scheme will generate signcryption ciphertext. Because different values are randomly generated in each signcryption phase, each batch of ciphertext information is different. Furthermore, the signcryption phase not only processes the message, but also encrypts the identity and message of the ciphertext sender. The attacker cannot directly obtain the real identity of the ciphertext sender. Therefore, this scheme has anonymity.

6.2. Performance analysis

This section will be explained from two parts. First, the security attribute analysis will be introduced. This part includes the theoretical basis of the signcryption scheme, the length of the generated ciphertext and the security. Second, the computational complexity of the scheme is analyzed. Next in importance, we give the computational complexity analysis of the scheme. The comparison of the time overhead obtained by simulation is shown. In order to facilitate comparison, this paper explains the required symbols, which are described in Table 2 below.

Table 2. Description of relevant symbols.

Symbols	Description	Symbols	Description
$\|G_p\|$	Representing the length of the elements in cyclic group $\|G_p\|$ with order p	$\|Z_p^*\|$	Representing the length of the integer elements in $\|Z_p^*\|$
$\|L_M\|$	The length of the plaintext message M	$\|L_{ID}\|$	The length of user identity ID
n	The number of receivers	m	The number of sender's disguised identities
✓	Satisfying the security attribute	×	Not satisfying the security attribute

6.2.1. Security analysis

Currently, cryptographic scheme research primarily focuses on discrete logarithm, elliptic curve, bilinear pairing operations and similar techniques. However, when considering equal security effectiveness, implementing discrete logarithm operations necessitates longer key lengths. As we primarily investigate the multi-message multi-receiver signcryption scheme for broadcasting purposes, which requires shorter ciphertext lengths, the scheme comparison in this section excludes schemes related to discrete logarithms. In addition, when comparing the length of ciphertext, the group order in the bilinear pair $e : G_1 \times G_1 \rightarrow G_2$ is set to p. That is, the total number of the elements in the set is $|G_p|$. The comparison results are shown in Table 3.

Table 3. Comparison of security attributes.

Encryption scheme	Theoretical foundation	Ciphertext length	Non-repudiation	Confidentiality
Scheme [16]	Elliptic curve	$2n\|Z_p^*\| + \|G_p\| + n\|L_M\|$	✓	✓
Scheme [27]	Bilinear pairing	$m\|Z_p^*\|+(m+2n+2)\|G_p\|+\|L_M\| + m\|L_{ID}\|$	✓	✓
Scheme [28]	Bilinear pairing	$(n + 2)\|G_p\| + \|L_M\|$	×	✓
Scheme [29]	Bilinear pairing	$4\|G_p\| + \|Z_p^*\|$	×	✓
Scheme [36]	Elliptic curve	$\|Z_p^*\|+(n+1)\|G_p\|+n\|L_M\|$	✓	✓
CMMSB	Elliptic curve	$n\|Z_p^*\| + \|G_p\| + n\|L_M\| + n\|L_{ID}\| + \|time\|$	✓	✓

Figure 2 shows the change of ciphertext length when the number of receivers changes from 5 to 25. It can be found from the figure that the ciphertext length of the CMMSB scheme proposed in this paper is short and the overhead is small. Specifically, when setting the number of receivers and the number of senders masquerading identities, the ciphertext length of scheme [27] is 24,768 bits. The ciphertext length of the scheme [28] is 12,448 bit. The ciphertext length of scheme [29] is 8013 bits. The ciphertext length of the scheme [36] is 13,888 bit. The ciphertext length of the scheme [16] is 23,104 bits. The ciphertext length of CMMSB scheme is 13,216 bit. Compared with the schemes [28, 29], the ciphertext length of the proposed CMMSB scheme is longer, but schemes [28, 29] are unforgeable. Therefore, the CMMSB scheme proposed in this paper costs the least communication overhead under the premise of ensuring unforgeability and confidentiality.

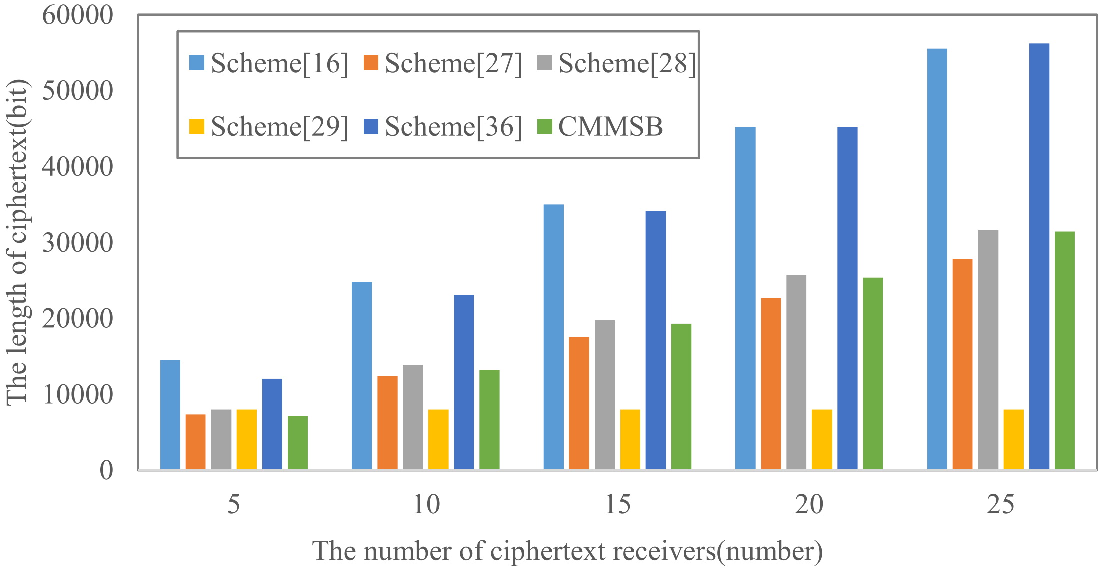

Figure 2. Comparison of ciphertext length.

6.2.2. Computational complexity analysis

The computational complexity analysis mainly analyzes the time overhead of the signcryption scheme in both signcryption and decryption. Table 4 gives some values of related operations in the experimental environment: Intel G630, frequency 2.7 GHz, memory 4 GB (DDR3-1600 MHz), Windows 7 operating system. We used pbc 0.5.14 library and A-type elliptic curve parameters for computations and evaluated the average execution time of cryptographic operations. Due to the low computational overhead of modular addition, modular subtraction and hashing, it is not investigated here. We mostly count bilinear pairing operation, exponential operation, modular multiplication operation, modular inverse operation and time-consuming point addition and point multiplication operation on elliptic curve. The specific comparison results are shown in Table 5 and Figure 3.

Table 4. Description of relevant symbols.

Symbols	Implication	Values	Symbols	Implication	Values
T_m	modular multiplication	T_m	T_{pm}	Elliptic curve point multiplication	$29T_m$
T_e	exponent arithmetic	$240T_m$	T_{pa}	Elliptic curve point addition	$0.12T_m$
T_b	bilinear pairing	$87T_m$	T_i	modular inversion	$11.6T_m$

Table 5. Comparison of computational complexity.

Signcryption scheme	Signcryption	Unsigncryption
Scheme [16]	$(2n+1)T_{pm}+nT_{pa} \approx (58.12n+29)T_m$	$7T_{pm}+4T_{pa} \approx 203.48T_m$
Scheme [27]	$(n+2)T_b+(m+n+3)T_{pm}+(m+n+2)T_{pa} \approx (116.12n+29.12m+261.24)T_m$	$6T_b+(m+2)T_{pm}+(m+2)T_{pa}+T_i \approx (29.12m+591.24)T_m$
Scheme [29]	$(n+2)T_e \approx (240n+480)T_m$	$2T_{pm}+3T_b+(n+3)T_e \approx (240n+1039)T_m$
Scheme [36]	$(3n+1)T_{pm}+3nT_{pa}+T_i \approx (87.36n+40.6)T_m$	$5T_{pm}+6T_{pa} \approx 145.72T_m$
CMMSB	$(2n+1)T_{pm}+nT_{pa} \approx (58.12n+29)T_m$	$5T_{pm}+2T_{pa} \approx 145.24T_m$

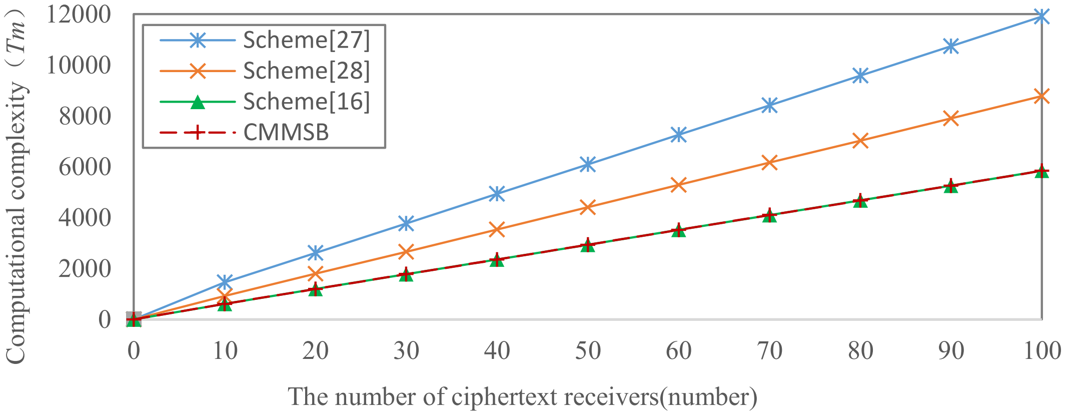

Figure 3. Comparison of signcryption calculation complexity.

Through Table 5 and Figure 3, it can be found that the CMMSB scheme proposed in this paper has the smallest computational complexity and the shortest time in the two stages of signcryption and decryption. Specifically, the computational complexity of the CMMSB scheme and the scheme [16] in the signcryption phase is the same and significantly smaller than the other two schemes. The computational complexity of the CMMSB scheme in the decryption phase is slightly less than that of the scheme [16]. It can be seen that the computational complexity of the scheme [36] and the

CMMSB scheme at this stage is very small. However, due to the large difference in the signcryption stage, the computational complexity and time overhead of the CMMSB scheme are still small in the two stages of signcryption and decryption. On the whole, the CMMSB scheme proposed in this paper is slightly better than other schemes in terms of time overhead compared with the same type of scheme.

7. Conclusions

We propose a certificateless multi-message multi-receiver signcryption scheme to establish secure communication among signcrypted messages. In this scheme, the blockchain-based smart contract algorithm functions as KGC within the CLC system. This approach effectively resolves the first type of forgery attack inherent in traditional KGC systems. To enhance communication concealment, the scheme incorporates encryption technology to obfuscate the user's identity. Furthermore, system-related parameters are disclosed on the blockchain during the specific design, mitigating the risk of potential attacks. By preventing unauthorized modification of system parameters, attackers are effectively thwarted. The performance evaluation of the proposed scheme encompasses various aspects, including the underlying mathematical basis, length of the signcryption ciphertext, unforgeability, confidentiality, as well as the computational costs of signcryption and decryption. The analysis demonstrates that our scheme exhibits significant advantages in terms of communication and time overhead. Furthermore, it guarantees both high security and concealment levels. In future work, we have two plans that address the importance of data security across various industries and the significant advancements in machine learning algorithms in recent years.

1). How to combine machine learning methods and algorithms to enhance data security. Moreover, how to use cryptography technology to ensure the security of communication and storage when protecting the privacy of user identity.

2). It is important to note that our proposed scheme has been proven secure under the random oracle model. Our next step is to extend the research and develop a multi-message multi-receiver scheme based on machine learning algorithms that can be proven secure in the standard model.

These plans will contribute to the advancement of data security by incorporating machine learning techniques and cryptographic protocols. Through academic research and experimentation, we aim to develop practical solutions that address the evolving challenges in ensuring data security and user privacy in various fields.

Use of AI tools declaration

This article does not use the Artificial Intelligence (AI) tools.

Acknowledgments

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

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. S. Basu, S. Tripathy, Secure multicast communication techniques for IoT, in *Security and Fault Tolerance in Internet of Things*, Springer Cham, (2019), 43–59. https://doi.org/10.1007/978-3-030-02807-7_3
2. X. Boyen, Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography, in *Annual International Cryptology Conference*, Berlin, Heidelberg, Springer, (2003), 383–399. https://doi.org/10.1007/978-3-540-45146-4_23
3. L. Pang, H. Li, Q. Pei, Improved multicast key management of Chinese wireless local area network security standard, *IET Commun.*, **6** (2012), 1126–1130. https://doi.org/10.1049/iet-com.2010.0954
4. H. Kashgarani, C. Miller, S. Suresh, A. Zacharias, Exploring Performance of GeoCAT data analysis routines on GPUs, *Supercomputing*, (2022).
5. H. Kashgarani, L. Kotthoff, Is algorithm selection worth it? Comparing selecting single algorithms and parallel execution, in *AAAI Workshop on Meta-Learning and MetaDL Challenge*, PMLR, **140** (2021), 58–64.
6. Y. Zheng, Digital signcryption or how to achieve cost (signature & encryption) <<cost (signature) + cost (encryption), in *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, Springer, (1997), 165–179. https://doi.org/10.1007/BFb0052234
7. W. Diffie, M. E. Hellman, New directions in cryptography, *IEEE Trans. Inf. Theory*, **22** (1976), 644–654. https://doi.org/10.1145/3549993.3550007
8. A. Shamir, Identity-based cryptosystems and signature schemes, in *Advances in Cryptology: Proceedings of CRYPTO*, Springer Berlin Heidelberg, (1984), 47–53. https://doi.org/10.1007/3-540-39568-7_5
9. S. S. Al-Riyami, K. G. Paterson, Certificateless public key cryptography, in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer Berlin Heidelberg, (2003), 452–473. https://doi.org/10.1007/978-3-540-40061-5_29
10. D. He, Security analysis of a certificateless signcryption scheme (in Chinese), *J. Software*, **24** (2013), 618–622. https://doi.org/10.3724/SP.J.1001.2013.04245
11. Z. Zhao, Security analysis and improvement of certificateless signcryption scheme (in Chinese), *J. Commun.*, **36** (2015), 129–134.
12. Y. Zhou, B. Yang, W. Zhang, Provably secure and efficient certificateless generalized signcryption scheme (in Chinese), *Chin. J. Comput.*, **39** (2016), 543–551. https://doi.org/10.11897/SP.J.1016.2016.00543
13. X. Liu, Z. Wang, Y. Ye, F. Li, An efficient and practical certificateless signcryption scheme for wireless body area networks, *Comput. Commun.*, **162** (2020), 169–178. https://doi.org/10.1016/j.comcom.2020.08.014

14. W. Zhang, W. Huang, J. Feng, Secure communication mechanism for VSN based on certificateless signcryption, *J. Commun. /Tongxin Xuebao*, **42** (2021), 128–136.

15. P. Thorncharoensri, W. Susilo, Y. W. Chow, Privacy-preserving file sharing on cloud storage with certificateless signcryption, *Theor. Comput. Sci.*, **916** (2022), 1–21. https://doi.org/10.1016/j.tcs.2022.02.033

16. L. Wang, J. Gao, Q. Li, Z. Chen, Blockchain-based multi-recipient multi-message signcryption scheme (in Chinese), *J. Software*, **32** (2021), 3606–3627. https://doi.org/10.13328/j.cnki.jos.006034

17. Z. Zhang, Y. Liu, X. Yin, K. Huang, Analysis and improvement of certificateless signature schemes (in Chinese), *J. Cryptologic Res.*, **7** (2020), 389–403. https://doi.org/10.13868/j.cnki.jcr.000375

18. W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, C. Su, Blockchain-based reliable and efficient certificateless signature for IIoT devices, *IEEE Trans. Ind. Inf.*, **18** (2021), 7059–7067. https://doi.org/10.1109/TII.2021.3084753

19. S. S. D. Selvi, S. S. Vivek, D. Shukla, P. Rangan Chandrasekaran, Efficient and provably secure certificateless multi-receiver signcryption, in *Provable Security: Second International Conference, ProvSec 2008*, Springer Berlin Heidelberg, (2008), 52–67. https://doi.org/10.1007/978-3-540-88733-1_4

20. S. Miao, F. Zhang, L. Zhang, Cryptanalysis of a certificateless multi-receiver signcryption scheme, in *2010 International Conference on Multimedia Information Networking and Security*, IEEE, (2010), 593–597. https://doi.org/10.1109/MINES.2010.130

21. J. Shen, Z. Gui, X. Chen, J. Zhang, Y. Xiang, Lightweight and certificateless multi-receiver secure data transmission protocol for wireless body area networks, *IEEE Trans. Dependable Secure Comput.*, **19** (2020), 1464–1475. https://doi.org/10.1109/TDSC.2020.3025288

22. J. Baek, R. Safavi-Naini, W. Susilo, Efficient multi-receiver identity-based encryption and its application to broadcast encryption, in *Public Key Cryptography-PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography*, Springer Berlin Heidelberg, (2005), 380–397. https://doi.org/10.1007/978-3-540-30580-4_26

23. L. Pang, J. Cui, H. Li, Q. Pei, Z. Jiang, Y. Wang, A new multi-receiver ID-based anonymous signcryption (in Chinese), *Chin. J. Comput.*, **34** (2011), 2104–2113.

24. H. Li, X. Chen, L. Ju, L. Pang, Y. Wang, Improved multi-receiver signcryption scheme, *J. Comput. Res. Dev.*, **50** (2013), 1418–1425.

25. L. Pang, H. Li, nMIBAS: A novel multi-receiver ID-based anonymous signcryption with decryption fairness, *Comput. Inf.*, **32** (2013), 441–460.

26. C. Zhou, Provably secure and efficient multi-receiver identity-based generalized signcryption scheme, in *2014 Ninth Asia Joint Conference on Information Security*, IEEE, (2014), 82–88. https://doi.org/10.1109/AsiaJCIS.2014.10

27. Y. Zhou, B. Yang, Q. Wang, Anonymous hybrid signcryption scheme with multi-receiver (multi-message) based on identity (in Chinese), *J. Software*, **29** (2018), 442–455. https://doi.org/10.13328/j.cnki.jos.005250

28. Q. Jing, B. Jun, S. Chuan, H. Sheng, Secure and efficient multi-message and multi-receiver ID-based signcryption for rekeying in ad hoc networks, *J. Chongqing Univ.*, **12** (2013), 91–96.

29. Y. Zhao, Y. Wang, Y. Liang, H. Yu, Y. Ren, Identity-based broadcast signcryption scheme for vehicular platoon communication, *IEEE Trans. Ind. Inf.*, **19** (2022), 7814–7824. https://doi.org/10.1109/TII.2022.3203724

30. Y. Zhou, B. Yang, W. Zhang, Anonymous certificateless signcryption scheme with multi-receiver (in Chinese), *Acta Electron. Sin.*, **44** (2016), 1784–1790. https://doi.org/10.3969/j.issn.0372-2112.2016.08.002

31. L. Pang, M. Kou, M. Wei, H. Li, Efficient anonymous certificateless multi-receiver signcryption scheme without bilinear pairings, *IEEE Access*, **6** (2018), 78123–78135. https://doi.org/10.1109/ACCESS.2018.2884798

32. L. Pang, M. Wei, H. Li, Efficient and anonymous certificateless multi-message and multi-receiver signcryption scheme based on ECC, *IEEE Access*, **7** (2019), 24511–24526. https://doi.org/10.1109/ACCESS.2019.2900072

33. C. Peng, J. Chen, M. S. Obaidat, P. Vijayakumar, D. He, Efficient and provably secure multireceiver signcryption scheme for multicast communication in edge computing, *IEEE Internet Things J.*, **7** (2019), 6056–6068. https://doi.org/10.1109/JIOT.2019.2949708

34. M. Fu, X. Gu, W. Dai, J. Lin, H. Wang, Secure multi-receiver communications: Models, proofs, and implementation, in *Algorithms and Architectures for Parallel Processing: 19th International Conference*, Springer Cham, (2019), 689–709. https://doi.org/10.1007/978-3-030-38991-8_45

35. B. Wang, J. Rong, S. Zhang, L. Liu, Research on data security of multicast transmission based on certificateless multi-recipient signcryption in AMI, *Int. J. Electr. Power Energy Syst.*, **121** (2020), 106123. https://doi.org/10.1016/j.ijepes.2020.106123

36. Y. Zhou, Y. Bo, W. Zhang, Multi-receiver and multi-message of certificateless signcryption scheme (in Chinese), *Chin. J. Comput.*, **40** (2020), 1714–1724.

37. X. Yu, W. Zhao, D. Tang, Efficient and provably secure multi-receiver signcryption scheme using implicit certificate in edge computing, *J. Syst. Archit.*, **126** (2022), 102457. https://doi.org/10.1016/j.sysarc.2022.102457

38. A. Karati, C. Fan, J. Huang, An efficient pairing-free certificateless signcryption without secure channel communication during secret key issuance, *Procedia Comput. Sci.*, **171** (2020), 110–119. https://doi.org/10.1016/j.procs.2020.04.012

AIMS Press