A Dual-KGC Based Certificateless Short Signature Scheme Postprint

Authors: Zuo Liming, Zhang Mengli, Hu Kaiyu, Yi Chuanjia

Date: 2019-04-01T00:00:00+00:00

Abstract

To address the issue of excessive power concentration in a single KGC in certificateless short signature schemes, a certificateless short signature scheme based on dual KGCs is proposed, wherein the dual KGCs mutually restrict each other, effectively reducing the harm caused by master key leakage and malicious manipulation of a single KGC. Subsequently, under the random oracle model and the hardness assumptions of the k-CAA and Inv-CDH problems, it is proven that the signature scheme is existentially unforgeable under adaptive chosen-message attacks. Finally, comparisons are made with other certificateless digital signature schemes, and the scheme is implemented in C language. Experimental results and analysis demonstrate that the scheme has low computational overhead, high operational efficiency, and strong security.

Full Text

Preamble

Vol. 37 No. 5
Application Research of Computers
ChinaXiv Partner Journal

Certificateless Short Signature Scheme with Double KGC

Zuo Liming[1] , Zhang Mengli[1] , Hu Kaiyu[1] , Yi Chuanjia[1]

[1]School of Science, [2]SEC Institute, East China Jiaotong University, Nanchang 330013, China

Abstract: To address the problem of excessive power concentration in a single Key Generation Center (KGC) in certificateless short signature schemes, this paper proposes a certificateless short signature scheme based on double KGC, where the two KGCs mutually restrict each other, effectively reducing

the harm caused by the leakage of a single KGC' s master key and malicious manipulation. Subsequently, under the random oracle model and the hardness assumptions of the k-CAA and Inv-CDH problems, we prove that the proposed signature scheme is existentially unforgeable against adaptive chosen-message attacks. Finally, we compare the scheme with other certificateless digital signature schemes and implement it in C language. Experimental results and analysis demonstrate that the scheme has low computational cost, high operational efficiency, and strong security.

Keywords: certificateless; double KGC; short signature; provably secure; random oracle model

0 Introduction

To solve the time-consuming issues of user public key certificate management and certificate transmission in traditional public key cryptography, Shamir [**?**] proposed identity-based cryptography in 1984, where all users' private keys are generated by a trusted third party called a Private Key Generator (PKG). Consequently, a malicious PKG can impersonate users to generate signatures or decrypt ciphertexts intended for users, leading to a key escrow problem. To address this key escrow issue in identity-based cryptography, Al-Riyami and Paterson [**?**] introduced a certificateless public key cryptography in 2003. In certificateless public key cryptography, a user's private key is jointly generated by the Key Generation Center (KGC) and the user, which eliminates the key escrow problem in identity-based cryptography while removing the certificate storage and management issues in traditional cryptography. Subsequently, numerous certificateless digital signature schemes have been proposed. In 2012, Feng et al. [**?**] proposed an efficient certificateless multi-signature scheme in the standard model. In 2016, Islam et al. [**?**] constructed a certificateless digital signature scheme based on bilinear pairings. In 2017, Gao et al. [**?**] proposed a leakage-free certificateless digital signature scheme based on the Computational Diffie-Hellman (CDH) problem.

In certificateless cryptosystems, there are two types of adversaries [**?**]: A_1 and A_2. One is the dishonest user A_1, who does not know the system master key or the user' s partial private key but can replace the user' s public key. The other is the malicious KGC A_2, who knows the system master key and the user' s partial private key but cannot replace the user' s public key.

Existing certificateless signature schemes based on a single KGC cannot resist malicious KGC attacks. In 2017, Zhang et al. [**?**] analyzed the security of certificateless aggregate signature schemes proposed in [**?**, **?**, **?**], pointing out that these three schemes suffer from malicious KGC attacks, and constructed attack algorithms respectively to achieve forgery attacks. In the same year, Ge et al. [**?**] analyzed a certificateless proxy signature scheme proposed by Liu and Zhang [**?**] and found that it could not resist malicious KGC attacks.

Therefore, it is evident that single-KGC certificateless signature schemes pose certain risks to the system due to the excessive power of the KGC, as a malicious KGC can master the system master key and users' partial private keys to forge signatures. To solve the problem of excessive power concentration in a single KGC, this paper proposes a provably secure certificateless short signature scheme based on double KGC under the random oracle model. Compared with other single-KGC certificateless signature schemes, on the one hand, the double KGCs can mutually restrict each other, achieving a separation of powers that reduces the success probability of attacks targeting the KGC and thus mitigates harm to the system. On the other hand, by incorporating the advantages of short signatures [**?**], the practicality of our scheme is enhanced.

1.1 Bilinear Pairings

Bilinear Pairings [**?**, **?**]: Let G_1 be an additive cyclic group of order q, and G_2 be a multiplicative cyclic group of order q. Let P be a generator of G_1. A mapping $e : G_1 \times G_1 \to G_2$ is called a bilinear pairing if it satisfies the following three properties:

a) **Bilinearity**: For any $m, n \in \mathbb{Z}_q^*$, $e(mP, nP) = e(P, P)^{mn}$.

b) **Non-degeneracy**: $e(P, P) \neq 1$.

c) **Computability**: There exists a polynomial-time algorithm to compute e.

1.2 Hard Problem Assumptions

Definition 1 (k-CAA Problem). Given $(P, sP, e_1, \ldots, e_k \in_R \mathbb{Z}_q^*)$ where s is unknown, and given $\left\{\frac{1}{s+e_1}P, \ldots, \frac{1}{s+e_k}P\right\}$, for a certain integer $c \notin \{e_1, \ldots, e_k\}$ and $c \in \mathbb{Z}_q^*$, compute $\frac{1}{s+c}P$.

Definition 2 (Inverse Computational Diffie-Hellman Problem, Inv-CDH). Given (P, aP) where $a \in \mathbb{Z}_q^*$ is unknown, compute $a^{-1}P$.

1.3 Definition of Certificateless Digital Signature with Double KGC

A certificateless digital signature scheme based on double KGC consists of seven algorithms, as shown in Figure 1 [Figure 1: see original paper].

Figure 1 Certificateless digital signature definition

1. **Setup**: Input a security parameter L. KGC_A runs the algorithm and returns its master key x_A. KGC_B runs the algorithm and returns its master key x_B and system public parameters $params$.

2. **Extract-Partial-Private-Key**: Input $params$, KGC_A' s master key x_A, and identity ID. KGC_A runs this algorithm and outputs KGC_A' s partial private key d_A. Input $params$, KGC_B' s master key x_B, and KGC_A' s partial private key d_A. KGC_B runs this algorithm and outputs the partial private key d_{ID}, which is sent to the corresponding user via a secure channel.

3. **Set-Secret-Value**: Input $params$ and user identity ID. Output the user' s secret value x_{ID}. This algorithm is executed by each user in the system. The secret value space is determined by the system public parameters $params$ and the user' s identity ID.

4. **Set-Private-Key**: Input $params$, the user' s partial private key d_{ID}, and their secret value. Output the private key sk_{ID}. This algorithm is executed by each user in the system.

5. **Set-Public-Key**: Input $params$ and the user' s secret value x_{ID}. Output the user' s public key pk_{ID}. This algorithm is executed by system users, and the public key is published after execution. The public key space is defined by the system public parameters $params$ and the user' s identity information ID.

6. **CL-Sign**: Input $params$, message m to be signed, user' s identity ID, public key pk_{ID}, and private key sk_{ID}. This algorithm outputs signature S.

7. **CL-Verify**: This is a deterministic signature verification algorithm. Input $params$, signer' s identity ID, public key pk_{ID}, message m, and signature S. It returns "1" if the signature is valid, otherwise returns "0" indicating the signature is invalid.

1.4 Security Model of Certificateless Digital Signature with Double KGC

In certificateless cryptosystems, there are two types of attacks. On one hand, because user public keys are not certified, adversaries have the right to replace a user' s public key with an illegal one of their choice, but they do not know the system master key or the user' s partial private key—this is the user public key replacement attack. On the other hand, since the KGC knows the system master key, it can compute the user' s partial private key but cannot replace the user' s public key—this is the malicious but passive KGC attack. Therefore, a secure certificateless cryptographic scheme should at least resist these two attacks.

Based on the traditional single-KGC attack types and combining the characteristics of double KGC, this paper presents in detail two types of adversaries with different capabilities for our scheme. One is the dishonest user, denoted as A_1, and the other is the malicious but passive KGC, denoted as A_2.

- **Type I Adversary** A_1: Knows at most one system master key, does not know the user' s partial private key, and can replace the user' s public key.
- **Type II Adversary** A_2: Masters the system master key, knows the user' s partial private key, but cannot replace the user' s public key.

Definition 3: A certificateless digital signature scheme is existentially unforgeable against adaptive chosen-message attacks if the probability of adversary A winning in the following two games is negligible.

The interaction between challenger C and Type I adversary A_1 is shown in Figure 2 [Figure 2: see original paper]. Finally, A_1 outputs a challenge identity ID^* and public key pk_{ID^*} and a message/signature pair (m^*, S^*). A_1 wins if the following conditions hold:

a) ID^* has never been submitted to the private key extraction oracle;

b) ID^* has never been submitted to both the public key replacement oracle and the partial private key extraction oracle simultaneously;

c) (m^*, S^*) is not obtained from the signature oracle.

Figure 2 Game process between Challenger C and Type I adversary

Figure 3 Game process between Challenger C and Type II adversary

The interaction between challenger C and Type II adversary A_2 is shown in Figure 3 [Figure 3: see original paper]. Finally, A_2 outputs a challenge identity ID^* and public key pk_{ID^*} and a message/signature pair (m^*, S^*). A_2 wins if the following conditions hold:

a) ID^* has never been submitted to the private key extraction oracle;

b) (m^*, S^*) is not obtained from the signature oracle.

2.1 Scheme Construction

a) Setup: Given security parameter l, G_1 and G_2 are additive and multiplicative cyclic groups respectively, P is a generator of G_1, and the order of G_1 is q. The bilinear pairing is $e : G_1 \times G_1 \to G_2$. Select two collision-resistant hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \times G_1 \to \mathbb{Z}_q$.

KGC_A selects a secret value $x_A \in_R \mathbb{Z}_q^*$ as KGC_A' s master key, computes $y_A = x_A P$, and secretly saves x_A.

KGC_B selects a secret value $x_B \in_R \mathbb{Z}_q^*$ as KGC_B' s master key, computes $y_B = x_B P$, and secretly saves x_B.

Compute $T = x_A y_B = x_A x_B P = x_B y_A$ as the joint public key and publish it. Anyone can verify the validity of T through $e(T, P) = e(y_A, y_B)$.

Publish the joint system parameters $\{G_1, G_2, e, q, P, H_1, H_2, y_A, y_B, T\}$.

Let p be the probability of single KGC key leakage, then the probability of key leakage in our constructed double KGC scheme is p^2. Therefore, if the single-KGC-based scheme is secure, then the double-KGC-based scheme is also secure.

Theorem 1: Under the random oracle model and the k-CAA and Inv-CDH problem assumptions, the proposed certificateless short signature scheme based on double KGC is existentially unforgeable against adaptive chosen-message attacks.

Theorem 1 can be derived from Lemmas 1-3 below.

Lemma 1: For Type I adversary, suppose there exists an adaptive chosen-message and signature attack algorithm A_1 that breaks our scheme with non-negligible advantage ε in probabilistic polynomial time t, where A_1 does not know KGC_A' s system master key but knows KGC_B' s system master key. Let q_H, q_E, q_{pk}, q_s denote the number of H_1 queries, partial private key extraction queries, private key extraction queries, public key queries, and signature queries respectively. Let t_H, t_E, t_{pk}, t_s denote the time required for one H_1 oracle query, one partial private key extraction oracle, one private key extraction query, one public key query, and one signature query respectively. Then there exists an algorithm C that solves the k-CAA problem with advantage $\varepsilon' \geq \frac{1}{q_H}\left(1-\frac{1}{q_H}\right)^{q_E+q_s}\left(1-\frac{1}{q_{pk}}\right)^{q_s}\varepsilon$ and running time $t' \leq t + (2q_E + 2q_{pk} + q_s)t_H + q_E t_E + q_{pk} t_{pk} + q_s t_s$.

Proof: Given a k-CAA problem instance: $\{P, sP, e_1, \ldots, e_k \in_R \mathbb{Z}_q^*\}$ and $\left\{\frac{1}{s+e_1}P, \ldots, \frac{1}{s+e_k}P\right\}$ where s is unknown, algorithm C' s goal is to call A_1 as a subroutine and finally output a solution to the k-CAA problem.

In the game, assume that for any user identity ID, the signature attack algorithm A_1 has queried H_1 and H_2 about the user identity before performing partial private key extraction queries, private key extraction queries, public key replacement, signature queries, and outputting signatures.

System Setup: Let $y_A = sP$ (here s acts as KGC_A' s system master key, which is unknown to C). Select $x_B \in_R \mathbb{Z}_q^*$ as KGC_B' s master key, compute $y_B = x_B P$ as KGC_B' s public key, compute $T = x_B y_A = x_B sP$ as the system' s joint public key (known to C), and send the system joint parameters $\{G_1, G_2, e, q, P, H_1, H_2, y_A, y_B, T\}$ and KGC_B' s master key x_B to A_1.

Then attack algorithm A_1 adaptively executes the following queries to challenger C:

a) H_1 **Queries**: C maintains a list $list_H$ consisting of entries (ID_i, Q_i). C prepares q_H responses $\{e_1, \ldots, e_k, \ldots, e_{q_H}\}$ where the data $\{e_1, \ldots, e_k\}$ are randomly distributed in the set. When A_1 makes an H_1 query about identity ID_i, C performs:

- If $ID_i = ID^*$, C selects one e_i from the list and returns $Q_i = e_i$ to A_1.
- Otherwise, C randomly selects a value e_i from set $\{e_{k+1}, \ldots, e_{q_H}\}$ and returns $Q_i = e_i$ to A_1, then adds record (ID_i, Q_i) to $list_H$.

b) Partial Private Key Extraction Queries: C maintains a list $list_E$ consisting of entries (ID_i, ID_i, Q_i, d_i). When A_1 makes a partial private key query about identity ID_i, C recovers Q_i from $list_H$ and performs:

- If $ID_i = ID^*$, C stops the simulation and outputs "FAILURE" (this event is denoted as E_1).
- Otherwise, C computes $d_i = \frac{1}{x_B + Q_i} P$ and returns d_i to A_1, then adds record (ID_i, ID_i, Q_i, d_i) to $list_E$.

c) Public Key Queries: When A_1 makes a public key query about identity ID_i, C checks if a corresponding value exists in the list. If it exists, C recovers pk_i and returns it. Otherwise, C selects a random number $r_i \in_R \mathbb{Z}_q^*$ and returns public key $pk_i = r_i P$ to A_1. Then C adds record $(ID_i, ID_i, Q_i, pk_i, r_i)$ to $list_{pk}$.

d) Private Key Extraction Queries: When A_1 makes a private key extraction query about identity ID_i, C first recovers (ID_i, ID_i, Q_i, d_i) from $list_E$ and $(ID_i, ID_i, Q_i, pk_i, r_i)$ from $list_{pk}$, then performs:

- If $ID_i = ID^*$, C stops the protocol and outputs "FAILURE" (this case is denoted as E_2).
- Otherwise, C runs the record (ID_i, ID_i, Q_i, d_i) and record $(ID_i, ID_i, Q_i, pk_i, r_i)$ and sends $sk_i = (d_i, r_i)$ as the corresponding private key to A_1.

e) Public Key Replacement: C inputs (ID_i, pk_i'), modifies $list_{pk}$, and replaces user ID_i' s public key pk_i with pk_i'.

f) H_2 **Queries**: C maintains a list $list_H$ consisting of entries $(ID_i, ID_i, m_i, Q_i, pk_i, h_i)$. When A_1 makes an H_2 query, C randomly selects $h_i \in_R \mathbb{Z}_q^*$. Then C adds record $(ID_i, ID_i, m_i, Q_i, pk_i, h_i)$ to $list_H$ and returns h_i to A_1.

g) Signature Queries: When A_1 makes a signature query about (ID_i, m_i), C first recovers (ID_i, ID_i, Q_i, d_i) from $list_E$, then performs:

- If $ID_i \neq ID^*$, C extracts record (ID_i, ID_i, Q_i, d_i) from $list_E$, extracts record $(ID_i, ID_i, Q_i, pk_i, r_i)$ from $list_{pk}$, extracts record $(ID_i, ID_i, m_i, Q_i, pk_i, h_i)$ from $list_H$, and computes $S_i = d_i + h_i r_i$.
- Otherwise, C stops the protocol and outputs "FAILURE" (this case is denoted as E_3).

Finally, C outputs signature S_i.

Finally, A_1 stops training and outputs a message/signature pair (m^*, S^*) for a

challenge identity ID^* and public key pk_{ID^*}, satisfying the verification equation $Verify(params, ID^*, m^*, pk_{ID^*}, S^*) = 1$. C extracts the corresponding record $(ID^*, ID^*, Q^*, pk^*, r^*)$ from $list_{pk}$ and record $(ID^*, ID^*, m^*, Q^*, pk^*, h^*)$ from $list_H$, then performs:

- If $\{Q^*\} \notin \{e_1, \ldots, e_k\}$, C outputs "FAILURE" and stops the protocol (this event is denoted as E_4).
- Otherwise, the following equation holds:

$$\begin{aligned} e(S^*, P) &= e\left(h^* r^* P + \frac{1}{x_B + Q^*} P, P\right) \\ &= e\left(h^* r^* P + \frac{1}{x_B + Q^*} P, P\right) \\ &= e\left(\frac{1}{x_B + Q^*} P, P\right) \cdot e(h^* r^* P, P) \\ &= e\left(\frac{1}{s + Q^*} P, P\right) \end{aligned}$$

Thus C can successfully compute $\frac{1}{s+Q^*}P$ as the solution to the k-CAA problem.

Analysis of A_1' s advantage in this game:

a) The responses to A_1' s H_1 and H_2 queries are uniformly distributed, indistinguishable from the real environment.

b) Partial private key extraction queries, private key extraction queries, and signature queries can proceed smoothly without stopping, i.e., events E_1, E_2, E_3 do not occur.

c) If events E_1, E_2, E_3, and E_4 do not occur, then C can solve an instance of the k-CAA problem. The probability that events E_1, E_2, E_3, and E_4 do not occur satisfies:

$$\Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3 \wedge \neg E_4] \geq \left(1 - \frac{1}{q_H}\right)^{q_E + q_s} \left(1 - \frac{1}{q_{pk}}\right)^{q_s} \frac{1}{q_H}$$

However, when A_1 forges a valid signature without querying H_2, this simulation has a flaw. Considering that the oracle output satisfies uniform distribution, the probability of this event is $\frac{1}{q}$, so A_1' s advantage in this game is $\varepsilon' \geq \frac{1}{q_H}\left(1 - \frac{1}{q_H}\right)^{q_E+q_s}\left(1 - \frac{1}{q_{pk}}\right)^{q_s}\varepsilon$, and the running time is $t' \leq t + (2q_E + 2q_{pk} + q_s)t_H + q_E t_E + q_{pk} t_{pk} + q_s t_s$.

Lemma 2: For Type I adversary, suppose there exists an adaptive chosen-message and signature attack algorithm A_1 that breaks our scheme with non-negligible advantage ε in probabilistic polynomial time t, where A_1 knows neither KGC_A' s nor KGC_B' s system master key. Let q_H, q_E, q_{pk}, q_s denote the

number of H_1 queries, partial private key extraction queries, private key extraction queries, public key queries, and signature queries respectively. Let t_H, t_E, t_{pk}, t_s denote the time required for one H_1 oracle query, one partial private key extraction oracle, one private key extraction query, one public key query, and one signature query respectively. Then there exists an algorithm C that solves the k-CAA problem with advantage $\varepsilon' \geq \frac{1}{q_H}\left(1-\frac{1}{q_H}\right)^{q_E+q_s}\left(1-\frac{1}{q_{pk}}\right)^{q_s}\varepsilon$ and running time $t' \leq t + (2q_E + 2q_{pk} + q_s)t_H + q_E t_E + q_{pk} t_{pk} + q_s t_s$.

Proof: Similar to the proof of Lemma 1, omitted here due to space limitations.

Lemma 3: For Type II adversary, suppose there exists an adaptive chosen-message and signature attack algorithm A_2 that breaks our scheme with non-negligible advantage ε in probabilistic polynomial time t. Let q_H, q_{pk}, q_s denote the number of H_2 queries, private key extraction queries, public key queries, and signature queries respectively. Let t_H, t_{pk}, t_s denote the time required for one H_2 query, one private key extraction query, one public key query, and one signature query respectively. Then there exists an algorithm C that solves the Inv-CDH problem with advantage $\varepsilon' \geq \frac{1}{q_H}\left(1-\frac{1}{q_{pk}}\right)^{q_s}\varepsilon$ and running time $t' \leq t + (2q_{pk} + q_s)t_H + q_{pk} t_{pk} + q_s t_s$.

Proof: Given an Inv-CDH problem instance: (P, hP) where $h \in_R \mathbb{Z}_q^*$ is unknown, C' s goal is to output $h^{-1}P$ through interaction with A_2.

System Setup: C runs the system setup algorithm, selects $x_A \in_R \mathbb{Z}_q^*$ as KGC_A' s master key, computes $y_A = x_A P$ as KGC_A' s public key (known to C). C selects $x_B \in_R \mathbb{Z}_q^*$ as KGC_B's master key, computes $y_B = x_B P$ as KGC_B's public key, computes $T = x_A y_B = x_A x_B P$ as the system joint public key (known to C), and sends the system joint parameters $\{G_1, G_2, e, q, P, H_1, H_2, y_A, y_B, T\}$ and KGC_A' s master key x_A and KGC_B' s master key x_B to A_2.

Then attack algorithm A_2 adaptively executes the following queries to challenger C:

a) H_2 Queries: C maintains a list $list_H$ consisting of entries $(ID_i, ID_i, m_i, Q_i, pk_i, h_i)$. When A_2 makes an H_2 query about identity ID_i, if the query value already exists in the list, C returns the corresponding value from the list. Otherwise, C performs:

- If $ID_i = ID^*$, C selects a random number $h_i \in_R \mathbb{Z}_q^*$, returns h_i to A_2, and adds record $(ID_i, ID_i, m_i, Q_i, pk_i, h_i)$ to $list_H$.
- Otherwise, C randomly selects a value h_i from set $\{h_1, \ldots, h_{q_H}\}$ and returns h_i to A_2.

b) Private Key Extraction Queries: When A_2 makes a private key query about ID_i, C performs:

- If $ID_i = ID^*$, C stops the protocol and outputs "FAILURE" (this case is denoted as E_1).

- Otherwise, C extracts the corresponding record (ID_i, ID_i, Q_i, d_i) from $list_E$ and $(ID_i, ID_i, Q_i, pk_i, r_i)$ from $list_{pk}$ and sends them to A_2.

c) Public Key Queries: C maintains a list $list_{pk}$ consisting of entries $(ID_i, ID_i, Q_i, pk_i, r_i)$. When A_2 makes a public key query about ID_i, C checks if a corresponding value exists in the list. If it exists, C outputs the corresponding value. Otherwise, C extracts record (ID_i, ID_i, Q_i) from $list_H$ and performs:

- If $ID_i = ID^*$, C selects a random number $r_i \in_R \mathbb{Z}_q^*$, returns $pk_i = r_i P$ to A_2, and adds record $(ID_i, ID_i, Q_i, pk_i, r_i)$ to $list_{pk}$.
- Otherwise, C selects a random number $r_i \in_R \mathbb{Z}_q^*$, returns $pk_i = r_i P$, computes $pk_i = r_i P$ and sends it to A_2. Then C adds record $(ID_i, ID_i, Q_i, pk_i, r_i)$ to $list_{pk}$.

d) Signature Queries: When A_2 makes a signature query about (ID_i, m_i), C performs:

- If $ID_i = ID^*$, C stops the protocol and outputs "FAILURE" (this case is denoted as E_2).
- Otherwise, C extracts (ID_i, ID_i, Q_i, d_i) from $list_E$, extracts $(ID_i, ID_i, Q_i, pk_i, r_i)$ from $list_{pk}$, extracts $(ID_i, ID_i, m_i, Q_i, pk_i, h_i)$ from $list_H$, and computes $S_i = d_i + h_i r_i P$.

Finally, C outputs signature S_i.

Finally, A_2 stops querying and outputs a message/signature pair (m^*, S^*) for a challenge identity ID^*, satisfying the verification equation $Verify(m^*, ID^*, pk_{ID^*}, S^*) = 1$. The corresponding public key is $pk_{ID^*} = r^* P$. C extracts the corresponding record $(ID^*, ID^*, Q^*, pk^*, r^*)$ from $list_{pk}$ and record $(ID^*, ID^*, m^*, Q^*, pk^*, h^*)$ from $list_H$, then performs:

- If $\{Q^*\} \notin \{e_1, \ldots, e_k\}$, C outputs "FAILURE" and stops the protocol (this event is denoted as E_3).
- Otherwise, the following equation holds:

$$\begin{aligned} e(S^*, P) &= e\left(h^* r^* P + \frac{1}{x_A + Q^*} P, P\right) \\ &= e\left(\frac{1}{h} P, P\right) \end{aligned}$$

Thus C can successfully compute $\frac{1}{h} P$ as the response to the Inv-CDH challenge, thereby solving the Inv-CDH problem.

Analysis of A_2' s advantage in this game:

a) The responses to A_2' s H_2 queries are uniformly distributed, indistinguishable from the real environment.

b) Private key extraction queries and signature oracle queries can proceed smoothly without stopping, i.e., events E_1 and E_2 do not occur.

c) Therefore, if events E_1, E_2, and E_3 do not occur, then C can solve an instance of the Inv-CDH problem. The probability that events E_1, E_2, and E_3 do not occur satisfies:

$$\Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3] \geq \left(1 - \frac{1}{q_H}\right)^{q_s} \frac{1}{q_H}$$

However, when A_2 forges a valid signature without querying H_2, this simulation has a flaw. Considering that the oracle output satisfies uniform distribution, the probability of this event is $\frac{1}{q}$, so A_2' s advantage in this game is $\varepsilon' \geq \frac{1}{q_H}\left(1 - \frac{1}{q_H}\right)^{q_s}\varepsilon$, and the running time is $t' \leq t + (2q_{pk} + q_s)t_H + q_{pk}t_{pk} + q_s t_s$.

2.3 Performance Analysis

As shown in Table 1 , we compare our scheme with recent elliptic curve-based certificateless digital signature schemes in terms of signature process, verification process, and signature length. S_m denotes one scalar multiplication on G_1, P_r denotes one bilinear pairing operation, and H denotes one hash-to-point operation.

The Tso [**?**] scheme uses one scalar multiplication on G_1 and one hash-to-point operation in the signature process, and 2 bilinear pairing operations, one scalar multiplication on G_1, and one hash-to-point operation in the verification process. The Chen [**?**] scheme uses one scalar multiplication on G_1 in the signature process, and two bilinear pairing operations, one scalar multiplication on G_1, and one hash-to-point operation in the verification process. The Gayathri [**?**] scheme uses 3 scalar multiplications on G_1 in the signature process, and 2 bilinear pairing operations and one scalar multiplication on G_1 in the verification process. The Karati [**?**] scheme uses 2 scalar multiplications on G_1 in the signature process, and 1 bilinear pairing operation and 2 scalar multiplications on G_1 in the verification process.

Compared with the above schemes, our scheme uses 1 scalar multiplication on G_1 in the signature process and 1 bilinear pairing operation and 1 scalar multiplication on G_1 in the verification process, demonstrating higher efficiency. In terms of signature length, among the compared certificateless digital signature schemes, except for Gayathri and Karati with signature length $2|G_1|$, the remaining schemes have signature length $|G_1|$. The above comparison shows that our scheme has a shorter signature length, lower computational cost, and higher operational efficiency.

Table 1 Performance comparison of schemes

Scheme	Signature Process	Verification Process	Signature Length
[?]	$1S_m + 1H$	$2P_r + 1S_m + 1H$	$2\|G_1\|$
[?]	$1S_m$	$2P_r + 1S_m + 1H$	$2\|G_1\|$
[?]	$3S_m$	$2P_r + 1S_m$	$2\|G_1\|$
[?]	$2S_m$	$1P_r + 2S_m$	$\|G_1\|$
Our Scheme	$1S_m$	$1P_r + 1S_m$	$\|G_1\|$

2.4 Experiments and Simulation

Under the Windows 7 operating system, using Microsoft Visual Studio 2012 platform and combining bilinear pairing operations on elliptic curves with the PBC library in C language, we implemented our scheme. In the same environment (OS: Windows 7 64-bit, CPU: Intel(R) CoreTM i3-4150 @ 3.50 GHz, RAM: Kingston HX424C15FB2 8 GB), we ran recent elliptic curve-based certificateless digital signature schemes and compared the average time consumption over 100 runs. The experimental results are shown in Table 2 . Our scheme' s average total time consumption is 0.126s, with average signature time of 0.024s and average verification time of 0.032s. The total time consumption of our scheme is approximately 27% less than the Tso scheme, 19% less than the Chen scheme, 25% less than the Gayathri scheme, and 21% less than the Karati scheme.

Table 2 Comparison of the average time-consuming of 100 results

Scheme	Total Time (s)
[?]	0.173
[?]	0.156
[?]	0.168
[?]	0.160
Our Scheme	0.126

3 Conclusion

Based on existing certificateless cryptography, this paper proposes a certificateless short signature scheme with double KGC. Under the random oracle model, we prove that our scheme is existentially unforgeable against adaptive chosen-message attacks. The double KGCs in our scheme mutually restrict each other, achieving a separation of powers that reduces the probability of KGC master key leakage and mitigates system damage caused by KGC master key leakage and malicious manipulation, thereby improving security. Moreover, the scheme has a short signature length—if G_1 is a 160-bit elliptic curve group, the signature length is only 160 bits. In summary, our scheme features high operational

efficiency, low computational cost, short signature length, and strong practicality and security, making it suitable for low-bandwidth environments. The double KGC characteristic also facilitates KGC deployment and cloud-based KGC implementation. In practical applications, for identity authentication, relying solely on signatures is insufficient and needs to be combined with other technologies such as two-factor authentication [**?**, **?**] to enhance security. Therefore, future research will focus on implementing multi-technology fused identity authentication in application development.

References

[1] Shamir A. Identity-based cryptosystems and signature schemes [C]//Proc of Workshop on the Theory & Application of Cryptographic Techniques. Berlin: Springer, 1984: 47-53.

[2] Alriyami S S, Paterson K G. Certificateless public key cryptography [J]. Asiacrypt, 2003, 2894(2): 452-473.

[3] Feng Lei, Peng Changgen, Peng Yanguo. Efficient certificateless multi-signature scheme [J]. Application Research of Computers, 2012, 29(2): 644-645.

[4] Islam S H, Obaidat M S. Design of provably secure and efficient certificateless blind signature scheme using bilinear pairing [J]. Security & Communication Networks, 2016, 8(18): 4319-4332.

[5] Gao Zhuo, Hu Liang, Li Hongtu. A new efficient leakage-free certificateless signature [C]//Proc of International Forum on Mechanical, Control and Automation. Paris: Atlantis Press, 2017.

[6] Gong Zheng, Long Yu, Hong Xuan, et al. Two certificateless aggregate signatures from bilinear maps [J]. Journal of Information Science & Engineering, 2007, 26(6): 2093-2106.

[7] Zhang Yongjie, Zhang Yulei, Wang Caifen. Security analysis and improvement of aggregate signature schemes [J]. Computer Applications and Software, 2017, 34(8): 307-311.

[8] Chen Yuchi, Horng G, Liu Chaoliang. Efficient certificateless aggregate signature scheme [J]. Journal of Electronic Science and Technology, 2012, 10(3): 209-214.

[9] Yu XiuYing, He Dake. New certificateless aggregate signature schemes [J]. Application Research of Computers, 2014, 31(8): 2485-24879.

[10] Zhang Yulei, Zhou Dongrui, Li Chenyi, et al. Certificateless-based efficient aggregate signature scheme with universal designated verifier [J]. Journal on Communications, 2015, 36(2): 48-55.

[11] Ge Lixia, Li Xiao, He Mingxing, et al. Improved certificateless proxy signature scheme [J]. Computer Engineering and Applications, 2017, 53(8): 92-94.

[12] Liu Xiaohong, Zhang Jianzhong. Analysis and improvement of certificateless proxy signature scheme [J]. Computer Engineering and Applications, 2014, 50(22): 115-117.

[13] Boneh D, Lynn B, Shacham H. Short signatures from the weil pairing [C]//Proc of the 7th International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2001: 514-532.

[14] Boneh D, Franklin M. Identity-based encryption from the weil pairing [J]. Siam Journal on Computing, 2003, 32(3): 213-229.

[15] Galbraith S, Harrison K, Soldera D. Implementing the tate pairing [C]//Proc of International Symposium on Algorithmic Number Theory. Berlin: Springer, 2002: 324-337.

[16] Tso R, Huang Xinyi, Susilo W. Strongly secure certificateless short signatures [J]. Journal of Systems & Software, 2012, 85(6): 1409-1417.

[17] Chen Yuchi, Horng G, Liu Chaoliang. Strong non-repudiation based on certificateless short signatures [J]. Iet Information Security, 2013, 7(3).

[18] Gayathri N B, Reddy P V. Efficient certificateless signature scheme with provable security [C]//Proc of IEEE International Conference on Advanced Computing. 2016: 322-337.

[19] Karati A, Islam S H, Karuppiah M. Provably secure and lightweight certificateless signature scheme for IIoT environments [J]. IEEE Transactions on Industrial Informatics, 2018, 14(8): 3701-3711.

[20] He Debiao, Wang Ding. Robust biometrics-based authentication scheme for multiserver environment [J]. IEEE Systems Journal, 2015, 9(3).

[21] Wang Ding, Wang Ping. Two birds with one stone: two-factor authentication with security beyond conventional bound [J]. IEEE Trans on Dependable and Secure Computing, 2016, 15(4): 708-722.

Note: Figure translations are in progress. See original paper for figures.