New Construction of Short Certificate-Based Signature against Existential Forgery Attacks

Yang Lu[1], Gang Wang[1], Jiguo Li[1], Jian Shen[2]
[1]College of Computer and Information, Hohai University, Nanjing, China
[e-mail: luyangnsd@163.com, wg15951977612@163.com, ljg1688@163.com]
[2]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[e-mail: s_shenjian@126.com]
*Corresponding author: Yang Lu

Received January 9, 2017; revised March 24, 2017; accepted April 17, 2017; published July 31, 2017

Abstract

Certificate-based cryptography is a useful public key cryptographic primitive that combines the merits of traditional public key cryptography and identity-based cryptography. It not only solves the key escrow problem inherent in identity-based cryptography, but also simplifies the cumbersome certificate management problem in traditional public key cryptography. So far, four short certificate-based signature schemes have been proposed. However, three of them fail in achieving the existential unforgeability under adaptive chosen-message attacks and the remaining one was not constructed in the normal framework of certificate-based signature. In this paper, we put forward a new short certificate-based signature scheme. The proposed scheme is devised in the normal framework of certificate-based signature and overcomes the security weaknesses in the previous short certificate-based signature schemes. In the random oracle model, we formally prove that it achieves the existential unforgeability against adaptive chosen-message attacks. Performance comparison shows that it is efficient and practical.

Keywords: Certificate-based cryptography; short signature; bilinear pairing; existential unforgeability; random oracle model

We would like to thank the anonymous referees for their helpful comments. This work is supported by the National Natural Science Foundation of China (grant No. 61672207), the Fundamental Research Funds for the Central Universities (grant No. 2016B10114), the Natural Science Foundation of Jiangsu Province (grant No. BK20161511), a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology.

https://doi.org/10.3837/tiis.2017.07.018 **ISSN : 1976-7277**

1. Introduction

In public key cryptography (PKC), each user has a public key and a private key. The public key is published and publicly accessible while the corresponding private key is kept secret by its owner. In traditional PKC, each public key is generated with no connection to the identity of its owner. Therefore, a public key infrastructure (PKI) is employed for vouching the relationship between a user's identity and a public key by certificates. However, the traditional PKI technology is faced with many challenges in practice, especially the complicated certificate management problem. To simplify the management of the public key certificates, Shamir [1] introduced the concept of identity-based cryptography (IBC) in 1984. In IBC, a user's public key could be an arbitrary string related to his identity and his private key is computed from his identity by a trusted authority called private key generator (PKG). The biggest merit of IBC is that it eliminates the need for public key certificates. However, IBC inevitably suffers from the key escrow problem as all users' private keys are known to the PKG.

In order to fill the gap between traditional PKC and IBC, Al-Riyami and Paterson [2] proposed the notion of certificateless public key cryptography (CL-PKC) in Asiacrypt 2003. In CL-PKC, a trusted third party called key generation center (KGC) is employed for generating a partial private key for each user. After generating a secret key and a public key independently, each user combines his secret key with the partial private key from the KGC to generate his full private key. Since KGC does not know any user's private key, CL-PKC overcomes the key escrow problem inherent in IBC. However, as partial private keys should be sent from KGC to users over secure channels, CL-PKC unavoidably suffers from the key distribution problem.

In Eurocrypt 2003, Gentry [3] introduced another new paradigm called certificate-based cryptography (CBC), which represents an interesting balance between IBC and traditional PKC. As in traditional PKC, each user in CBC first generates a pair of public key and private key, and then requests a certificate from a certificate authority (CA). A certificate in CBC acts not only as a public key certificate (as in traditional PKC) but also as a partial decryption/signing key, namely that each user should combine his private key with his certificate to generate his decryption/signing key. This additional functionality provides an effective implicit certificate property so that a user needs both his private key and certificate to perform cryptographic operations (such as decryption and signing), while the other communication parties need not obtain the fresh information on this user's certificate status. As a result, CBC eliminates the third-party query for the certificate status and simplifies the cumbersome certificate revocation problem in traditional PKI. As introduced by Gentry [3], CBC can be used to construct an efficient PKI requiring fewer infrastructures than the traditional one. Furthermore, since the CA does not know users' private keys and the certificates can be sent to the users publicly, CBC overcomes both the key escrow problem and the secure channel problem.

Since its advent, CBC has aroused great interest in the academia and numerous schemes have been proposed, including many certificate-based encryption (CBE) schemes and certificate-based signature (CBS) schemes. In [3], Gentry brought forth the first CBE scheme from the well-known identity-based encryption (IBE) scheme proposed by Boneh and Franklin [4]. A subsequent work by Yum and Lee [5] indicated that IBE implies both CBE and certificateless public-key encryption (CL-PKE) [2] and provided a generic conversion from

IBE to those two primitives respectively. However, Galindo *et al*. [6] pointed out that Yum-Lee's constructions are insecure against chosen-ciphertext attacks due to a naive use of double encryption without further security treatments. In [7], Al-Riyami and Paterson refined the definition of CBE and proposed a generic construction of CBE from CL-PKE. However, Kang and Park [8] demonstrated that Al-Riyami and Paterson's proposal suffers from a fault in the security proof. Wu *et al*. [9] made a further observation on the relations between CL-PKE and CBE and proposed a new generic construction of CBE from CL-PKE. In [10], Galindo *et al*. proposed the first CBE scheme without random oracles. Their scheme was constructed by combining Waters' IBE scheme [11] with Boneh and Boyen's IBE scheme [12]. So far, many researchers have attempted to develop CBE schemes with better efficiency or stronger security, *e.g*. [13-19]. Furthermore, some variants of CBE (*e.g*., multi-receiver CBE [20], certificate-based proxy encryption [21] and certificate-based proxy re-encryption [22]) have also been proposed. Following the idea of CBE in [3], Kang *et al*. [23] proposed the notion of CBS and two concrete CBS schemes in the random oracle model. However, one of Kang *et al*.'s CBS schemes is vulnerable to the public key replacement attack [24]. In [24], Li *et al*. improved Kang *et al*.'s CBS security model by introducing the public key replacement (PKR) attack and proposed a novel CBS scheme secure against PKR attacks in the random oracle model. And later in [25], Li *et al*. further proposed a CBS scheme secure against PKR attacks in the standard model. In [26], Liu *et al*. proposed two new CBS schemes. Their first scheme does not depend on the bilinear pairings while the second one can be proven secure without resorting to the random oracles. However, Zhang *et al*. [27] showed that the pairing-free CBS scheme proposed by Liu *et al*. [26] is insecure against both two adversarial games of CBS. To fix the flaws in Liu *et al*.'s pairing-free CBS scheme, Zhang *et al*. [27] proposed an improved CBS scheme with bilinear pairings and proved its security in the random oracle model. In [28], Wu *et al*. considered the relations between CBS and certificateless signature (CLS) [2] and proposed a generic construction of CBS from CLS [2]. They also derived two concrete CBS schemes, where the first one is a short CBS scheme. Later, Liu *et al*. [29], Li *et al*. [30] and Hung *et al*. [31] respectively proposed a short CBS scheme. To improve the efficiency, Li *et al*. [32] proposed the first pairing-free CBS scheme that can achieve the existential unforgeability against adaptive chosen-message attacks. Recently, Lu and Li [33] indicated that the previous CBS schemes without random oracles [25, 26] are insecure under the malicious CA attack. To fix this problem, they proposed a new CBS scheme that can be proven secure against malicious CA attacks in the standard model. Besides, many variants of CBS (*e.g*., certificate-based ring signature [34], certificate-based sequential aggregate signature [35] and certificate-based key-insulated signature [36]) have also been proposed.

1.1 Motivation and contribution

Due to the short signature length, short signatures have many potential applications in real life. A short signature scheme is particularly desirable in low-computation power and/or low-bandwidth communication environments such as PDAs, cell phones and sensors. As mentioned in [37], communicating even one bit of data costs significantly more power than executing one 32-bit instruction. Thus, reducing the number of bits in communication is important to increase the battery life. Also, in many applications, communication channels are not reliable and thus the number of bits in communication should be kept as few as possible. In recent years, short signature has aroused great interest in the research community and many schemes have been proposed over different PKC, including traditional short signature [38-40], short identity-based signature [41-42] and short certificateless signature [43-45].

As far as we know, there exist four short CBS (SCBS) schemes in the literature so far [28-31]. However, three of them [28-30] fail in achieving the existential unforgeability against adaptive chosen-message attacks. In [46], Cheng *et al*. first pointed out that the SCBS scheme proposed by Liu *et al*. [29] is insecure against existential forgery attacks. The insecurity of Liu *et al*.'s scheme is that an adversary who knows the target user's private key can arbitrarily forge signatures on behalf of the target user after seeing a message/signature pair issued by the target user. Later in [31], Hung *et al*. showed that the SCBS scheme proposed by Li *et al*. [30] is also insecure against existential forgery attacks. Our cryptanalysis in this paper demonstrates that the SCBS scheme proposed by Wu *et al*. [28] suffers from the similar security weakness. Thus, the SCBS scheme proposed by Hung *et al*. in [31] is the only one that satisfies the existential unforgeability against adaptive chosen-message attacks. However, Hung *et al*.'s scheme was not constructed in the normal framework of CBS introduced by Kang *et al*. [23]. In a normal CBS scheme, a user's public/private key pair is produced independently by the user himself/herself. But, in Hung *et al*.'s SCBS scheme, any user's public key should be generated under the help of a CA, which unavoidably increases the burden on the CA. Therefore, devising a secure SCBS scheme in the normal framework of CBS remains an unsolved problem until now.

In this paper, we first show that the SCBS scheme proposed by Wu *et al*. [28] cannot resist the existential forgery attack. To overcome the weaknesses in the previous SCBS schemes, we design a new SCBS scheme. This new scheme is constructed in the normal framework of CBS and provides resistance to the existential forgery attack. In the random oracle model [47, 48], we formally prove that it achieves the existential unforgeability against adaptive chosen-message attacks. Comparison analysis shows that it is more efficient than the SCBS scheme proposed by Hung *et al*. [31] and the previous pairing-based CBS schemes in both the computation efficiency and the communication bandwidth.

1.2 Paper Organization

The rest of our paper is organized as follows. In Section 2, we briefly review some related preliminaries. In Section 3, we present the existential forgery attack against Wu *et al*.'s SCBS scheme. Our proposed SCBS scheme is described and analyzed in Section 4. Finally, we draw our conclusions in Section 5.

2. Preliminaries

2.1 Bilinear pairing and computational hardness problems

Let k be a security parameter and p be a k-bit prime number. Let G and G_T be two cyclic groups of same prime order p and P be a generator of G. A bilinear pairing is a map e: $G \times G \rightarrow G_T$ satisfying the following three properties:

(1) Bilinearity: $e(aP, bP) = e(P, P)^{ab}$ for any $a, b \in Z_p^*$;

(2) Non-degeneracy: $e(P, P) \neq 1$;

(3) Computability: For any $a, b \in Z_p^*$, there exists an efficient algorithm to compute $e(aP, bP)$.

The security of our SCBS scheme is based on the hardness of the q-collusion attack algorithm (q-CAA) problem [49] and the weak modified q-CAA problem [29].

Definition 1. The q-collusion attack algorithm (q-CAA) problem in G is, given a tuple $(P, xP, (x+\omega_1)^{-1}P, \ldots, (x+\omega_q)^{-1}P, \omega_1, \ldots, \omega_q) \in G^{q+2} \times (Z_p^*)^q$ for unknown $x \in Z_p^*$, to compute $(x+\omega^*)^{-1}P \in G$ for some value $\omega^* \in Z_p^* - \{\omega_1, \ldots, \omega_q\}$.

Definition 2. The weak modified q-CAA problem in G is, given a tuple $(P, xP, aP, bP, (x+\omega_1)^{-1}abP, \ldots, (x+\omega_q)^{-1}abP, \omega_1, \ldots, \omega_q) \in G^{q+2} \times (Z_p^*)^q$ for unknown $x, a, b \in Z_p^*$, to compute $(x+\omega^*)^{-1}abP \in G$ for some value $\omega^* \in Z_p^* - \{\omega_1, \ldots, \omega_q\}$ or $abP \in G$.

2.2 Formal model of certificate-based signature

Normally, a certificate-based signature (CBS) scheme consists of five algorithms [23, 24]:

(1) ***Setup***(k): On input a security parameter k, this algorithm generates a master secret key msk and a list of public parameters $params$. After the algorithm is performed, the CA publishes $params$ and keeps msk secret.

(2) ***UserKeyGen***($params$): On input the public parameters $params$, this algorithm generates a public key PK_{ID} and a private key SK_{ID} for a user with identity ID.

(3) ***CertGen***($params$, msk, ID, PK_{ID}): On input the public parameters $params$, the master secret key msk and a user's identity ID and public key PK_{ID}, this algorithm generates a certificate $Cert_{ID}$ for the user with identity ID. After the algorithm is executed, the CA sends the certificate to the user via a public channel.

(4) ***Sign***($params$, M, ID, SK_{ID}, $Cert_{ID}$): On input the public parameters $params$, a message M, a signer's identity ID, private key SK_{ID} and certificate $Cert_{ID}$, this algorithm generates a signature σ of the message M.

(5) ***Verify***($params$, M, σ, ID, PK_{ID}): On input the public parameters $params$, a message M, a candidate signature σ and the signer's identity ID and public key PK_{ID}, this algorithm outputs 1 if σ is a valid signature of the message M under the identity ID and public key PK_{ID} or 0 otherwise.

For correctness, it is requires that if σ = ***Sign***($params$, M, ID, SK_{ID}, $Cert_{ID}$), then ***Verify***($params$, M, σ, ID, PK_{ID}) = 1, where the public parameters $params$, the singer's private/public key pair (SK_{ID}, PK_{ID}) and certificate $Cert_{ID}$ are respectively generated according to the specification of the algorithms ***Setup***, ***UserKeyGen*** and ***CertGen***.

Remark. In the framework of Hung *et al.*'s SCBS scheme [31], each user first uses the algorithm ***UserKeyGen*** to generate a partial public key and a private key and then sends his/her identity and partial public key to the CA. Once receiving a user's identity and partial public key, the CA uses the algorithm ***CertGen*** to generate a certificate and a full public key for the user. It is clear that Hung *et al.*'s framework is slightly weaker than the normal one, as it increases both the computation and communication burdens on the CA.

A secure CBS scheme requires that a user can generate a valid signature successfully only when both his private key and certificate are known. In other words, knowing one of a user's private key and certificate is unable to impersonate this user. Therefore, two different types of adversaries should be considered in the security model of CBS [24], namely the Type-I adversary and the Type-II adversary. The Type-I adversary (denoted by A_I) models an uncertified user who wants to impersonate a forged victim by using a public/private key pair of its choice. Because the CA can generate the certificate of a user only after authenticating, the Type-I adversary A_I cannot obtain the certificate of the victim. However, the Type-I adversary A_I may already posses the certificates of other users. Hence, the security model should allow the Type-I adversary A_I to obtain the certificates of other users except for the certificate of the victim. The Type-II adversary (denoted by A_{II}) models an honest-but-curious CA who knows

the master secret key. However, the Type-II adversary A_{II} is disallowed to access the victim's private key and replace public keys. Otherwise, there would be not any security at all since the Type-II adversary A_{II} would know all private keys and certificates.

In this paper, we will use the enhanced CBS security model by Li *et al*. [24] to prove the security of the proposed SCBS scheme. This security model refines the one proposed by Kang *et al*. [23] by introducing the PKR attack. We note that most of the subsequently proposed CBS schemes (*i.e*., [24-27, 33]) were essentially proven secure under the security model that captures the PKR attack. To formalize the security model, we will use the following five oracles. A Type-I or Type-II adversary can query some of these oracles adaptively.

(1) $O^{CreateUser}$: On receiving an identity *ID*, the challenger responds with a public key PK_{ID}. If the identity *ID* has not been created, the challenger generates a pair of private key SK_{ID} and public key PK_{ID} and then returns PK_{ID}. In this case, the identity *ID* is said to be created. For simplicity, we assume that other oracles defined below only respond to an identity that has been created.

(2) $O^{PrivateKey}$: On receiving an identity *ID*, the challenger responds with a private key SK_{ID}.

(3) $O^{Certificate}$: On receiving an identity *ID*, the challenger responds with a certificate $Cert_{ID}$ for the current public key of the identity *ID*. This oracle is only queried by the Type-I adversary since the Type-II adversary can generate any user's certificate by itself.

(4) $O^{ReplacePublicKey}$: On receiving an identity *ID* and a false public key PK'_{ID}, the challenger replaces the current public key associated with the identity *ID* with PK'_{ID}. Because the challenger may be not aware of the private key of an identity *ID* if the associated public key has been replaced, the adversary may be asked to supply the private key corresponding to the false public key PK'_{ID} so that the challenger can correctly answer the signing queries with respect to the identity *ID*. This oracle is only queried by the Type-I adversary since the Type-II adversary is disallowed to replace public keys.

(5) O^{Sign}: On receiving an identity *ID* and a message *M*, the challenger responds with a signature σ. For the Type-I adversary, this oracle works as same as the oracle $O^{CB\text{-}StrongSign}$ defined in [28]. If the public key of the identity *ID* is the original public key generated by the oracle $O^{CreateUser}$, it responds in the normal way. Else if the public key of the identity *ID* has been replaced, it generates a signature σ using the private key corresponding to the false public key provided by the Type-I adversary and the certificate for the false public key.

A CBS scheme should satisfy the existential unforgeability against adaptive chosen-message attacks (hereafter referred to as "EUF-CMA security"). The EUF-CMA security is the strongest security notion for a signature scheme. It guarantees that the adversary cannot forge a valid message/signature pair that is not already known to him even if the adversary first learns signatures on arbitrary messages of the adversary's choice. This security notion can be defined by the following adversarial game, in which a challenger interacts with a Type-I adversary or a Type-II adversary.

(1) **Setup Phase.** On input a security parameter *k*, the challenger simulates the algorithm ***Setup***(*k*) to generate a master secret key *msk* and a list of public parameters *params*. Then, it sends *params* to the adversary *A*. If *A* is a Type-II adversary, the challenger also sends the master secret key *msk* to it.

(2) **Query Phase.** In this phase, the adversary *A* can adaptively make request to the oracles $O^{CreateUser}$, $O^{PrivateKey}$, $O^{Certificate}$, $O^{ReplacePublicKey}$ and O^{Sign} if it is a Type-I adversary or the oracles $O^{CreateUser}$, $O^{PrivateKey}$ and O^{Sign} if it is a Type-II adversary. The challenger responds as described above.

(3) **Forge Phase.** Finally, the adversary A outputs a forgery (ID^*, M^*, σ^*). We say that the adversary A wins the game if the following conditions are satisfied:

- **Verify**$(params, m^*, \sigma^*, ID^*, PK_{ID^*}) = 1$;
- The adversary A has never queried the oracle O^{Sign} on (ID^*, M^*);
- The adversary A has never queried the oracle $O^{Certificate}$ on ID^* if it is a Type-I adversary;
- The adversary A has never queried the oracle $O^{PrivateKey}$ on ID^* if it is a Type-II adversary.

The advantage of the adversary A succeeding in the above game is defined to be the probability that it wins.

Definition 3. A CBS scheme is said to achieve the EUF-CMA security if no PPT adversary has non-negligible advantage in the above game.

3. Cryptanalysis of Wu et al.'s SCBS Scheme

In this section, we show that the SCBS scheme proposed by Wu *et al.* [28] does not achieve the EUF-CMA security.

3.1 Description of Wu *et al.*'s SCBS scheme

Wu *et al.*'s SCBS scheme [28] is described as follows:

(1) ***Setup***: On input a security parameter k, this algorithm performs as follows: Generate two cyclic groups G and G_T of k-bit prime order p and a bilinear pairing e: $G \times G \rightarrow G_T$; randomly select an integer $s \in Z_p^*$ and a generator $P \in G$ and then compute $mpk = sP$; choose two hash functions H_0, H_1: $\{0,1\}^* \rightarrow G$; set the master secret key $msk = s$ and the public parameters $params = \{G, G_T, e, p, P, mpk, H_0, H_1\}$.

(2) ***UserKeyGen***: On input the public parameters $params$, this algorithm chooses a random value $x_{ID} \in Z_p^*$ as a private key SK_{ID} for the user with identity ID and then computes the corresponding public key $PK_{ID} = x_{ID}P$.

(3) ***CertGen***: On input the public parameters $params$, the master secret key $msk = s$ and a user's identity ID and public key PK_{ID}, this algorithm computes $Cert_{ID} = sQ_{ID}$, where $Q_{ID} = H_0(ID, PK_{ID})$.

(4) ***Sign***: On input the public parameters $params$, a message M and a signer's identity ID, private key SK_{ID} and certificate $Cert_{ID}$, this algorithm computes a signature $\sigma = Cert_{ID} + SK_{ID}H_1(M, ID, PK_{ID})$.

(5) ***Verify***: On input the public parameters $params$, a message M, a candidate signature σ and the signer's identity ID and public key PK_{ID}, this algorithm checks whether $e(\sigma, P) = e(H_0(ID, PK_{ID}), mpk) \cdot e(H_1(M, ID, PK_{ID}), PK_{ID})$ holds. It outputs 1 if the equation holds or 0 otherwise.

3.2 Existential forgery attack against Wu *et al.*'s SCBS scheme

Recall that the Type-I adversary A_I does not have access to the target user's certificate since it models an uncertified user. The insecurity of Wu *et al.*'s scheme lies in the fact that, given a signature on an arbitrary message issued by the target user, the adversary A_I can easily compute the target user's certificate directly. Thus, it can arbitrarily forge signatures on behalf of the target user.

Concretely, the adversary A_I does in the following way. Assume that ID^* is the target identity chosen by the adversary A_I. After querying the public key PK_{ID^*} and the private key SK_{ID^*} of the target identity ID^*, the adversary A_I queries the oracle O^{Sign} on the target identity ID^* and an arbitrary message M^* to obtain a signature σ^*. According to the specification of the algorithm ***Sign*** in Wu *et al.*'s SCBS scheme, the signature σ^* has the form

$$\sigma^* = Cert_{ID^*} + SK_{ID^*} \cdot H_1(M^*, ID^*, PK_{ID^*}), \quad (1)$$

where H_1: $\{0,1\}^* \rightarrow G$ is a hash function. Therefore, the adversary A_I can compute the certificate of the target identity ID^* as

$$Cert_{ID^*} = \sigma^* - SK_{ID^*} \cdot H_1(M^*, ID^*, PK_{ID^*}). \quad (2)$$

Since the adversary A_I obtains both the private key SK_{ID^*} and the certificate $Cert_{ID^*}$ of the target identity ID^*, the existential unforgeability of Wu *et al.*'s SCBS scheme is broken.

4. The Proposed SCBS Scheme

In this section, we design a new SCBS scheme in the normal framework of CBS.

4.1 Description of the proposed scheme

Our SCBS scheme consists of the following algorithms:

(1) ***Setup***: On input a security parameter k, the CA performs the following steps to generate the master secret key msk and the public parameters $params$: Generate two cyclic groups G and G_T of k-bit prime order p and a bilinear pairing e: $G \times G \rightarrow G_T$; randomly select two integers $s_1, s_2 \in Z_p^*$ and a generator $P \in G$ and then compute $mpk_1 = s_1P$ and $mpk_2 = s_2P$; choose three hash functions H_0: $\{0,1\}^* \times G \rightarrow G$, H_1: $\{0,1\}^* \times \{0,1\}^* \times G \times G \rightarrow Z_p^*$ and H_2: $\{0,1\}^* \times \{0,1\}^* \times G \times G \times G \rightarrow Z_p^*$; set the master secret key $msk = (s_1, s_2)$ and the public parameters $params = \{G, G_T, e, p, P, mpk_1, mpk_2, H_0, H_1, H_2\}$.

(2) ***UserKeyGen***: On input the public parameters $params$, this algorithm chooses a random value $x_{ID} \in Z_p^*$ as a private key SK_{ID} for the user with identity ID and then computes the corresponding public key $PK_{ID} = x_{ID}P$.

(3) ***CertGen***: On input the public parameters $params$, the master secret key $msk = (s_1, s_2)$ and a user's identity ID and public key PK_{ID}, this algorithm computes the certificate $Cert_{ID} = (Cert_{ID,1}, Cert_{ID,2}) = (s_1Q_{ID}, s_2Q_{ID})$, where $Q_{ID} = H_0(ID, PK_{ID})$.

(4) ***Sign***: On input the public parameters $params$, a message M and a signer's private key SK_{ID} and certificate $Cert_{ID} = (Cert_{ID,1}, Cert_{ID,2})$, this algorithm computes the signature $\sigma = (\alpha + SK_{ID})^{-1}(Cert_{ID,1} + \beta Cert_{ID,2})$, where $\alpha = H_1(M, ID, PK_{ID}, mpk_1)$ and $\beta = H_2(M, ID, PK_{ID}, mpk_1, mpk_2)$.

(5) ***Verify***: On input the public parameters $params$, a message M, a signature σ and the signer's identity ID and public key PK_{ID}, this algorithm checks whether $e(\sigma, \alpha P + PK_{ID}) = e(Q_{ID}, mpk_1 + \beta mpk_2)$ holds, where $Q_{ID} = H_0(ID, PK_{ID})$, $\alpha = H_1(M, ID, PK_{ID}, mpk_1)$ and $\beta = H_2(M, ID, PK_{ID}, mpk_1, mpk_2)$. It outputs 1 if the equation holds or 0 otherwise.

The correctness of the proposed scheme can be verified as follows:

$$\begin{aligned} e(\sigma, \alpha P + PK_{ID}) &= e((\alpha + SK_{ID})^{-1}(Cert_{ID,1} + \beta Cert_{ID,2}), (\alpha + SK_{ID})P) \\ &= e(s_1 Q_{ID} + \beta s_2 Q_{ID}, P) \\ &= e(Q_{ID}, mpk_1 + \beta mpk_2). \end{aligned} \quad (3)$$

4.2 Security proof

Theorem 1. If a Type-I adversary A_I has non-negligible advantage ε against our SCBS scheme, then there exists an algorithm B to solve the weak modified q-CAA problem over the group G with non-negligible probability

$$\varepsilon' \geq \frac{\varepsilon}{q_C(1+q_S)e}(1-\frac{1}{q_C})^{q_K}, \quad (4)$$

where e denotes the base of natural logarithm, q_C, q_K and q_S respectively denote the maximal number of A_I's queries to $O^{CreateUser}$, $O^{PrivateKey}$ and O^{Sign} and $q \geq q_S$.

Proof. Assume that the algorithm B is given a random instance (P, xP, aP, bP, $(x + \omega_1)^{-1}abP,\ldots, (x + \omega_q)^{-1}abP$, $\omega_1,\ldots, \omega_q$) of the weak modified q-CAA problem. It goal is to compute $(x + \omega^*)^{-1}abP$ for some value $\omega^* \in Z_p^* - \{\omega_1,\ldots, \omega_q\}$ or abP. In order to use the adversary A_I to solve the given problem, the algorithm B needs to simulate a challenger and all oracles for the adversary A_I.

Setup Phase. The algorithm B randomly chooses $s_1, s_2 \in Z_p^*$, sets $mpk_1 = s_1(aP)$ and $mpk_2 = s_2(aP)$. Additionally, it randomly chooses an index $I \in \{1, 2,\ldots, q_C\}$. It then outputs $params = \{G, G_T, e, p, P, mpk_1, mpk_2, H_0, H_1, H_2\}$ to the adversary A_I, where H_0, H_1 and H_2 are three random oracles controlled by the algorithm B.

Query Phase. During the query phase, the adversary A_I can adaptively make queries to the oracles H_0, H_1, H_2, $O^{CreateUser}$, $O^{PrivateKey}$, $O^{ReplacePublicKey}$, $O^{Certificate}$ and O^{Sign}. The algorithm B responds as follows:

(1) H_0 *queries*: The algorithm B maintains a list H_0List of tuples (ID_i, PK_i, Q_i, u_i). On receiving such a query on (ID_i, PK_i), B returns Q_i directly if the list H_0List contains a tuple (ID_i, PK_i, Q_i, u_i). Otherwise, it does the following:

- If $ID_i = ID_I$ (i.e., ID_i is the I-th distinct identity submitted to the oracle $O^{CreateUser}$), it sets $Q_i = bP$, inserts a new tuple (ID_i, PK_i, Q_i, $\perp$) into the list H_0List and returns Q_i;
- Otherwise, it randomly chooses $u_i \in Z_p^*$, computes $Q_i = u_iP$, inserts a new tuple (ID_i, PK_i, Q_i, u_i) into the list H_0List and returns Q_i.

(2) H_1 *queries*: The algorithm B maintains a list H_1List of tuples (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$). On receiving such a query on (M_i, ID_i, PK_i, mpk_1), the algorithm B returns α_i directly if H_1List contains a tuple (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$). Otherwise, it does the following:

- If $ID_i = ID_I$ and PK_i is the original public key generated by the oracle $O^{CreateUser}$ (namely that PK_i has not been replaced), then it picks a random coin $coin_i \in \{0, 1\}$ with probability $\Pr[coin_i = 1] = \gamma$ for some value γ that will be determined later. If $coin_i = 0$, it randomly chooses a value $\omega_i \in \{\omega_1, \omega_2,\ldots, \omega_q\}$ that has not been selected before and sets $\alpha_i = \omega_i$; else if $coin_i = 1$, it randomly chooses $\alpha_i \in Z_p^*$ such that $\alpha_i \notin \{\omega_1, \omega_2,\ldots, \omega_q\}$. In either cases, the algorithm B inserts a new tuple (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$) into the list H_1List and returns α_i.

– Otherwise, it randomly chooses $\alpha_i \in Z_p^*$, inserts a new tuple (M_i, ID_i, PK_i, mpk_1, α_i, $\perp$) into the list H_1List and returns α_i.

(3) H_2 *queries*: The algorithm B maintains a list H_2List of tuples (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i). On receiving such a query on (M_i, ID_i, PK_i, mpk_1, mpk_2), the algorithm B returns β_i directly if the list H_2List contains a tuple (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i). Otherwise, it randomly chooses $\beta_i \in Z_p^*$, inserts a new tuple (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i) into H_2List and returns β_i.

(4) $O^{CreateUser}$ *queries*: The algorithm B maintains a list $UserList$ of tuples (ID_i, SK_i, PK_i). On receiving such a query on an identity ID_i, the algorithm B returns PK_i directly if the list $UserList$ contains a tuple (ID_i, SK_i, PK_i). Otherwise, it does the following:

– If $i = I$ (*i.e.*, ID_i is the *I-th* distinct identity submitted to this oracle), it sets $PK_i = xP$, inserts a new tuple (ID_i, $\perp$, PK_i) into the list $UserList$ and returns PK_i;

– Otherwise, it randomly chooses $x_i \in Z_p^*$, sets $SK_i = x_i$ and $PK_i = x_iP$, inserts a new tuple (ID_i, SK_i, PK_i) into the list $UserList$ and returns PK_i.

(5) $O^{PrivateKey}$ *queries*: On receiving such a query on an identity ID_i, the algorithm B aborts if $ID_i = ID_I$. Otherwise, it retrieves a tuple of the form (ID_i, SK_i, PK_i) from the list $UserList$ and returns SK_i.

(6) $O^{Certificate}$ *queries*: On receiving such a query on an identity ID_i, the algorithm B aborts if $ID_i = ID_I$. Otherwise, it simulates a H_0 query on (ID_i, PK_i) to get a tuple (ID_i, PK_i, Q_i, u_i) and returns $Cert_i = (u_is_1aP, u_is_2aP)$.

(7) $O^{ReplacePublicKey}$ *queries*: On receiving such a query on (ID_i, PK_i', SK_i'), the algorithm B replaces the current key pair (SK_i, PK_i) associated with the identity ID_i with (PK_i', SK_i'). Note that such a query is rejected if (PK_i', SK_i') is not a valid key pair.

(8) O^{Sign} *queries*: On receiving such a query on (M_i, ID_i), the algorithm B retrieves a tuple (ID_i, SK_i, PK_i) from $UserList$, a tuple (ID_i, PK_i, Q_i, u_i) from H_0List, a tuple (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$) from H_1List and a tuple (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i) from H_2List respectively and does the following:

– If $ID_i = ID_I$ and $coin_i = 1$, it aborts.

– Else if $ID_i = ID_I$ and $coin_i = 0$, it computes $\sigma_i = (s_1 + \beta_is_2)(x + \omega_i)^{-1}abP$ and returns σ_i as a signature to the adversary A_I. Obviously, σ_i is a valid signature of the message M_i because it can pass the verification $e(\sigma_i, \alpha_iP + PK_i) = e(Q_i, mpk_1 + \beta_impk_2)$, where $\alpha_i = \omega_i$, $PK_i = xP$ and $Q_i = bP$.

– Otherwise, it computes $\sigma_i = (s_1 + \beta_is_2)(SK_i + \alpha_i)^{-1}u_iaP$ and returns σ_i as a signature to the adversary A_I. Obviously, σ_i is a valid signature of the message M_i because it can pass the verification $e(\sigma_i,\alpha_iP + PK_i) = e(Q_i, mpk_1 + \beta_impk_2)$, where $PK_i = SK_iP$ and $Q_i = u_iP$.

Forge Phase. Finally, if the algorithm B does not abort, the adversary A_I outputs a forgery (ID_{ch}, M_{ch}, σ_{ch}). If $ID_{ch} \neq ID_I$, B aborts the game. Otherwise, it retrieves a tuple (ID_{ch}, SK_{ch}, PK_{ch}) from the list $UserList$ and a tuple (M_{ch}, ID_{ch}, PK_{ch}, mpk_1, α_{ch}, $coin_{ch}$) from the list H_1List respectively and does the following:

– If $coin_{ch} = 0$, it aborts;

– Else if $coin_{ch} = 1$ which implies $\alpha_{ch} \notin \{\omega_1, \omega_2, \ldots, \omega_q\}$, it retrieves a tuple ($M_{ch}$, ID_{ch}, PK_{ch}, mpk_1, mpk_2, β_{ch}) from H_2List. If PK_{ch} is the original public key generated by the oracle $O^{CreateUser}$ (namely that PK_{ch} has not been replaced), it computes $T_1 = (s_1 + \beta_{ch}s_2)^{-1}\sigma_{ch}$ as the solution to the given weak modified q-CAA problem. Otherwise, it computes $T_2 = (s_1 + \beta_{ch}s_2)^{-1}(SK_{ch} + \alpha_{ch})\sigma_{ch}$ as the solution to the given weak modified q-CAA problem.

Notice that, if σ_{ch} is a valid signature for the message M_{ch} under the target identity ID_{ch} and the public key PK_{ch}, then $\sigma_{ch} = (s_1 + \beta_{ch}s_2)(x + \alpha_{ch})^{-1}abP$ if PK_{ch} is the original public key generated by the oracle $O^{CreateUser}$ or $\sigma_{ch} = (s_1 + \beta_{ch}s_2)(SK_{ch} + \alpha_{ch})^{-1}abP$ otherwise. Therefore, we have that $T_1 = (x + \alpha_{ch})^{-1}abP$ and $T_2 = abP$, which are both the correct solutions to the given weak modified q-CAA problem.

Advantage analysis. To derive the algorithm B's advantage in solving the given problem, we define the following events:

- E_1: The adversary A_I's forgery satisfies $ID_{ch} = ID_I$ and $coin_{ch} = 1$;
- E_2: The algorithm B dose not abort during the simulation of the oracle $O^{PrivateKey}$;
- E_3: The algorithm B dose not abort during the simulation of the oracle $O^{Certificate}$;
- E_4: The algorithm B dose not abort during the simulation of the oracle O^{Sign};
- *Abort*: The algorithm B aborts abnormally during the game.

We clearly have that the probability $\Pr[\neg Abort] = \Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4]$. From the above simulation, we can conclude that $\Pr[E_1] = \gamma/q_C$, $\Pr[E_2] = (1-1/q_C)^{q_K}$, $\Pr[\neg E_4] = (1-\gamma/q_C)^{q_S} \geq (1-\gamma)^{q_S}$ and E_1 implies E_3. Thus, $\Pr[\neg Abort] \geq \frac{\gamma}{q_C}(1-\frac{1}{q_C})^{q_K}(1-\gamma)^{q_S}$. Because $\gamma(1-\gamma)^{q_S}$ is maximized at $\gamma_{\max} = 1/(q_S+1)$, we get that $\Pr[\neg Abort]$ is at least $\frac{1}{q_C(1+q_S)e}(1-\frac{1}{q_C})^{q_K}$. Since the adversary A_I can forge the a valid signature with advantage ε only if the game is not terminated abnormally, the algorithm B can successfully solve the given weak modified q-CAA problem with advantage

$$\varepsilon' \geq \varepsilon \cdot \Pr[\neg Abort] \geq \frac{\varepsilon}{q_C(1+q_S)e}(1-\frac{1}{q_C})^{q_K}. \quad (5)$$

This completes the proof of Theorem 1. #

Theorem 2. If a Type-II adversary A_{II} has non-negligible advantage ε against our SCBS scheme, then there exists an algorithm B to solve the q-CAA problem over the group G with non-negligible advantage

$$\varepsilon' \geq \frac{\varepsilon}{q_C(q_S+1)e}, \quad (6)$$

where e denotes the base of natural logarithm, q_C and q_S respectively denote the maximal number of the adversary A_{II}'s queries to the oracles $O^{CreateUser}$ and O^{Sign} and $q \geq q_S$.

Proof. Assume that the algorithm B is given a random instance $(P, xP, (x+\omega_1)^{-1}P, (x+\omega_2)^{-1}P, \ldots, (x+\omega_q)^{-1}P, \omega_1, \ldots, \omega_q)$ of the q-CAA problem. It goal is to compute $(x+\omega^*)^{-1}abP$ for some value $\omega^* \in Z_p^* - \{\omega_1, \ldots, \omega_q\}$. In order to use the adversary A_{II} to solve the given problem, the algorithm B needs to simulate a challenger and all oracles for the adversary A_{II}.

Setup Phase. The algorithm B randomly chooses $s_1, s_2 \in Z_p^*$, computes $mpk_1 = s_1P$ and $mpk_2 = s_2P$. Additionally, it randomly chooses an index $I \in \{1, 2, \ldots, q_C\}$. It then outputs *params* =

$\{G, G_T, e, p, P, mpk_1, mpk_2, H_0, H_1, H_2\}$ and $msk = (s_1, s_2)$ to the adversary A_{II}, where $H_0 \sim H_2$ are three random oracles controlled by the algorithm B.

Query Phase. During the query phase, the adversary A_{II} can adaptively query the oracles H_0, H_1, H_2, $O^{CreateUser}$, $O^{PrivateKey}$ and O^{Sign}. The algorithm B handles these queries as follows:

(1) H_0 *queries*: The algorithm B maintains a list H_0List of tuples (ID_i, PK_i, Q_i, u_i). On receiving such a query on (ID_i, PK_i), the algorithm B returns Q_i directly if the list H_0List contains a tuple (ID_i, PK_i, Q_i, u_i). Otherwise, it randomly chooses $u_i \in Z_p^*$, computes $Q_i = u_iP$, inserts a new tuple (ID_i, PK_i, Q_i, u_i) into the list H_0List and returns Q_i.

(2) H_1 *queries*: The algorithm B maintains a list H_1List of tuples (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$). On receiving such a query on (M_i, ID_i, PK_i, mpk_1), the algorithm B returns α_i directly if the list H_1List contains a tuple (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$). Otherwise, it does the following:

- If $ID_i = ID_I$ (*i.e.*, ID_i is the *I-th* distinct identity submitted to the oracle $O^{CreateUser}$), it picks a random coin $coin_i \in \{0, 1\}$ with probability $\Pr[coin_i = 1] = \gamma$ for some value γ that will be determined later. If $coin_i = 0$, it randomly chooses a value $\omega_t \in \{\omega_1, \omega_2, \ldots, \omega_q\}$ that has not been selected before and sets $\alpha_i = \omega_t$; else if $coin_i = 1$, it randomly chooses $\alpha_i \in Z_p^*$ such that $\alpha_i \notin \{\omega_1, \omega_2, \ldots, \omega_q\}$. In either cases, the algorithm B inserts a new tuple (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$) into the list H_1List and returns α_i.
- Otherwise, it randomly chooses $\alpha_i \in Z_p^*$, inserts a new tuple (M_i, ID_i, PK_i, mpk_1, α_i, $\perp$) into the list H_1List and returns α_i.

(3) H_2 *queries*: The algorithm B maintains a list H_2List of tuples (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i). On receiving such a query on (M_i, ID_i, PK_i, mpk_1, mpk_2), the algorithm B returns β_i directly if the list H_2List contains a tuple (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i). Otherwise, it randomly chooses $\beta_i \in Z_p^*$, inserts a new tuple (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i) into H_2List and returns β_i.

(4) $O^{CreateUser}$ *queries*: The algorithm B maintains a list *UserList* of tuples (ID_i, SK_i, PK_i). On receiving such a query on an identity ID_i, the algorithm B returns PK_i directly if the list *UserList* contains a tuple (ID_i, SK_i, PK_i). Otherwise, it does the following:

- If $i = I$ (*i.e.*, ID_i is the *I-th* distinct identity submitted to this oracle), it sets $PK_i = xP$, inserts a new tuple (ID_i, $\perp$, PK_i) into the list *UserList* and returns PK_i;
- Otherwise, it randomly chooses $x_i \in Z_p^*$, sets $SK_i = x_i$ and $PK_i = x_iP$, inserts a new tuple (ID_i, SK_i, PK_i) into the list *UserList* and returns PK_i.

(5) $O^{PrivateKey}$ *queries*: On receiving such a query on an identity ID_i, the algorithm B aborts if $ID_i = ID_I$. Otherwise, it retrieves a tuple (ID_i, SK_i, PK_i) from the list *UserList* and returns SK_i.

(6) O^{Sign} *queries*: On receiving such a query on (M_i, ID_i), the algorithm B retrieves a tuple (ID_i, SK_i, PK_i) from the list *UserList*, a tuple (ID_i, PK_i, Q_i, u_i) from the list H_0List, a tuple (M_i, ID_i, PK_i, mpk_1, α_i, $coin_i$) from the list H_1List and a tuple (M_i, ID_i, PK_i, mpk_1, mpk_2, β_i) from the list H_2List respectively and does the following:

- If $ID_i = ID_I$ and $coin_i = 1$, it aborts the game;
- Else if $ID_i = ID_I$ and $coin_i = 0$, it computes a signature $\sigma_i = u_i(s_1 + \beta_i s_2)(x + \omega_t)^{-1}P$ and returns it to the adversary A_{II}. Obviously, σ_i is a valid signature of the message M_i because it can pass the verification $e(\sigma_i, \alpha_iP + PK_i) = e(Q_i, mpk_1 + \beta_i mpk_2)$, where $\alpha_i = \omega_t$, $PK_i = xP$ and $Q_i = u_iP$.
- Otherwise, it computes a signature $\sigma_i = (s_1 + \beta_i s_2)(SK_i + \alpha_i)^{-1}u_iP$ and returns it to the adversary A_{II}. Obviously, σ_i is a valid signature of the message M_i because it can pass the verification $e(\sigma_i, \alpha_iP + PK_i) = e(Q_i, mpk_1 + \beta_i mpk_2)$, where $PK_i = SK_iP$ and $Q_i = u_iP$.

Forge Phase. Finally, if the algorithm B does not abort, then the adversary A_{II} outputs a

forgery (ID_{ch}, M_{ch}, σ_{ch}). If $ID_{ch} \neq ID_I$, the algorithm B aborts the game. Otherwise, it retrieves a tuple (ID_{ch}, SK_{ch}, PK_{ch}) from the list *UserList* and a tuple (M_{ch}, ID_{ch}, PK_{ch}, mpk_1, α_{ch}, $coin_{ch}$) from the list H_1List respectively and does the following:

- If $coin_{ch} = 0$, it aborts;
- Else if $coin_{ch} = 1$ which implies $\alpha_{ch} \notin \{\omega_1, \omega_2, \ldots, \omega_q\}$, it retrieves a tuple ($ID_{ch}$, PK_{ch}, Q_{ch}, u_{ch}) from H_0List and a tuple (M_{ch}, ID_{ch}, PK_{ch}, mpk_1, mpk_2, β_{ch}) from H_2List respectively and computes $T = (u_{ch})^{-1}(s_1 + \beta_{ch}s_2)^{-1}\sigma_{ch}$ as the solution to the given q-CAA problem.

Notice that, if σ_{ch} is a valid signature for the message M_{ch} under the identity ID_{ch} and the public key PK_{ch}, then $\sigma_{ch} = u_{ch}(s_1 + \beta_{ch}s_2)(x + \alpha_{ch})^{-1}P$. Therefore, it is easy to deduce that $T = (x + \alpha_{ch})^{-1}P$ which is the correct solution to the given q-CAA problem.

Advantage analysis. To derive the algorithm B's advantage in solving the given problem, we define the following events:

- E_1: The adversary A_{II}'s forgery satisfies $ID_{ch} = ID_I$ and $coin_{ch} = 1$;
- E_2: The algorithm B dose not abort during the simulation of the oracle $O^{PrivateKey}$;
- E_3: The algorithm B dose not abort during the simulation of the oracle O^{Sign};
- *Abort*: The algorithm B aborts abnormally during the game.

We clearly have that the probability $\Pr[\neg Abort] = \Pr[E_1 \wedge E_2 \wedge E_3]$. From the above simulation, we can conclude that $\Pr[E_1] = \gamma/q_C$, $\Pr[\neg E_3] = (1-\gamma/q_C)^{q_S} \geq (1-\gamma)^{q_S}$ and E_1 implies E_2. Thus, $\Pr[\neg Abort] \geq \gamma(1-\gamma)^{q_S}/q_C$. Because $\gamma(1-\gamma)^{q_S}$ is maximized at $\gamma_{\max} = 1/(q_S+1)$, we get the probability $\Pr[\neg Abort]$ is at least $\frac{1}{q_C(q_S+1)e}$.

Since the adversary A_{II} can forge a valid signature with advantage ε only if the game is not terminated abnormally, the algorithm B can successfully solve the given q-CAA problem with advantage

$$\varepsilon' \geq \varepsilon \cdot \Pr[\neg Abort] \geq \frac{\varepsilon}{q_C(q_S+1)e}. \quad (7)$$

This completes the proof of Theorem 2. #

4.3 Performance comparison

To evaluate the performance of our scheme, we compare it with the previous four SCBS schemes [28-31] and some previous pairing-based CBS schemes [24-28].

As listed in **Table 1**, five basic cryptographic operations are considered in the computation comparison. They are bilinear pairing, exponentiation in the group G_T, scalar multiplication in the group G, map-to-point hash and general cryptographic hash. Note that if G is a multiplicative group, the scalar multiplication is then called exponentiation correspondingly. The cost of an algorithm is measured by the sum of the costs of all cryptographic operations. For example, to sign a message m, the algorithm ***Sign*** in our SCBS scheme requires computing two scalar multiplications in G and two general cryptographic hashes to generate a signature while the algorithm ***Verify*** requires computing two pairings, two scalar multiplications in G and three general cryptographic hashes to verify a signature. Therefore, the costs of the algorithms ***Sign*** and ***Verify*** are $2M + 2H$ and $2P + 2M + 3H$ respectively. Furthermore, the size of a signature is measured in terms of the total number of the group elements included. The

details of the compared schemes are shown in **Table 2** and **Table 3**. For ease of representation, we denote the SCBS schemes in [28-31] by WMSH-SCBS, LBZ-SCBS, LHZX-SCBS and HHT-SCBS respectively and the CBS schemes in [24-28, 33] by LHMSW-CBSa, LHMSW-CBSb, LBSZ-CBS, ZHANG-CBS, WMSH-CBS and LL-CBS respectively.

Table 1. Notations in the comparison

Notations	Description of the notations
P	Bilinear pairing
E	Exponentiation in G_T
M	Scalar multiplication in G
H_M	Map-to-Point hash
H	General cryptographic hash
$\|G\|$	Bit length of an element in G
$\|Z_p^*\|$	Bit length of an element in Z_p^*

Table 2. Comparison of our scheme and the previous SCBS schemes

Schemes	Sign	Verify	Signature	Security	Normal Framework
WMSH-SCBS	$1M + 1H_M$	$3P + 2H_M$	$\|G\|$	broken	yes
LBZ-SCBS	$1M + 1H$	$2P + 1M + H_M + H$	$\|G\|$	broken	yes
LHZX-SCBS	$1M + 1H$	$4P + 1M + H_M + H$	$\|G\|$	broken	yes
HHT-SCBS	$2M + 2H_M$	$3P + 1M + 2H_M + 1H$	$\|G\|$	EUF-CMA	no
Our scheme	$2M + 2H$	$2P + 2M + 3H$	$\|G\|$	EUF-CMA	yes

Table 3. Comparison of our scheme and some previous pairing-based CBS schemes

Schemes	Sign	Verify	Signature
LHMSW-CBSa	$3M + 2H$	$4P + 3H$	$2\|G\|$
LHMSW-CBSb	$9M + 2H$	$7P + 2M + 2H$	$5\|G\|$
LBSZ-CBS	$8M + 2H$	$4P + 2M + 2H$	$3\|G\|$
ZHANG-CBS	$5M + 2H$	$3P + 3M + 3H$	$2\|G\|$
WMSH-CBS	$1P + 1E + 3M + 2H$	$2P + 1E + 2M + 2H$	$2\|G\| + \|Z_p^*\|$
LL-CBS	$6M + 2H$	$3P + 2M + 2H$	$3\|G\|$
Our scheme	$2M + 2H$	$2P + 2M + 3H$	$\|G\|$

To give a more intuitive comparison, we test the running time of different schemes on a Lenovo L440 laptop running 64-bit Windows 7 with Interl(R) Core™ i7-4712MQ@2.3GHz and 8GB RAM memory. **Table 4** shows the benchmark time of different operations and the bit-length of different elements. The running time is obtained by computing the average of running the operation 10 times with random inputs using the PBC (Pairing-Based Cryptography) library (Version 0.5.14) [50]. In order to perform faster pairing and achieve the security level of 1024-bit RSA, we use the Type A pairing defined over the supersingular elliptic curve $E(F_q)$: $y^2 = x^3 + x$ with embedding degree 2, where the group size q is a 512-bit

prime satisfying $q + 1 = pr$ and p is a 160-bit Solinas prime. For the parameters p and r, we use the values (where r = 1201601226489114607938882136674053420480295440125131182291961513104720728935970453110284480218390653778677 6 and p = 73075081866545162136111924557150490140597655961 7) recommended by the PBC library. In addition, we use SHA-1 as the general cryptographic hash function. The running time and communication costs of the compared schemes are shown in **Table 5**.

Table 4. Benchmark time of different operations and bit-length of different elements

Operations/Elements	Running Time/Bit-Length
P	3.296 ms
E	0.404 ms
M	2.757 ms
H_M	6.236 ms
H	0.001 ms
$\|G\|$	512 bits
$\|Z_p^*\|$	160 bits

Table 5. Computation and communication costs of the compared schemes on Type-A curve

Schemes	Sign (ms)	Verify (ms)	Signature (bits)
WMSH-SCBS	8.99	22.36	512
LBZ-SCBS	2.76	15.59	512
LHZX-SCBS	2.76	22.18	512
HHT-SCBS	17.99	25.12	512
LHMSW-CBSa	8.27	13.19	1024
LHMSW-CBSb	24.81	28.58	2560
LBSZ-CBS	22.06	18.70	1536
ZHANG-CBS	13.79	18.16	1024
WMSH-CBS	11.97	12.51	1184
LL-CBS	16.54	15.40	1536
Our scheme	5.51	12.11	512

Our SCBS scheme requires about 5.51 ms to sign a randomly chosen message and 12.11 ms to verify a signature. Although our scheme is a little less efficient than the LBZ-SCBS scheme [29] and the LHZX-SCBS scheme [30] in the algorithm **Sign**, it outperforms the WMSH-SCBS scheme [28] and the HHT-SCBS scheme [31] in both the algorithms **Sign** and **Verify**. As for the signature length, it is 512 bits in our scheme, which is 50% of the LHMSW-CBSa scheme [24], 20% of the LHMSW-CBSb scheme [25], 33.3% of the LBSZ-CBS scheme [26], 50% of the ZHANG-CBS scheme [27], 43.2% of the WMSH-CBS scheme [28] and 33.3% of the LL-CBS scheme [33]. Compared with the previous pairing-based CBS schemes [24-28, 33], our scheme has obvious advantages in both the computation efficiency and the communication bandwidth.

5. Conclusions

In this paper, we first demonstrate that the SCBS scheme proposed by Wu *et al*. [28] is insecure against existential forgery attack. Then, we proposed a new SCBS scheme in the normal framework of CBS. The proposed scheme overcomes the security weaknesses in the previous SCBS schemes [28-30] and can provide resistance to the existential forgery attack. In the random oracle model, it is proven secure under the q-CAA problem and the weak modified q-CAA problem. The comparison shows that it outperforms the scheme proposed by Hung *et al*. [31] which is the only SCBS scheme achieving the EUF-CMA security. In addition, compared with the previous pairing-based CBS schemes [24-28, 33], it enjoys better performance on both the computation cost and the communication bandwidth. Due to the short signature length, the proposed SCBS scheme is particularly suitable for the low-bandwidth communication environments. A limitation of the scheme is that its security can only be achieved in the random oracle model. So, it would be interesting to construct provably secure SCBS schemes in the standard model.

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of Crypto 1984*, pp. 47-53, August 19-22, 1984. Article (CrossRef Link).
[2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. of Asiacrypt 2003*, pp. 452-473, November 30-December 4, 2003. Article (CrossRef Link).
[3] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Proc. of Eurocrypt 2003*, pp. 272-293, May 4-8, 2003. Article (CrossRef Link).
[4] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. of Crypto 2001*, pp. 213-229, August 19-23, 2001. Article (CrossRef Link).
[5] D.H. Yum and P.J. Lee, "Identity-based cryptography in public key management," in *Proc. of EuroPKI 2004*, pp.71-84, June 25-26, 2004. Article (CrossRef Link).
[6] D. Galindo, P. Morillo and C. Ràfols, "Breaking Yum and Lee generic constructions of certificateless and certificate-based encryption schemes," in *Proc. of EuroPKI 2006*, pp.81-91, June 19-20, 2006. Article (CrossRef Link).
[7] S.S. Al-Riyami and K.G. Paterson, "CBE from CL-PKE: a generic construction and efficient schemes," in *Proc. of PKC 2005*, pp. 398-415, January 23-26, 2005. Article (CrossRef Link).
[8] B.G. Kang and J.H. Park, "Is it possible to have CBE from CL-PKE?" *Cryptology ePrint Archive*, Report 2005/431. Article (CrossRef Link).
[9] W. Wu, Y. Mu, W. Susilo, X. Huang and L. Xu, "A provably secure construction of certificate-based encryption from certificateless encryption," *The Computer Journal*, vol. 55, no. 10, pp. 1157-1168, January, 2012. Article (CrossRef Link).
[10] D. Galindo, P. Morillo and C. Ràfols, "Improved certificate-based encryption in the standard model," *Journal of Systems and Software*, vol. 81, no. 7, pp. 1218-1226, July, 2008. Article (CrossRef Link).
[11] B. Waters, "Efficient identity-based encryption without random oracles," in *Proc. of Eurocrypt 2005*, pp. 114-127, May 22-26, 2005. Article (CrossRef Link).
[12] D. Boneh and X. Boyen, "Efficient selective-id secure identity based encryption without random oracles," in *Proc. of Eurocrypt 2004*, pp. 223-238, May 2-6, 2004. Article (CrossRef Link).
[13] J. K. Liu and J. Zhou, "Efficient certificate-based encryption in the standard model," in *Proc. of SCN 2008*, pp. 144-155, September 10-12, 2008. Article (CrossRef Link).
[14] Y. Lu and J. Li, "Efficient construction of certificate-based encryption secure against public key replacement attacks in the standard model," *Journal of Information Science and Engineering*, vol. 30, no. 5, pp. 1553-1568, September, 2014. Article (CrossRef Link).

[15] Q. Yu, J. Li and Y. Zhang, "Leakage-resilient certificate-based encryption," *Security and Communication Networks*, vol. 8, no, 18, pp. 3346-3355, May, 2015. Article (CrossRef Link).
[16] Y. Lu and Q. Zhang, "Enhanced certificate-based encryption scheme without bilinear pairings," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 2, pp. 881-896, February, 2016. Article (CrossRef Link).
[17] Q. Yu, J. Li, Y. Zhang, W. Wu, X. Huang and Y. Xiang, "Certificate-based encryption resilient to key leakage," *Journal of Systems and Software*, vol. 116, pp. 101-112, June, 2016. Article (CrossRef Link).
[18] J. Li, Y. Guo, Q. Yu, Y. Lu, Y. Zhang and F. Zhang, "Continuous leakage-resilient certificate-based encryption," *Information Sciences*, vol. 355-356, pp. 1-14, August, 2016. Article (CrossRef Link).
[19] Y. Lu and J. Li, "A provably secure certificate-based encryption scheme secure against malicious CA attacks in the standard model," *Information Sciences*, vol. 372, pp. 745-757, December, 2016. Article (CrossRef Link).
[20] C. Sur, C. D. Jung and K. H. Rhee, "Multi-receiver certificate-based encryption and application to public key broadcast encryption," in *Proc. of 2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security*, pp. 35-40, August 4-6, 2007. Article (CrossRef Link).
[21] L. Wang, J. Shao, Z. Cao, M. Mambo and A. Yamamura, "A certificate-based proxy cryptosystem with revocable proxy decryption power," in *Proc. of Indocrypt 2007*, pp. 297-311, December 9-13, 2007. Article (CrossRef Link).
[22] Y. Lu and J. Li, "A pairing-free certificate-based proxy re-encryption scheme for secure data sharing in public clouds," *Future Generation Computer Systems*, vol. 62, pp. 140-147, September, 2016. Article (CrossRef Link).
[23] B. G. Kang, J. H. Park and S. G. Hahn, "A certificate-based signature scheme," in *Proc. of Topics in Cryptology - CT-RSA 2004*, pp. 99-111, February 23-27, 2004. Article (CrossRef Link).
[24] J. Li, X. Huang, Y. Mu, W. Susilo and Q. Wu, "Certificate-based signature: security model and efficient construction," in *Proc. of EuroPKI 2007*, pp. 110-125, June 28-30, 2007. Article (CrossRef Link).
[25] J. Li, X. Huang, Y. Mu, W. Susilo and Q. Wu, "Constructions of certificate-based signature secure against key replacement attacks," *Journal of Computer Security*, vol. 18, no. 3, pp. 421-449, August, 2010. Article (CrossRef Link).
[26] J. K. Liu, J. Baek, W. Susilo, and J. Zhou, "Certificate based signature schemes without pairings or random oracles," in *Proc. of ISC 2008*, pp. 285-297, September 15-18, 2008. Article (CrossRef Link).
[27] J. Zhang, "On the security of a certificate-based signature scheme and its improvement with pairings," in *Proc. of ISPEC 2009*, pp. 47-58, April 13-15, 2009. Article (CrossRef Link).
[28] W. Wu, Y. Mu, W. Susilo, X. Huang, "Certificate-based signatures, revisited," *Journal of Universal Computer Science*, vol. 15, no. 8, pp. 1659-1684, April, 2009. Article (CrossRef Link).
[29] J.K. Liu, F. Bao and J. Zhou, "Short and efficient certificate-based signature," in *Proc. of Networking 2011 Workshops*, pp. 167-178, May 13, 2011. Article (CrossRef Link).
[30] J. Li, X. Huang, Y. Zhang and L. Xu, "An Efficient short certificate-based signature scheme," *Journal of Systems and Software*, vol. 85, no. 2, pp. 314-322, February, 2012. Article (CrossRef Link).
[31] Y.H. Hung, S.S. Huang and Y.M. Tseng, "A short certificate-based signature scheme with provable security," *Information Technology and Control*, vol. 45, no. 3, pp. 243-253, March, 2016. Article (CrossRef Link).
[32] J. Li, Z. Wang and Y. Zhang, "Provably secure certificate-based signature scheme without pairings," *Information Science*, vol. 233, pp. 313-320, June, 2013. Article (CrossRef Link).
[33] Y. Lu and J. Li, "An improved certificate-based signature scheme without random oracles," *IET Information Security*, vol. 10, no. 2, pp. 80-86, February, 2016. Article (CrossRef Link).
[34] M.H. Au, J.K. Liu, W. Susilo and T.H. Yuen, "Certificate based (linkable) ring signature," in *Proc. of ISPEC 2007*, pp. 79-92, May 7 - 10, 2007. Article (CrossRef Link).

[35] J. K. Liu, J. Baek and J. Zhou, "Certificate-based sequential aggregate signature," in *Proc. of the 2nd ACM Conference on Wireless Network Security*, pp. 21-28, March 16 - 19, 2009. Article (CrossRef Link).
[36] J. Li, H. Du and Y. Zhang, "Certificate-based key-insulated signature in the standard model," *The Computer Journal*, vol. 59, no. 7 pp. 1028-1039, July, 2016. Article (CrossRef Link).
[37] K. Barr and K. Asanovic, "Energy-aware lossless data compression," *ACM Transactions on Computer Systems*, vol. 24, no. 3, pp. 250-291, August, 2006. Article (CrossRef Link).
[38] D. Boneh, B. Lynn and H. Shacham, "Short signatures from the Weil pairing," in *Proc. of Asiacrypt 2001*, pp. 514-533, December 9-13, 2001. Article (CrossRef Link).
[39] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Proc. of Eurocrypt 2004*, pp.56-73, May 2-6, 2004. Article (CrossRef Link).
[40] F. Zhang, R. Safavi-Naini and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Proc. of PKC 2004*, pp. 277-290, March 1-4, 2004. Article (CrossRef Link).
[41] X. Huang, W. Susilo, Y. Mu and F. Zhang, "Short designated verifier signature scheme and its identity-based variant," *International Journal of Network Security*, vol. 6, no. 1, pp. 82-93, January, 2008. Article (CrossRef Link).
[42] L. Zhang, Y. Hu and Q. Wu, "New identity-based short signature without random oracles," *Procedia Engineering*, vol. 15, pp. 3445-3449, 2011. Article (CrossRef Link).
[43] X. Huang, Y. Mu, W. Susilo, D.S. Wong and W. Wu, "Certificateless signature revisited," in *Proc. of ACISP 2007*, pp. 308-322, July 2-4, 2007. Article (CrossRef Link).
[44] R. Tso, X. Yi and X. Huang, "Efficient and short certificateless signatures secure against realistic adversaries," *Journal of Supercomputing*, vol. 55, no. 2, pp. 173-191, February, 2011. Article (CrossRef Link).
[45] R. Tso, X. Huang and W. Susilo, "Strongly secure short certificateless signatures," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1409-1417, June, 2012. Article (CrossRef Link).
[46] L. Cheng, Y. Xiao and G. Wang, "Cryptanalysis of a certificate-based on signature scheme," *Procedia Engineering*, vol. 29, no. 4, pp. 2821-2825, February, 2012. Article (CrossRef Link).
[47] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proc. of ACMCCS 1993*, pp. 62-73, November 3-5, 1993. Article (CrossRef Link).
[48] R. Canetti, O. Goldreich and S. Halevi, "The random oracle methodology, revisited," *Journal of ACM*, vol. 51, no. 4, pp. 209-218, July, 2004. Article (CrossRef Link).
[49] S. Mitsunari, R. Sakai and M. Kasahara, "A new traitor tracing," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E85-A, no.2, pp. 481-484, February, 2002. Article (CrossRef Link).
[50] B. Lynn, "PBC library: The pairing-based cryptography library," http://crypto.stanford.edu/pbc/. Article (CrossRef Link).

Yang Lu received the Ph.D. degree from PLA University of Science and Technology, Nanjing, China, in 2009. Since 2003, he has been working in HoHai University, Nanjing, China. Currently, he is an Assistant Professor in College of Computer and Information, HoHai University, Nanjing, China. His major research interests include information security and cryptography, network security and cloud security, etc. He has published more than 50 scientific papers in international conferences and journals.

Gang Wang has been studying in HoHai University, Nanjing, China, from 2014. Currently, he is a postgraduate student in College of Computer and Information, HoHai University, Nanjing, China. His research interest is cryptography.

Jiguo Li received the Ph.D. degree from Harbin Institute of Technology in 2003. He has been working in HoHai University from 2003. Currently, he is a Professor in College of Computer and Information. His major research interests include information security and cryptography, network security, wireless security and trusted computing etc. He has published more than 70 scientific papers and two books. He has served as a PC member of several international conferences and the reviewer of some international journals and conferences.

Jian Shen received the BE degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2007 and the ME and PhD degrees in Computer Science from Chosun University, Gwangju, Korea, in 2009 and 2012, respectively. Since 2012, he has been a full professor in the School of Computer and Software at Nanjing University of Information Science and Technology, Nanjing, China. His research interests include information and network security, security systems, public-key cryptography, etc.