Yao *et al. J. Surveill. Secur. Saf.* **2025**, 6, 17-34
DOI: 10.20517/jsss.2024.33

Journal of Surveillance, Security and Safety

Research Article

Open Access

Certificateless signature and auditing schemes secure against super type adversaries without random oracle

Suxuan Yao[1], Ge Wu[2], Xueqiao Liu[3], Sihan Hu[2]

[1]School of Cyber Science and Engineering, Southeast University, Wuxi 214000, Jiangsu, China.
[2]School of Cyber Science and Engineering, Southeast University, Nanjing 210096, Jiangsu, China.
[3]School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia.

Correspondence to: A. Prof. Ge Wu, School of Cyber Science and Engineering, Southeast University, 2# Southeast University Road, Nanjing 210096, Jiangsu, China. E-mail: gewu@seu.edu.cn

How to cite this article: Yao, S; Wu, G; Liu, X; Hu, S. Certificateless signature and auditing schemes secure against super type adversaries without random oracle. *J. Surveill. Secur. Saf.* **2025**, *6*, 17-34. http://dx.doi.org/10.20517/jsss.2024.33

Received: 24 Oct 2024 **First Decision:** 7 Dec 2024 **Revised:** 24 Jan 2025 **Accepted:** 25 Feb 2025 **Published:** 15 Apr 2025

Academic Editor: Panayiotis Kotzanikolaou **Copy Editor:** Ting-Ting Hu **Production Editor:** Ting-Ting Hu

Abstract

Cryptographic algorithms are essential for securing data in modern internet applications. As the volume of data increases and security challenges evolve, the significance of these algorithms intensifies. Certificateless public key cryptography addresses the challenges of certificate management inherent in traditional public key cryptography and resolves the key escrow issue associated with identity-based public key cryptography. Notably, previous certificateless signature schemes secure in the random oracle model exhibit vulnerabilities when instantiated in the standard model. There are two types of adversaries in certificateless signature scheme. Type I and Type II adversaries are further categorized into three levels: Normal, Strong, and Super, with Super denoting the most powerful known adversaries. In this work, we present a new certificateless signature scheme designed against Super Type I and Type II adversaries in the standard model based on the computational Diffie-Hellman problem; additionally, the certificateless signature approach can be extended to develop secure cloud auditing schemes, which is for addressing data integrity and security in cloud environments.

Keywords: Certificateless cryptography, digital signature, super type adversary, standard model, cloud auditing

Table 1. Three types of Sign oracles

Normal sign	Public key has not been replaced
Strong sign	If the public key is replaced, additional information must be provided
Super sign	No additional information is required, even if the public key is replaced

1. INTRODUCTION

Cryptographic algorithms are fundamental to modern Internet technology, which ensures data security during transmission, storage, and processing. As data volume surges and security challenges intensify, their importance grows. In 1976, Diffie and Hellman [1] introduced public key cryptography (PKC), effectively addressing the inherent issues of key management and non-repudiation in traditional symmetric systems, thereby expanding the application of cryptography in network security. Public key encryption and key agreement techniques provide effective key management, while digital signature meets non-repudiation needs. In 1984, Shamir [2] proposed identity-based cryptography (IBC), using public identity information as public keys to avoid the complexity of traditional public key infrastructure (PKI), though it raises key escrow problem; i.e., the user's private key is entirely generated by the key generation center (KGC) in IBC, which can impersonate any user without being detected. In 2003, Al-Riyami and Paterson [3] introduced certificateless PKC (CL-PKC), discarding the use of public key certificates, blending the benefits of traditional PKC and IBC, and enhancing usability and security. Subsequently, Huang *et al.* [4] established the first formal security model for certificateless signature and proposed a provably secure scheme under this model. In 2012, Huang *et al.* [5] further classified Type I and Type II adversaries in certificateless signature systems into three levels: Normal, Strong, and Super, with Super representing the strongest known adversaries. In the security model, the adversary's attack capabilities are characterized by three types of Sign oracles (which take a message as input and return the signature), each with different operating conditions, as shown in Table 1.

There are many provably secure certificateless signature schemes in the random oracle model (ROM), such as constructions [6–12]; in particular, the schemes [7,11,12] are secure against Super adversaries. The ROM is widely utilized in the security proofs of cryptographic schemes; however, in 1998, Canetti *et al.* [13] presented a scheme that is secure in the ROM but cannot be securely instantiated in the Standard Model (STM). This implies that proving a scheme secure in the ROM does not guarantee it is free from security flaws in practice, whereas proving security in the STM offers a more reliable assurance of the scheme's security. In 2007, the provably secure certificateless signature scheme in the STM was first introduced by Liu *et al.* [14]. Thereafter, the scheme was improved by Xiong *et al.* [15]. Xia *et al.* [16] further analyzed the scheme by Xiong *et al.*, demonstrating it is vulnerable to public key replacement attacks. Similarly, subsequent schemes [17–19] have been proven insecure against public key replacement attacks. Table 2 below summarizes recent certificateless signature schemes that claim security in the STM, with classifications in the model column—Normal, Strong, and Super—reflecting the adversary types as categorized in the work by Huang *et al* [5]. "NaS" indicates that it is not as specified, meaning that no proof exists or the current scheme cannot resist security analysis as defined in the security model. As shown in Table 2, existing certificateless signature schemes that are secure in the STM can only withstand attacks from Strong adversaries. To overcome these limitations, we propose a new certificateless signature scheme against Super Type I and Super Type II adversaries in the STM, which is reduced to the hardness of computational Diffie–Hellman (CDH) problem. In addition, we extend our technique to enable its application in the certificateless cloud auditing (CLCA) scheme.

1.1. Technical Overview

In 2015, Hung *et al.* [20] shed light on achieving a certificateless signature scheme that is secure against Super adversaries in the STM. However, Yang *et al.* [21] pointed out a flaw in the proof provided by Huang *et al.* regarding the Type II adversary; i.e., the simulated signature provided by the simulator fails to pass verification of validity, preventing it from always correctly responding to adversary's signature queries.

Table 2. Comparison of some certificateless signature schemes in the STM

Scheme	Type I	Type II
Hung *et al.* [20]	Super	NaS [21]
Pang *et al.* [22]	Normal	Normal
Wang *et al.* [23]	Strong	Super
Shim [24]	NaS [25]	Strong
Tseng *et al.* [26]	Strong	Strong
Wu *et al.* [25]	Strong	Strong
Rastegari *et al.* [27]	Strong	Strong
Yang *et al.* [28]	Strong	Strong
Ours	Super	Super

The base idea of our construction is that to achieve security against Super adversaries, the challenger should be able to simulate signatures using only existing secret information and the user's public key during the simulation. This ensures that regardless of the adversary's attempts to replace the user's public key, the challenger can respond to signature queries. Since the secret information is known only to the challenger, the adversary cannot forge signatures merely by knowing the user's public key. We observed that in the security proof of the Waters signature scheme [29], the challenger can compute a valid signature using a series of secret information embedded in the public parameters, the public parameters, and the message. The hard problem embedded in the public parameters and the public parameters computed from the secret information exhibit a certain degree of independence. This insight inspired us to construct a certificateless signature scheme based on the structure of the Waters scheme.

On another note, Hu *et al.* [30] proposed in 2007 that a certificateless signature scheme could be constructed by a signature scheme and an identity-based signature (IBS) scheme. However, this generic construction has certain limitations and cannot be directly considered secure against Super adversaries without modification. Paterson's IBS scheme [31] is an extension of the Waters signature scheme; both are based on bilinear maps. In this scheme, the user's private key corresponds to the Waters signature of the user's identity. Similarly, this can be utilized as the partial private key for each user in the certificateless signature scheme. In the Waters signature scheme, a secret $\alpha \in \mathbb{Z}_q$ and a random $g_2 \in \mathbb{G}$ are chosen to compute the private key g_2^α and $g_1 = g^\alpha$ and g_2 are public parameters. Therefore, in our certificateless signature construction, we select $x_{ID} \xleftarrow{\$} \mathbb{Z}_q$ as the secret value for each user and $g^{x_{ID}}$ as the user's public key, while each user shares $g_3 \in \mathbb{G}$ as a system public parameter. In the certificateless signing process, we combine the signature on the message signed by the partial private key, which is analogous to the signing process in Paterson's IBS scheme, and signature on the message signed by the user's secret value mirroring the signing process in the Waters signature scheme. Since both signing processes occur in group $\mathbb{G}$, we can obtain the certificateless signature on the message by multiplying the two components in group $\mathbb{G}$.

1.2. Certificateless Cloud Auditing

Additionally, the certificateless signature construction technique employed in this paper can also be applied to the development of CLCA schemes. By using a similar approach, we can achieve a CLCA scheme that is against Super adversaries in the STM. As presented in Figure 1, cloud auditing is widely utilized in cloud storage services to address data security concerns. For instance, user data may be deleted or partially lost due to internal changes or cost considerations of cloud service providers. Moreover, the presence of attackers and malicious users exacerbates these risks. After uploading data to cloud servers, users often delete local copies, necessitating the mitigation of risks associated with traditional verification methods. In 2007, Ateniese *et al.* [32] proposed the provable data possession (POP), while Juels and Kaliski [33] independently introduced the proofs of retrievability (POR), both proven secure in the ROM. On the other hand, there are some cloud auditing schemes in the STM proposed. In 2016, Ma *et al.* [34] proposed a cloud auditing scheme based on the strong RSA assumption, and Zhang *et al.* [35] introduced an identity-based cloud auditing scheme that is also proven secure in the STM. However, existing secure CLCA schemes in the STM, such as those by Deng *et al.* [36] and Yang *et al.* [37], are only proved secure against Strong adversaries.

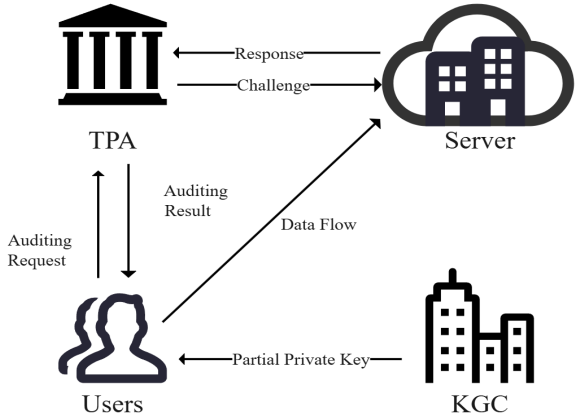

Figure 1. Key entities of a certificateless cloud auditing system

1.3. Organization

The rest of this paper is organized as follows. In Section 2, we first review some mathematical preliminaries including bilinear maps and hardness assumption. Then, we give the definition of certificateless signature and corresponding security model in Section 3. Next, our concrete construction is presented in Section 4, together with the security and efficiency analysis. In addition, we show an extension of our techniques for CLCA in Section 5. Finally, the conclusion part comes in Section 6.

2. PRELIMINARIES

In this section, we describe the definition of mathematical tools and mathematical assumptions.

Definition 1(Bilinear Maps): $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups of a prime order q. Let g be a generator of $\mathbb{G}$. $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map that satisfies the following properties.

1) Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$ for $a, b \in \mathbb{Z}_q$.
2) Nondegeneracy: $e(g, g) \neq 1_{\mathbb{G}_T}$, $1_{\mathbb{G}_T}$ is the identity of $\mathbb{G}_T$.
3) Computability: e is efficiently computable.

Definition 2(CDH Problem): On inputs $(\mathbb{G}, q, g, g^a, g^b)$, where $g, g^a, g^b \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, compute $g^{ab} \in \mathbb{G}$. An adversary $\mathcal{A}$ has advantage at least ϵ in solving the CDH problem on $\mathbb{G}$, if

$$\Pr[\mathcal{A}(\mathbb{G}, q, g, g^a, g^b) = g^{ab}] \geq \epsilon.$$

We say that the (ϵ, t)-CDH assumption holds in group $\mathbb{G}$ if no polynomial-time algorithm can solve the CDH problem with non-negligible probability ϵ in time t.

3. DEFINITION AND SECURITY MODEL

3.1. Definition of Certificateless Signature Schemes

According to[3], a certificateless signature scheme consists of the following seven algorithms: **Setup**, **PartialKeyExt**, **SetSecretValue**, **SetPrivateKey**, **SetPublicKey**, **Sign**, and **Verify**. The specific descriptions of the algorithms are as follows.

- **Setup**$(1^\lambda) \rightarrow (params, msk)$: Given a security parameter 1^λ as input, this algorithm outputs the master

secret key msk and public parameter $params$.

- **PartialKeyExt**$(params, msk, ID) \rightarrow psk_{ID}$: Given the public parameter $params$, master secret key msk, and a user's identity ID as input, this algorithm outputs partial private key psk_{ID}.
- **SetSecretValue**$(params, ID) \rightarrow x_{ID}$: Given the public parameter $params$ and a user's identity ID as input, this algorithm outputs the secret value x_{ID}.
- **SetPrivateKey**$(params, ID, psk_{ID}, x_{ID}) \rightarrow sk_{ID}$: Given the public parameter $params$, partial private key psk_{ID}, secret value x_{ID}, and a user's identity ID as input, this algorithm outputs user's private key sk_{ID}.
- **SetPublicKey**$(params, ID, x_{ID}) \rightarrow pk_{ID}$: Given the public parameter $params$, secret value x_{ID}, and a user's identity ID as input, this algorithm outputs user's public key pk_{ID}.
- **Sign**$(params, m, ID, sk_{ID}) \rightarrow \sigma$: Given the public parameter $params$, a message m, a user's identity ID, and the user's private key sk_{ID}, this algorithm outputs a signature σ.
- **Verify**$(params, m, \sigma, ID, pk_{ID}) \rightarrow 1/0$: Given the public parameter $params$, a message m, a signature σ, a user's identity ID, and the user's public key pk_{ID}, this algorithm outputs 0 or 1.

Correctness Signatures generated by the algorithm **Sign** can pass through the verification in **Verify**. That is, **Verify**$(paaram, m,$ **Sign**$(param, m, ID, sk_{ID}), ID, pk_{ID}) \rightarrow 1$.

3.2. Security Models of Certificateless Signature

There are two types of adversaries in certificateless signature scheme. Type I adversary, denoted by $\mathcal{A}_I$, is equivalent to an attacker outside the system, who can replace the user's public key but does not know the master private key of the KGC. Type II adversary, denoted by $\mathcal{A}_{II}$, is equivalent to the KGC, who knows the master private key but cannot replace the user's public key. In 2012, Huang *et al.*[5] further classified these two types of adversaries into three levels of attack capabilities, from low to high: Normal, Strong, and Super. The Normal adversary cannot obtain signatures on messages under the replaced public key. The Strong adversary can obtain signatures on messages under the replaced public key after providing the challenger with the corresponding secret value. The Super adversary can obtain signatures on messages under the replaced public key without the corresponding secret value.

Due to space limitations, we define Game 1 and Game 2, which simulate the interactions between the challenger and the Super adversary (The security models against Normal and Strong adversaries can be seen as special cases where the adversary is restricted more).

Game 1(against Super $\mathcal{A}_I$)

- *Setup:* The Challenger C_I runs **Setup** with a security parameter 1^λ, and then returns the public parameter $params$, while keeping the master secret key msk.
- *Query:* The adversary $\mathcal{A}_I$ can adaptively perform queries as follows.
  - Create-User: Upon receiving a Create-User query with the user's identity ID_j from adversary $\mathcal{A}_I$, C_I checks the user record table $U\text{-}list$. If the user already exists, it returns $\perp$. Otherwise, it runs **PartialKeyExt**, **SetSecretValue**, and **SetPublicKey** with the relevant parameters. C_I stores the user's identity ID_j, partial private key psk_{ID_j}, user's secret value x_{ID_j}, and user's public key pk_{ID_j} in $U\text{-}list$, and returns pk_{ID_j}.
  - Partial-Private-Key-Extract: Upon receiving a Partial-Private-Key-Extract query with the user's identity ID_j from adversary $\mathcal{A}_I$, C_I checks the user record table $U\text{-}list$. If this user has not been created, it first creates the user and then returns user's partial private key psk_{ID_j}.
  - Secret-Value-Extract: Upon receiving a Secret-Value-Extract query with the user's identity ID_j from adversary $\mathcal{A}_I$, C_I checks the user record table $U\text{-}list$. If this user has not been created, it first creates the user and then returns user's secret value x_{ID_j}. Note that the output of Secret-Value-Extract is not associated with the replaced public key pk'_{ID_j}, i.e., it always output pk_{ID_j}.
  - Public-Key-Replace: Upon receiving a Public-Key-Replace query with the user's identity ID_j from

adversary $\mathcal{A}_I$, C_I checks the user record table $U\text{-}list$. If the user has not been created, it returns $\perp$; otherwise, it updates the user's public key to pk'_{ID_j}, where pk'_{ID_j} is the new public key provided by $\mathcal{A}_I$.

- Super-Sign: Upon receiving a Super-Sign query with the user's identity ID_j and message m from adversary $\mathcal{A}_I$, C_I returns the signature σ of the message m under the public key pk'_{ID_j}, where pk'_{ID_j} is the latest public key for this user in $U\text{-}list$.

- *Forgery:* The adversary $\mathcal{A}_I$ outputs signature σ^* of message m^* for user with identity ID_{j^*} and it wins the game if satisfying the following conditions.
  1) The adversary $\mathcal{A}_I$ has never made the Super-Sign query for the user's identity ID_{j^*} and the message m^*.
  2) The adversary $\mathcal{A}_I$ has never made the Partial-Private-Key-Extract query for the user with identity ID_{j^*}.
  3) Signature σ^* is valid signature of message m^* for user with identity ID_{j^*}.

Game 2(against Super $\mathcal{A}_{II}$)

- *Setup:* The Challenger C_{II} runs **Setup** with a security parameter 1^λ, and then returns the public parameter $params$ and master secret key msk.
- *Query:* The adversary $\mathcal{A}_{II}$ can adaptively perform queries as follows.
  - Create-User: Upon receiving a Create-User query with the user's identity ID_j from adversary $\mathcal{A}_{II}$, C_{II} checks the user record table $U\text{-}list$. If the user already exists, it returns $\perp$. Otherwise, it runs **PartialKeyExt**, **SetSecretValue**, and **SetPublicKey** with the relevant parameters. C_{II} stores the user's identity ID_j, partial private key psk_{ID_j}, user's secret value x_{ID_j}, and user's public key pk_{ID_j} in $U\text{-}list$, and returns pk_{ID_j}.
  - Secret-Value-Extract: Upon receiving a Secret-Value-Extract query with the user's identity ID_j from adversary $\mathcal{A}_{II}$, C_{II} checks the user record table $U\text{-}list$. If this user has not been created, it first creates the user and then returns user's secret value x_{ID_j}. Note that the output of Secret-Value-Extract is not associated with the replaced public key pk'_{ID_j}; i.e., it always outputs pk_{ID_j}.
  - Public-Key-Replace: Upon receiving a Public-Key-Replace query with the user's identity ID_j from adversary $\mathcal{A}_{II}$, C_{II} checks the user record table $U\text{-}list$. If the user has not been created, it returns $\perp$; otherwise, it updates the user's public key to pk'_{ID_j}, where pk'_{ID_j} is the new public key provided by $\mathcal{A}_{II}$.
  - Super-Sign: Upon receiving a Super-Sign query with the user's identity ID_j and message m from adversary $\mathcal{A}_{II}$, C_{II} returns the signature σ of the message m under the public key pk'_{ID_j}, where pk'_{ID_j} is the latest public key for this user in $U\text{-}list$.
- *Forgery:* The adversary $\mathcal{A}_{II}$ outputs signature σ^* of message m^* for user with identity ID_{j^*} and it wins the game if satisfying the following conditions.
  1) The adversary $\mathcal{A}_{II}$ has never made the Super-Sign query for the user's identity ID_{j^*} and the message m^*.
  2) The adversary $\mathcal{A}_{II}$ has never made the Secret-Value-Extract query and Public-Key-Replace for the user with identity ID_{j^*}.
  3) Signature σ^* is valid signature of message m^* for user with identity ID_{j^*}.

4. CONSTRUCTION AND SECURITY PROOF

4.1. Construction

- **Setup**$(1^\lambda) \rightarrow (params, msk)$: Given a security parameter 1^λ as input, select a pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of prime order q and g is a generator of $\mathbb{G}$. Select $\alpha \xleftarrow{\$} \mathbb{Z}_q$, $g_2, g_3 \xleftarrow{\$} \mathbb{G}$ and compute $g_1 = g^\alpha$. Let $\mathsf{H}_u : \{0,1\}^* \rightarrow \{0,1\}^{n_u}$, $\mathsf{H}_m : \{0,1\}^* \rightarrow \{0,1\}^{n_m}$, and $\mathsf{H}'_m : \{0,1\}^* \rightarrow \{0,1\}^{n'_m}$

be three collision-resistant cryptographic hash functions for some $n_u, n_m, n'_m \in \mathbb{Z}$. Select the following elements:

$$
\begin{aligned}
&u', m'_1, m'_2 \leftarrow\$ \mathbb{G} \\
&\tilde{u}_i \leftarrow\$ \mathbb{G}, i = 1, \ldots, n_u \\
&\tilde{m}_{1,i} \leftarrow\$ \mathbb{G}, i = 1, \ldots, n_m \\
&\tilde{m}_{2,i} \leftarrow\$ \mathbb{G}, i = 1, \ldots, n'_m
\end{aligned}
$$

Let $\tilde{U} = \{\tilde{u}_i\}$, $\tilde{M}_1 = \{\tilde{m}_{1,i}\}$, and $\tilde{M}_2 = \{\tilde{m}_{2,i}\}$.The public parameter is $params = \{\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, g_3, u', m'_1, m'_2, \tilde{U}, \tilde{M}_1, \tilde{M}_2, \mathsf{H}_u, \mathsf{H}_m, \mathsf{H}'_m\}$ and the master secret key is $msk = g_2^{\alpha}$.

- **PartialKeyExt**$(params, msk, ID) \rightarrow psk_{ID}$: Given the public parameter $params$, master secret key msk, and a user's identity ID as input, compute $\mathsf{u} = \mathsf{H}_u(ID)$. Let $\mathsf{u}[i]$ denote i-th bit of u. Define $\mathcal{U} \subset \{1, \ldots, n_u\}$ to be set of indices i such that $\mathsf{u}[i] = 1$. Select $h_{ID} \leftarrow\$ \mathbb{Z}_q$. Compute

$$
\begin{aligned}
psk_{ID} &= (psk_{ID,1}, psk_{ID,2}) \\
&= \left(g_2^{\alpha} \left(u' \prod_{i \in \mathcal{U}} \tilde{u}_i \right)^{h_{ID}}, g^{h_{ID}} \right)
\end{aligned}
$$

and output psk_{ID} as the partial private key.

- **SetSecretValue**$(params, ID) \rightarrow x_{ID}$: Given the public parameter $params$ and a user's identity ID as input, select $x_{ID} \leftarrow\$ \mathbb{Z}_q$ and output x_{ID} as the secret value.
- **SetPrivateKey**$(params, ID, psk_{ID}, x_{ID}) \rightarrow sk_{ID}$: Given the public parameter $params$, partial private key psk_{ID}, secret value x_{ID}, and a user's identity ID as input, set $sk_{ID} = (psk_{ID}, x_{ID})$ as the private key.
- **SetPublicKey**$(params, ID, x_{ID}) \rightarrow pk_{ID}$: Given the public parameter $params$, secret value x_{ID}, and a user's identity ID as input, compute $pk_{ID} = g^{x_{ID}}$ and output pk_{ID} as the public key.
- **Sign**$(params, m, ID, sk_{ID}) \rightarrow \sigma$: Given the public parameter $params$, a message m, a user's identity ID, and the user's private key sk_{ID}, compute $\mathsf{m}_1 = \mathsf{H}_m(m)$ and $\mathsf{m}_2 = \mathsf{H}'_m(m)$. Let $\mathsf{m}_1[i]$ and $\mathsf{m}_2[i]$ denote the i-th bit of m_1 and m_2. $\mathcal{M}_1 \subset \{1, \ldots, n_m\}$ and $\mathcal{M}_2 \subset \{1, \ldots, n'_m\}$ are sets of indices i such that $\mathsf{m}_1[i] = 1$ and $\mathsf{m}_2[i] = 1$, respectively. Select $h', h_m, r \leftarrow\$ \mathbb{Z}_q$. Compute

$$
\begin{aligned}
\sigma_1 &= psk_{ID,1} \cdot \left(u' \prod_{i \in \mathcal{U}} \tilde{u}_i \right)^{h'} \cdot \left(m'_1 \prod_{i \in \mathcal{M}_1} \tilde{m}_{1,i} \right)^{h_m} \cdot g_3^{x_{ID}} \cdot \left(m'_2 \prod_{i \in \mathcal{M}_2} \tilde{m}_{2,i} \right)^{r} \\
&= g_2^{\alpha} \left(u' \prod_{i \in \mathcal{U}} \tilde{u}_i \right)^{h_{ID}+h'} \left(m'_1 \prod_{i \in \mathcal{M}_1} \tilde{m}_{1,i} \right)^{h_m} g_3^{x_{ID}} \left(m'_2 \prod_{i \in \mathcal{M}_2} \tilde{m}_{2,i} \right)^{r} \\
\sigma_2 &= psk_{ID,2} \cdot g^{h'} = g^{h_{ID}+h'}, \sigma_3 = g^{h_m}, \sigma_4 = g^{r}
\end{aligned}
$$

and output $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ as the signature.

- **Verify**$(params, m, \sigma, ID, pk_{ID}) \rightarrow 1/0$: Given the public parameter $params$, a message m, a signature σ, a user's identity ID, and the user's public key pk_{ID}, verify whether

$$
e(\sigma_1, g) \stackrel{?}{=} e(g_1, g_2)\, e(g_3, pk_{ID})\, e\left(u' \prod_{i \in \mathcal{U}} \tilde{u}_i, \sigma_2 \right) e\left(m'_1 \prod_{i \in \mathcal{M}_1} \tilde{m}_{1,i}, \sigma_3 \right) e\left(m'_2 \prod_{i \in \mathcal{M}_2} \tilde{m}_{2,i}, \sigma_4 \right)
$$

holds or not. Output 1 if the equality holds; otherwise output 0.

Correctness Analysis

$$
\begin{aligned}
e\left(\sigma_1, g\right) =& e\left(g_2^{\alpha}\left(u^{\prime}\prod_{i\in\mathcal{U}}\tilde{u}_i\right)^{h_{ID}+h^{\prime}}\left(m_1^{\prime}\prod_{i\in\mathcal{M}_1}\tilde{m}_{1,i}\right)^{h_m} g_3^{x_{ID}}\left(m_2^{\prime}\prod_{i\in\mathcal{M}_2}\tilde{m}_{2,i}\right)^{r}, g\right)\\
=& e\left(g_2^{\alpha}, g\right) e\left(\left(u^{\prime}\prod_{i\in\mathcal{U}}\tilde{u}_i\right)^{h_{ID}+h^{\prime}}, g\right) e\left(\left(m_1^{\prime}\prod_{i\in\mathcal{M}_1}\tilde{m}_{1,i}\right)^{h_m}, g\right) e\left(g_3^{x_{ID}}, g\right) e\left(\left(m_2^{\prime}\prod_{i\in\mathcal{M}_2}\tilde{m}_{2,i}\right)^{r}, g\right)\\
=& e\left(g_2, g^{\alpha}\right) e\left(\left(u^{\prime}\prod_{i\in\mathcal{U}}\tilde{u}_i\right), g^{h_{ID}+h^{\prime}}\right) e\left(\left(m_1^{\prime}\prod_{i\in\mathcal{M}_1}\tilde{m}_{1,i}\right), g^{h_m}\right) e\left(g_3, g^{x_{ID}}\right) e\left(\left(m_2^{\prime}\prod_{i\in\mathcal{M}_2}\tilde{m}_{2,i}\right), g^{r}\right)\\
=& e\left(g_1, g_2\right) e\left(g_3, pk_{ID}\right) e\left(u^{\prime}\prod_{i\in\mathcal{U}}\tilde{u}_i, \sigma_2\right) e\left(m_1^{\prime}\prod_{i\in\mathcal{M}_1}\tilde{m}_{1,i}, \sigma_3\right) e\left(m_2^{\prime}\prod_{i\in\mathcal{M}_2}\tilde{m}_{2,i}, \sigma_4\right)
\end{aligned}
$$

4.2. Security Proof

Theorem 1 Assume the (ϵ, t)-CDH assumption holds for $\mathbb{G}$. Then, the proposed construction is $(q_c, q_{psk}, q_{sv}, q_{pkr}, q_\sigma, \epsilon', t')$-secure against the Super $\mathcal{A}_I$, such that $\epsilon \geq \frac{\epsilon'}{16(q_{psk}+q_\sigma)q_\sigma(n_u+1)(n_m+1)}$ and $t \approx t' + O((q_{psk}n_u + q_\sigma(n_u+n_m+n_m'))t_m + (q_c+q_{psk}+q_\sigma)t_e)$, where t_m and t_e are the time for a multiplication and an exponentiation in $\mathbb{G}$, and $q_c, q_{psk}, q_{sv}, q_{pkr}, q_\sigma$ are the numbers of the queries to Create-User,Partial-Private-Key-Extract, Secret-Value-Extract, Public-Key-Replace, Super-Sign, respectively.

Proof We construct a simulator $\mathcal{B}_I$ that simulates the challenger interacting with the Super $\mathcal{A}_I$. $\mathcal{B}_I$ receives a CDH problem instance $< \mathbb{G}, q, g, g^a, g^b >$. Its goal is to compute $g^{ab} \in \mathbb{G}$. The detailed description is as follows.

- *Setup:* Let $l_u = 2(q_{psk} + q_\sigma)$ and $l_m = 2q_\sigma$. Assume that $l_u(n_u + 1) < q$ and $l_m(n_m + 1) < q$. Select two integers $\gamma_u \leftarrow\!\!\$\ [0, n_u]$ and $\gamma_m \leftarrow\!\!\$\ [0, n_m]$, also select the following integers:

$$
\begin{aligned}
&x^{\prime} \leftarrow\!\!\$\ \mathbb{Z}_{l_u}; \quad y^{\prime} \leftarrow\!\!\$\ \mathbb{Z}_{l_m}\\
&\xi^{\prime}, \delta^{\prime} \leftarrow\!\!\$\ \mathbb{Z}_q; \quad c^{\prime}, c_1, \ldots, c_{n_m^{\prime}}, d \leftarrow\!\!\$\ \mathbb{Z}_q\\
&\bar{X} = (\bar{x}_i)_{i=1,2,\ldots,n_u},\ \bar{x}_i \leftarrow\!\!\$\ \mathbb{Z}_{l_u}; \quad \bar{\Xi} = \left(\bar{\xi}_i\right)_{i=1,2,\ldots,n_u},\ \bar{\xi}_i \leftarrow\!\!\$\ \mathbb{Z}_{l_u}\\
&\bar{Y} = (\bar{y}_i)_{i=1,2,\ldots,n_m},\ \bar{y}_i \leftarrow\!\!\$\ \mathbb{Z}_{l_m}; \quad \bar{\Delta} = \left(\bar{\delta}_i\right)_{i=1,2,\ldots,n_m},\ \bar{\delta}_i \leftarrow\!\!\$\ \mathbb{Z}_{l_m}
\end{aligned}
$$

  Define the following functions for binary string u, m_1 and m_2, where $\mathsf{u} = \mathsf{H}_u(ID)$ for user's identity ID and $\mathsf{m}_1 = \mathsf{H}_m(m)$ and $\mathsf{m}_2 = \mathsf{H}_m^{\prime}(m)$ for a message m:

$$
\begin{aligned}
&F(\mathsf{u}) = x^{\prime} + \sum_{i\in\mathcal{U}}\bar{x}_i - l_u\gamma_u; \quad J(\mathsf{u}) = \xi^{\prime} + \sum_{i\in\mathcal{U}}\bar{\xi}_i\\
&G(\mathsf{m}_1) = y^{\prime} + \sum_{i\in\mathcal{M}}\bar{y}_i - l_m\gamma_m; \quad P(\mathsf{m}_1) = \delta^{\prime} + \sum_{i\in\mathcal{M}}\bar{\delta}_i\\
&C(\mathsf{m}_2) = c^{\prime} + \sum_{i\in\mathcal{M}^{\prime}} c_i
\end{aligned}
$$

  Then, we have:

$$
\begin{aligned}
&g_1 = g^a; \quad g_2 = g^b; \quad g_3 = g^d\\
&u^{\prime} = g_2^{x^{\prime}-l_u\gamma_u}g^{\xi^{\prime}}; \quad \tilde{u}_i = g_2^{\bar{x}_i}g^{\bar{\xi}_i}, i \in [1, n_u]; \quad \tilde{U} = \{\tilde{u}_i\}\\
&m_1^{\prime} = g_2^{y^{\prime}-l_m\gamma_m}g^{\delta^{\prime}}; \quad \tilde{m}_{1,i} = g_2^{\bar{y}_i}g^{\bar{\delta}_i}, i \in [1, n_m]; \quad \tilde{M}_1 = \{\tilde{m}_{1,i}\}\\
&m_2^{\prime} = g^{c^{\prime}},\ \tilde{m}_{2,i} = g^{c_i}, i \in [1, n_m^{\prime}]; \quad \tilde{M}_2 = \{\tilde{m}_{2,i}\}
\end{aligned}
$$

$\mathcal{B}_I$ returns the public parameters $PP = \{\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, g_3, u', m'_1, m'_2, \tilde{U}, \tilde{M}_1, \tilde{M}_2, \mathsf{H}_u, \mathsf{H}_m, \mathsf{H}'_m\}$ to $\mathcal{A}_I$, while the master key is $msk = g^{ab}$. And the following equations hold:

$$
\begin{aligned}
u' \prod_{i \in \mathcal{U}} \tilde{u}_i &= g_2^{F(\mathsf{u})} g^{J(\mathsf{u})} \\
m'_1 \prod_{i \in \mathcal{M}_1} \tilde{m}_{1,i} &= g_2^{G(\mathsf{m}_1)} g^{P(\mathsf{m}_1)} \\
m'_2 \prod_{i \in \mathcal{M}_2} \tilde{m}_{2,i} &= g^{C(\mathsf{m}_2)}
\end{aligned}
$$

- *Query:* The adversary $\mathcal{A}_I$ can adaptively perform queries as follows.
  - Create-User: Upon receiving a Create-User query with the user's identity ID_j from adversary $\mathcal{A}_I$, $\mathcal{B}_I$ checks the user record table $U\text{-}list$. If the user already exists, it returns $\perp$. Otherwise, it selects user's secret value $x_{ID_j} \leftarrow\!\!\$\ \mathbb{Z}_q$ and computes user's public key $pk_{ID_j} = g^{x_{ID_j}}$. Then $\mathcal{B}_I$ inserts $(ID_j, \_, x_{ID_j}, pk_{ID_j})$ to $U\text{-}list$ and returns user's public key pk_{ID_j}. Note that $U\text{-}list$ is initially empty and stores the corresponding information as $(ID_j, psk_{ID_j}, x_{ID_j}, pk_{ID_j})$.
  - Partial-Private-Key-Extract: Upon receiving a Partial-Private-Key-Extract query with the user's identity ID_j from adversary $\mathcal{A}_I$, $\mathcal{B}_I$ checks the user record table $U\text{-}list$. If this user has not been created, it first creates the user. Then if there is no information of the partial private key, $\mathcal{B}_I$ returns the partial private key psk_{ID_j} to $\mathcal{A}_I$. Otherwise, it computes $\mathsf{u}_j = \mathsf{H}_u(ID_j)$ and works as follows.
    1) $F(\mathsf{u}_j) \neq 0 \mod q$: Select $h'_{ID_j} \leftarrow\!\!\$\ \mathbb{Z}_q$ and compute

$$
\begin{aligned}
psk_{ID_j} &= (psk_{ID_j,1}, psk_{ID_j,2}) \\
&= \left(g^{ab} \left(g_2^{F(\mathsf{u}_j)} g^{J(\mathsf{u}_j)} \right)^{h'_{ID_j} - \frac{a}{F(\mathsf{u}_j)}}, g^{h'_{ID_j} - \frac{a}{F(\mathsf{u}_j)}} \right) \\
&= \left(g_1^{-\frac{J(\mathsf{u}_j)}{F(\mathsf{u}_j)}} \left(g_2^{F(\mathsf{u}_j)} g^{J(\mathsf{u}_j)} \right)^{h'_{ID_j}}, g_1^{-\frac{1}{F(\mathsf{u}_j)}} g^{h'_{ID_j}} \right)
\end{aligned}
$$

    $\mathcal{B}_I$ stores the partial private key psk_{ID_j} to this user's entry in $U\text{-}list$ and returns it to $\mathcal{A}_I$.
    2) $F(\mathsf{u}_j) = 0 \mod q$: $\mathcal{B}_I$ returns $\perp$.
  - Secret-Value-Extract: Upon receiving a Secret-Value-Extract query with the user's identity ID_j from adversary $\mathcal{A}_I$, $\mathcal{B}_I$ checks the user record table $U\text{-}list$. If this user has not been created, it first creates the user and then returns user's secret value x_{ID_j}.
  - Public-Key-Replace: Upon receiving a Public-Key-Replace query with the user's identity ID_j from adversary $\mathcal{A}_I$, $\mathcal{B}_I$ checks the user record table $U\text{-}list$. If the user has not been created, it returns $\perp$; otherwise, it updates the user's public key to pk'_{ID_j}, where pk'_{ID_j} is the new public key provided by $\mathcal{A}_I$.
  - Super-Sign: Upon receiving a Super-Sign query with the user's identity ID_j and message m from adversary $\mathcal{A}_I$, $\mathcal{B}_I$ checks the user record table $U\text{-}list$. If the user has not been created or lacks information of the private key, $\mathcal{B}_I$ executes Create-User and Partial-Private-Key-Extract accordingly. Then it computes $\mathsf{m}_1 = \mathsf{H}_m(m)$ and $\mathsf{m}_2 = \mathsf{H}'_m(m)$ and works as follows.
    1) $F(\mathsf{u}_j) \neq 0 \mod q$: Select $h', h_m, r \leftarrow\!\!\$\ \mathbb{Z}_q$ and compute

$$\begin{aligned}
\sigma_1 &= g^{ab}\left(g_2^{F(\mathsf{u}_j)}g^{J(\mathsf{u}_j)}\right)^{h'_{ID_j}-\frac{a}{F(\mathsf{u}_j)}}\left(g_2^{F(\mathsf{u}_j)}g^{J(\mathsf{u}_j)}\right)^{h'}\left(g_2^{G(\mathsf{m}_1)}g^{P(\mathsf{m}_1)}\right)^{h_m} g^{dx'_{ID_j}}\left(g^{C(\mathsf{m}_2)}\right)^{r} \\
&= g_1^{-\frac{J(\mathsf{u}_j)}{F(\mathsf{u}_j)}}\left(g_2^{F(\mathsf{u}_j)}g^{J(\mathsf{u}_j)}\right)^{h'_{ID_j}+h'}\left(g_2^{G(\mathsf{m}_1)}g^{P(\mathsf{m}_1)}\right)^{h_m}(pk'_j)^{d}\left(g^{C(\mathsf{m}_2)}\right)^{r} \\
&= psk_{j,1}\left(g_2^{F(\mathsf{u}_j)}g^{J(\mathsf{u}_j)}\right)^{h'}\left(g_2^{G(\mathsf{m}_1)}g^{P(\mathsf{m}_1)}\right)^{h_m}(pk'_j)^{d}\left(g^{C(\mathsf{m}_2)}\right)^{r} \\
\sigma_2 &= g^{h'_{ID_j}+h'-\frac{a}{F(\mathsf{u}_j)}} \\
&= psk_{j,2}g^{h'} \\
\sigma_3 &= g^{h_m} \\
\sigma_4 &= g^{r}
\end{aligned}$$

2) $F(\mathsf{u}_j) = 0 \mod q$: If $G(\mathsf{m}_1) \neq 0 \mod q$, $\mathcal{B}_I$ selects $H', h'_m, r \leftarrow\!\!\$\ \mathbb{Z}_q$. Let $h_{id_j} + h' = H'$ and compute

$$\begin{aligned}
\sigma_1 &= g^{ab}\left(g^{J(\mathsf{u}_j)}\right)^{h_{id_j}}\left(g^{J(\mathsf{u}_j)}\right)^{h'}\left(g_2^{G(\mathsf{m}_1)}g^{P(\mathsf{m}_1)}\right)^{h'_m-\frac{a}{G(\mathsf{m}_1)}} g^{dx'_{id_j}}\left(g^{C(\mathsf{m}_2)}\right)^{r} \\
&= g_1^{-\frac{P(\mathsf{m}_1)}{G(\mathsf{m}_1)}}\left(g^{J(\mathsf{u}_j)}\right)^{H'}\left(g_2^{G(\mathsf{m}_1)}g^{P(\mathsf{m}_1)}\right)^{h'_m}(pk'_j)^{d}\left(g^{C(\mathsf{m}_2)}\right)^{r} \\
\sigma_2 &= g^{H'} \\
\sigma_3 &= g^{h'_m-\frac{a}{G(\mathsf{m}_1)}} \\
&= g_1^{-\frac{1}{G(\mathsf{m}_1)}}g^{h'_m} \\
\sigma_4 &= g^{r}
\end{aligned}$$

Otherwise, $\mathcal{B}_I$ returns $\perp$.

If $\mathcal{B}_I$ does not abort, it returns $\sigma_{ID_j,m} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ to $\mathcal{A}_{\mathrm{I}}$. Note that $\mathcal{B}_I$ needs no additional information other than the user's current public key pk'_{ID_j} to generate the signature with Super-Sign.

- *Forgery:* The adversary $\mathcal{A}_I$ outputs signature σ^* of message m^* for user with identity ID^*_j. $\mathcal{B}_I$ compute $\mathsf{u}^* = \mathsf{H}_u(ID^*_j)$ and $\mathsf{m}^*_1 = \mathsf{H}_m(m^*)$ and checks following conditions.
  1) $F(\mathsf{u}^*) = 0 \mod q$.
  2) $G(\mathsf{m}^*_1) = 0 \mod q$.
  3) Signature σ^* is valid signature of message m^* for user with identity ID^*_j.

  If any of the above conditions are not met, $\mathcal{B}_I$ returns $\perp$. Otherwise, it computes g^{ab} as follows:

$$\begin{aligned}
\frac{\sigma^*_1}{\left(\sigma^*_2\right)^{J(\mathsf{u}^*)}\left(\sigma^*_3\right)^{P(\mathsf{m}^*_1)}\left(pk^{*\prime}_j\right)^{d}\left(\sigma^*_4\right)^{C(\mathsf{m}^*_2)}} &= \frac{g^{ab}\left(g^{J(\mathsf{u}^*)}\right)^{h_{ID^*_j}+h'}\left(g^{P(\mathsf{m}^*_1)}\right)^{h'_{m^*}} g^{dx'_{ID^*_j}}\left(g^{C(\mathsf{m}^*_2)}\right)^{r}}{\left(g^{h_{ID^*_j}+h'}\right)^{J(\mathsf{u}^*)}\left(g^{h'_{m*}}\right)^{P(\mathsf{m}^*_1)}\left(g^{x'_{ID^*_j}}\right)^{d}(g^{r})^{C(\mathsf{m}^*_2)}} \\
&= g^{ab}
\end{aligned}$$

Then $\mathcal{B}_I$ outputs g^{ab} as a solution of the CDH problem.

Probability analysis To make analysis simple, we need following conclusions. Form $l_u(n_u+1) < q, \gamma_u \in [0, n_u]$ and $x', \bar{x}_1, \ldots, \bar{x}_{n_u} \in \mathbb{Z}_{l_u}$, these conditions imply $F(\mathsf{u}) = x' + \sum_{i\in\mathcal{U}} \bar{x}_i - l_u\gamma_u \in (-q, q)$, where $\mathsf{u} = \mathsf{H}_u(ID)$. Then we have the proposition that if $F(\mathsf{u}) = 0 \mod q$ then $F(\mathsf{u}) = 0 \mod l_u$ and its contrapositive that $F(\mathsf{u}) \neq 0 \mod l_u$ then $F(\mathsf{u}) \neq 0 \mod q$. Similarly, the corresponding conclusion holds for $G(\mathsf{m}_1)$, where $\mathsf{m}_1 = \mathsf{H}_m(m)$.

Let $\mathsf{u}_1, \ldots, \mathsf{u}_{q_{ID}}$ be the output of the hash function H_u appearing in either Partial-Private-Key-Extract queries

or in Super-Sign queries not involving ID_j^*, and let $\mathsf{m}_{1,1}, ..., \mathsf{m}_{1,q_M}$ be the output of the hash function H_m in Super-Sign queries not involving m^*. We have $q_{ID} \leq q_{psk} + q_\sigma$ and $q_M \leq q_\sigma$. Then we define the following events A_i, A^*, B_j, B^* and E.

$$
\begin{aligned}
&A_i : F(\mathsf{u}_i) \neq 0 \mod l_u, i = 1, \ldots, q_{ID} \\
&A^* : F(\mathsf{u}^*) = 0 \mod q \\
&B_j : G(\mathsf{m}_{1,j}) \neq 0 \mod l_m, j = 1, \ldots, q_M \\
&B^* : F(\mathsf{m}_1^*) = 0 \mod q \\
&E : Signature\ \sigma^*\ is\ valid\ signature\ of\ message\ m^*\ for\ user\ with\ identity\ ID_j^*
\end{aligned}
$$

According to the simulation, the probability of $\mathcal{B}_I$ not aborting is

$$
\Pr\left[\overline{\text{abort}}\right] \geq \Pr\left[\left(\bigwedge_{i=1}^{q_{ID}} A_i \wedge A^*\right) \wedge \left(\bigwedge_{j=1}^{q_M} B_i j \wedge B^*\right) \wedge E\right]
$$

In the simulation, since all variates are chosen randomly, with above conclusions, we have

$$
\begin{aligned}
\Pr[A^*] &= \Pr\left[F(\mathsf{u}^*) = 0 \mod q\right] \\
&= \Pr\left[F(\mathsf{u}^*) = 0 \mod q \wedge F(\mathsf{u}^*) = 0 \mod l_u\right] \\
&= \Pr\left[F(\mathsf{u}^*) = 0 \mod q \mid F(\mathsf{u}^*) = 0 \mod l_u\right] \Pr\left[F(\mathsf{u}^*) = 0 \mod l_u\right] \\
&= \frac{1}{n_u + 1} \frac{1}{l_u}
\end{aligned}
$$

Also, we have

$$
\begin{aligned}
\Pr[\bigwedge_{i=1}^{q_{ID}} A_i \mid A^*] &= 1 - \Pr\left[\bigvee_{i=1}^{q_{ID}} \overline{A_i} \mid A^*\right] \\
&\geq 1 - \Sigma_{i=1}^{q_{ID}} \Pr\left[\overline{A_i} \mid A^*\right]
\end{aligned}
$$

We can get the probability $\Pr\left[\overline{A_i} \mid A^*\right] = \frac{1}{l_u}$, since the events $F(\mathsf{u}_{i_1}) = 0 \mod l_u$ and $F(\mathsf{u}_{i_2}) = 0 \mod l_u$ are independent, where $i_1 \neq i_2$, and the events A_i and A^* are independent for any i. Hence, we compute

$$
\begin{aligned}
\Pr\left[\bigwedge_{i=1}^{q_{ID}} A_i \wedge A^*\right] &= \Pr\left[\bigwedge_{i=1}^{q_{ID}} A_i \mid A^*\right] \Pr[A^*] \\
&\geq \left(1 - \frac{q_{ID}}{l_u}\right) \frac{1}{n_u + 1} \frac{1}{l_u} \\
&\geq \left(1 - \frac{q_{psk} + q_\sigma}{l_u}\right) \frac{1}{n_u + 1} \frac{1}{l_u} \\
&= \frac{1}{4(q_{psk} + q_\sigma)(n_u + 1)}
\end{aligned}
$$

Using a similar analysis technique, we can have $\Pr\left[\wedge_{j=1}^{q_M} B_j \wedge B^*\right] = \frac{1}{4q_\sigma(n_m+1)}$. Building on the above results, we can get the probability of $\mathcal{B}_I$ not aborting

$$\Pr\left[\left(\bigwedge_{i=1}^{q_{ID}} A_i \wedge A^*\right) \wedge \left(\bigwedge_{j=1}^{q_M} B_i j \wedge B^*\right)\right] \geq \frac{1}{16(q_{psk}+q_\sigma)q_\sigma(n_u+1)(n_m+1)}$$

If $\mathcal{A}_I$ will forge a valid signature with the probability ϵ' and time t', simulator $\mathcal{B}_I$ can solve CDH problem with the probability $\epsilon \geq \frac{\epsilon'}{16(q_{psk}+q_\sigma)q_\sigma(n_u+1)(n_m+1)}$.The time complexity of simulation is primarily determined by the exponentiations and multiplications in the queries. A Create-User query involves one exponentiation, a Partial-Private-Key-Extract query involves $O(n_u)$ multiplications and $O(1)$ exponentiations, and a Super-Sign query involves $O(n_u + n_m + n'_m)$ multiplications and $O(1)$ exponentiations. Thus the time complexity of solving CDH problem is $t \approx t' + O((q_{psk}n_u + q_\sigma(n_u + n_m + n'_m))t_m + (q_c + q_{psk} + q_\sigma)t_e)$. □

Theorem 2 Assume the (ϵ, t)-CDH assumption holds for $\mathbb{G}$. Then, the proposed construction is $(q_c, q_{sv}, q_{pkr}, q_\sigma, \epsilon', t')$-secure against the Super $\mathcal{A}_{II}$, such that $\epsilon \geq \frac{\epsilon'}{4q_\sigma(n'_m+1)}$ and $t \approx t' + O(q_\sigma(n_u+n_m+n'_m)t_m + (q_c+q_\sigma)t_e)$, where t_m and t_e are the time for a multiplication and an exponentiation in $\mathbb{G}$, and $q_c, q_{sv}, q_{pkr}, q_\sigma$ are the numbers of queries to Create-User, Secret-Value-Extract, Public-Key-Replace, Super-Sign, respectively.

Proof We construct a simulator $\mathcal{B}_{II}$ that simulates the challenger interacting with the Super $\mathcal{A}_{II}$. $\mathcal{B}_{II}$ receives a CDH problem instance $< \mathbb{G}, q, g, g^a, g^b >$. Its goal is to compute $g^{ab} \in \mathbb{G}$. The detailed description is as follows.

- *Setup:* Let $l'_m = 2q_\sigma$. Assume that $l'_m(n'_m + 1) < q$. Select two integers $\gamma'_m \leftarrow\!\!\$\ [0, n'_m]$, also select the following integers:

$$\begin{aligned}
&z' \leftarrow\!\!\$\ \mathbb{Z}_{l'_m}; \quad \theta' \leftarrow\!\!\$\ \mathbb{Z}_q \\
&c'_1, c_{1,1}, \ldots, c_{1,n_u} \leftarrow\!\!\$\ \mathbb{Z}_q \\
&c'_2, c_{2,1}, \ldots, c_{2,n_m} \leftarrow\!\!\$\ \mathbb{Z}_q \\
&d_1, d_2 \leftarrow\!\!\$\ \mathbb{Z}_q \\
&\bar{Z} = (\bar{z}_i)_{i=1,2,\ldots,n'_m},\ \bar{z}_i \leftarrow\!\!\$\ \mathbb{Z}_{l'_m}; \quad \bar{\Theta} = (\bar{\theta}_i)_{i=1,2,\ldots,n'_m},\ \bar{\theta}_i \leftarrow\!\!\$\ \mathbb{Z}_{l'_m}
\end{aligned}$$

  Define the following functions for binary string u, m_1 and m_2, where $\mathsf{u} = \mathsf{H}_u(ID)$ for user's identity ID and $\mathsf{m}_1 = \mathsf{H}_m(m)$ and $\mathsf{m}_2 = \mathsf{H}'_m(m)$ for a message m:

$$\begin{aligned}
&R(\mathsf{m}_2) = z' + \sum_{i\in\mathcal{M}'} \bar{z}_i - l'_m\gamma'_m; \quad Q(\mathsf{m}_2) = \theta' + \sum_{i\in\mathcal{M}'} \bar{\theta}_i \\
&C_1(\mathsf{u}) = c'_1 + \sum_{i\in\mathcal{U}} c_{1,i} \\
&C_2(\mathsf{m}_1) = c'_2 + \sum_{i\in\mathcal{M}} c_{2,i}
\end{aligned}$$

  Then, we have:

$$\begin{aligned}
&g_1 = g^{d_1}; \quad g_2 = g^{d_2}; \quad g_3 = g^b \\
&m'_2 = g_3^{z'-l'_m\gamma'_m} g^{\theta'}; \quad \tilde{m}_{2,i} = g_3^{\bar{z}_i} g^{\bar{\theta}_i},\ i \in [1, n'_m]; \quad \tilde{M}_2 = \{\tilde{m}_{2,i}\} \\
&u' = g^{c'_1}; \quad \tilde{u}_i = g^{c_{1,i}}, i \in [1, n_u]; \quad \tilde{U} = \{\tilde{u}_i\} \\
&m'_1 = g^{c'_2}, \quad \tilde{m}_{1,i} = g^{c_{2,i}}, i \in [1, n_m]; \quad \tilde{M}_1 = \{\tilde{m}_{1,i}\}
\end{aligned}$$

$\mathcal{B}_{II}$ returns the public parameters $PP = \{\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, g_3, u', m'_1, m'_2, \tilde{U}, \tilde{M}_1, \tilde{M}_2, \mathsf{H}_u, \mathsf{H}_m, \mathsf{H}'_m\}$ to $\mathcal{A}_{II}$, while the master key is $msk = g^{d_1 d_2}$. And the following equations hold:

$$m'_2 \prod_{i \in \mathcal{M}_2} \tilde{m}_{2,i} = g_3^{R(\mathsf{m}_2)} g^{Q(\mathsf{m}_2)}$$
$$u' \prod_{i \in \mathcal{U}} \tilde{u}_i = g^{C_1(\mathsf{u})}$$
$$m'_1 \prod_{i \in \mathcal{M}_1} \tilde{m}_{1,i} = g^{C_2(\mathsf{m}_1)}$$

- *Query:* The adversary $\mathcal{A}_{II}$ can adaptively perform queries as follows.
  - Create-User: Upon receiving a Create-User query with the user's identity ID_j from adversary $\mathcal{A}_{II}$, $\mathcal{B}_{II}$ checks the user record table $U\text{-}list$. If the user already exists, it returns $\perp$. Otherwise, it selects user's secret value $x_{ID_j} \leftarrow\!\!\$\ \mathbb{Z}_q$ and computes user's public key $pk_{ID_j} = g^{x_{ID_j}}$. Then $\mathcal{B}_{II}$ inserts $(ID_j, \_, x_{ID_j}, pk_{ID_j})$ to $U\text{-}list$ and returns user's public key pk_{ID_j}. Among all Create-User queries, $\mathcal{B}_{II}$ randomly picks one and let its entry be $(ID_{j'}, \_, \_, g^a)$. Note that $U\text{-}list$ is initially empty and stores the corresponding information as $(ID_j, psk_{ID_j}, x_{ID_j}, pk_{ID_j})$.
  - Secret-Value-Extract: Upon receiving a Secret-Value-Extract query with the user's identity ID_j from adversary $\mathcal{A}_{II}$, $\mathcal{B}_{II}$ checks the user record table $U\text{-}list$. If this user has not been created, it first creates the user and works as follows.
    1) $j \neq j'$: $\mathcal{B}_{II}$ returns x_{id_j}.
    2) $j = j'$: $\mathcal{B}_{II}$ returns $\perp$.
  - Public-Key-Replace: Upon receiving a Public-Key-Replace query with the user's identity ID_j from adversary $\mathcal{A}_{II}$, $\mathcal{B}_{II}$ checks the user record table $U\text{-}list$. If the user has not been created, it returns $\perp$; otherwise, it works as follows.
    1) $j \neq j'$: $\mathcal{B}_{II}$ updates the user's public key to pk'_{ID_j}, where pk'_{ID_j} is the new public key provided by $\mathcal{A}_{II}$.
    2) $j = j'$: $\mathcal{B}_{II}$ returns $\perp$.
  - Super-Sign: Upon receiving a Super-Sign query with the user's identity ID_j and message m from adversary $\mathcal{A}_{II}$, $\mathcal{B}_{II}$ checks the user record table $U\text{-}list$. If the user has not been created, it first creates the user. Then it computes $\mathsf{u} = \mathsf{H}_u(ID)$, $\mathsf{m}_1 = \mathsf{H}_m(m)$, $\mathsf{m}_2 = \mathsf{H}'_m(m)$ and works as follows.
    1) $R(\mathsf{m}_2) \neq 0 \mod q$: Select $h', h_m, r' \leftarrow\!\!\$\ \mathbb{Z}_q$ and compute

$$\begin{aligned}
\sigma_1 &= g^{d_1 d_2} \left(g^{C_1(\mathsf{u}_j)}\right)^{h_{ID_j}} \left(g^{C_1(\mathsf{u}_j)}\right)^{h'} \left(g^{C_2(\mathsf{m}_1)}\right)^{h_m} g^{ab} \left(g_3^{R(\mathsf{m}_2)} g^{Q(\mathsf{m}_2)}\right)^{r' - \frac{x_{ID_j}}{R(\mathsf{m}_2)}} \\
&= psk_{j,1} \left(g^{C_1(\mathsf{u}_j)}\right)^{h'} \left(g^{C_2(\mathsf{m}_1)}\right)^{h_m} \left(pk_{ID_j}\right)^{-\frac{Q(\mathsf{m}_2)}{R(\mathsf{m}_2)}} \left(g_3^{R(\mathsf{m}_2)} g^{Q(\mathsf{m}_2)}\right)^{r'} \\
\sigma_2 &= g^{h'_{ID_j} + h'} \\
&= psk_{j,2} g^{h'} \\
\sigma_3 &= g^{h_m} \\
\sigma_4 &= g^{r' - \frac{x_{ID_j}}{R(\mathsf{m}_2)}} \\
&= \left(pk_{ID_j}\right)^{-\frac{1}{R(\mathsf{m}_2)}} g^{r'}
\end{aligned}$$

    2) $R(\mathsf{m}_2) = 0 \mod q$: $\mathcal{B}_{II}$ returns $\perp$.

    If $\mathcal{B}_{II}$ does not abort, it returns $\sigma_{ID_j,m} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ to $\mathcal{A}_{\mathrm{II}}$. Note that $\mathcal{B}_{II}$ needs no additional information other than the user's current public key pk'_{ID_j} to generate the signature with Super-Sign.
- *Forgery:* The adversary $\mathcal{A}_{II}$ outputs signature σ^* of message m^* for user with identity ID_j^*. $\mathcal{B}_{II}$ computes $\mathsf{u}^* = \mathsf{H}_u(ID_j^*)$, $\mathsf{m}_1^* = \mathsf{H}_m(m^*)$, $\mathsf{m}_2^* = \mathsf{H}'_m(m^*)$ and checks following conditions.

1) $j = j'$
2) $R(\mathsf{m}_2^*) = 0 \mod q$.
3) Signature σ^* is valid signature of message m^* for user with identity ID_j^*.

If any of the above conditions are not met, $\mathcal{B}_{II}$ returns $\perp$. Otherwise, it computes g^{ab} as follows:

$$\frac{\sigma_1^*}{g^{d_1 d_2}\left(\sigma_2^*\right)^{C_1(\mathsf{u}_{j^*})}\left(\sigma_3^*\right)^{C_2(\mathsf{m}_1^*)}\left(\sigma_4^*\right)^{Q(\mathsf{m}_2^*)}} = \frac{g^{d_1 d_2}\left(g^{C_1(\mathsf{u}_{j^*})}\right)^{h_{ID_j^*}+h'}\left(g^{C_2(\mathsf{m}_1^*)}\right)^{h'_{m^*}} g^{ab}\left(g^{Q(\mathsf{m}_2^*)}\right)^{r}}{g^{d_1 d_2}\left(g^{h_{ID_j^*}+h'}\right)^{C_1(\mathsf{u}_{j^*})}\left(g^{h'_{m*}}\right)^{C_2(\mathsf{m}_1^*)}(g^r)^{Q(\mathsf{m}_2^*)}}$$
$$= g^{ab}$$

Then $\mathcal{B}_{II}$ outputs g^{ab} as a solution of the CDH problem.

Probability analysis Its probability analysis is similar to analysis for Theorem 1. Let $\mathsf{m}_{2,1}, ..., \mathsf{m}_{2,q_{M'}}$ be the output of the hash function H'_m in Super-Sign queries not involving m^*. We have $q_{M'} \leq q_\sigma$. Then we define the following events C_k, C^*, D and E.

$$C_k : R(\mathsf{m}_{2,j}) \neq 0 \mod l'_m, k = 1, \ldots, q_{M'}$$
$$C^* : F(\mathsf{m}_2^*) = 0 \mod q$$
$$D : j = j'$$
$$E : Signature\ \sigma^*\ is\ valid\ signature\ of\ message\ m^*\ for\ user\ with\ identity\ ID_j^*$$

According to the simulation, the probability of $\mathcal{B}_{II}$ not aborting is

$$\Pr\left[\overline{\text{abort}}\right] \geq Pr\left[\left(\bigwedge_{k=1}^{q_{M'}} C_k \wedge C^*\right) \wedge D \wedge E\right]$$
$$\geq \frac{\epsilon'}{4q_\sigma q_c(n'_m+1)}$$

If $\mathcal{A}_{II}$ will forge a valid signature with the probability ϵ' and time t', simulator $\mathcal{B}_{II}$ can solve CDH problem with the probability $\epsilon \geq \frac{\epsilon'}{4q_\sigma q_c(n'_m+1)}$.The time complexity of simulation is primarily determined by the exponentiations and multiplications in the queries. A Create-User query involves one exponentiation and a Super-Sign query involves $O(n_u + n_m + n'_m)$ multiplications and $O(1)$ exponentiations. Thus the time complexity of solving CDH problem is $t \approx t' + O(q_\sigma(n_u + n_m + n'_m)t_m + (q_c + q_\sigma)t_e)$. □

4.3. Efficiency Analysis

In this section, the proposed scheme is compared with some existing certificateless signature schemes in terms of efficiency. For efficiency comparison, we use the PBC library and select the Type A curve, conducting experiments on an Ubuntu22 virtual machine with the 12th Gen Intel(R) Core(TM) i7-12700H 2.70GHz processor and 16GB RAM. Then, the experiment results show that, every pairing operation (P) needs 1.58022ms, every multiplication ($Mul_{\mathbb{G}}$) in $\mathbb{G}$ needs 0.01114ms , every multiplication ($Mul_{\mathbb{G}_T}$) in $\mathbb{G}_T$ needs 0.00181ms, every exponentiation ($E_{\mathbb{G}}$) in $\mathbb{G}$ needs 0.00061ms, and every inversion (Inv) in $\mathbb{Z}_{q^*}$ needs 0.00281ms. The variables n_u, n_m, n'_m and n_p represent the output lengths of the hash functions, while $x|\mathbb{G}|$ denotes the binary length of x elements in $\mathbb{G}$. We present our results in Table 3 and Figure 2.

Table 3. The comparison of efficiency and signature length

Scheme	Signing cost	Verification cost	Signature length
Wu *et al.* [25]	$Inv + 3E_{\mathbb{G}} + (n_m + 1)Mul_{\mathbb{G}}$	$5P + Mul_{\mathbb{G}_T} + (n_u + n_m)Mul_{\mathbb{G}}$	$2\lvert\mathbb{G}\rvert$
Tseng *et al.* [26]	$6E_{\mathbb{G}} + (n_m + 2)Mul_{\mathbb{G}}$	$7P + 3Mul_{\mathbb{G}_T} + E_{\mathbb{G}} + (n_m + n_p + n_u + 1)Mul_{\mathbb{G}}$	$5\lvert\mathbb{G}\rvert$
Rastegari *et al.* [27]	$2E_{\mathbb{G}} + (n_m + 1)Mul_{\mathbb{G}}$	$7P + 3Mul_{\mathbb{G}_T} + (n_u + n_p + n_m)Mul_{\mathbb{G}}$	$4\lvert\mathbb{G}\rvert$
Ours	$6E_{\mathbb{G}} + (n_u + n_m + n'_m + 5)Mul_{\mathbb{G}}$	$6P + 4Mul_{\mathbb{G}_T} + (n_u + n_m + n'_m)Mul_{\mathbb{G}}$	$4\lvert\mathbb{G}\rvert$

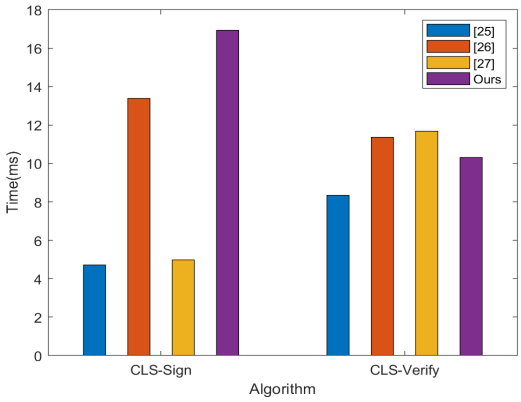

Figure 2. Sign and Verify time of the four schemes

5. EXPANSION: CERTIFICATELESS CLOUD AUDITING SCHEME

We can easily take advantage of the structure of the certificateless signature scheme to construct a CLCA scheme against Super adversaries in STM. In general, the CLCA scheme can be specified by nine algorithms: **Setup**, **PartialKeyExt**, **SetSecretValue**, **SetPrivateKey**, **SetPublicKey**, **TagGen**, **Challenge**, **Respond** and **Verify**. The first five algorithms are similar to those in the certificateless signature; **TagGen**, **Challenge**, **Respond** and **Verify** are as follows:

- **TagGen**$(params, M, sk_{ID}) \rightarrow \{t_i\}_{i=1,\ldots,n}$: Given the public parameter $params$, a file M, and the user's private key sk_{ID}. Splits M into n blocks. For each block m_i, the tag is t_i. This algorithm outputs tags $\{t_i\}_{i=1,\ldots,n}$ for the file.
- **Challenge**$(params, I) \rightarrow chal$: Given the public parameter $params$, a set $I \in [1, n]$. This algorithm outputs the challenge $chal$
- **Respond**$(params, chal, \mathcal{M}, \mathcal{T}) \rightarrow res$: Given the public parameter $params$, a challenge $chal$, a set of messages $\mathcal{M}$, and a set of tags $\mathcal{T}$. This algorithm outputs the response res.
- **Verify**$(params, chal, res) \rightarrow 1/0$: Given the public parameter $params$, a challenge $chal$, and a response res. This algorithm outputs 0 or 1.

Now we can outline our CLCA scheme: **Setup**, **PartialKeyExt**, **SetSecretValue**, **SetPrivateKey** and **SetPublicKey**: Identical to our certificateless signature scheme.

- **TagGen**$(params, M, sk_{ID}) \rightarrow \{t_i\}_{i=1,\ldots,n}$: Given the public parameter $params$, a file M, and the user's private key sk_{ID}. Splits M into n blocks. For each block m_i, we can compute

$$t_{i,1} = g_2^{\alpha} \left(u' \prod_{k \in \mathcal{U}} \tilde{u}_k \right)^{h_{ID}+h'} \left(v'_1 \prod_{k \in \mathcal{V}_1} \tilde{v}_{1,k} \right)^{h_v} g_3^{x_{ID} m_i} \left(v'_2 \prod_{k \in \mathcal{V}_2} \tilde{v}_{2,k} \right)^{r}$$

$$t_{i,2} = psk_{ID,2} \cdot g^{h'} = g^{h_{ID}+h'}, t_{i,3} = g^{h_v}, t_{i,4} = g^{r}$$

and output $t_i = (t_{i,1}, t_{i,2}, t_{i,3}, t_{i,4})$ as the tag. Note that we handle the index i in the same manner as the message m in the signature scheme.

- **Challenge**$(params, I) \rightarrow chal$: Given the public parameter $params$, a set $I \in [1, n]$. Select $s_i \leftarrow\$ \mathbb{Z}_q$ for $i \in I$ and output $chal = \{(i, s_i) | i \in I\}$.
- **Respond**$(params, chal, \mathcal{M}, \mathcal{T}) \rightarrow res$: Given the public parameter $params$, a challenge $chal$, a set of messages $\mathcal{M}$, and a set of tags $\mathcal{T}$. Compute

$$\omega_1 = \prod_{i \in I} t_{i,1}^{s_i}$$
$$\omega_2 = \prod_{i \in I} t_{i,2}^{s_i}$$
$$\mu = \Sigma_{i \in I} s_i m_i$$

and output $res = (\omega_1, \omega_2, \{t_{i,3}^{s_i}\}_{i \in I}, \{t_{i,4}^{s_i}\}_{i \in I}, \mu)$.
- **Verify**$(params, chal, res) \rightarrow 1/0$: Given the public parameter $params$, a challenge $chal$, and a response res, verify whether

$$e(\omega_1, g) \stackrel{?}{=} e(g_1, g_2)^{\Sigma_{i \in I} s_i} e(g_3, pk_{ID})^{\mu} e\left(u' \prod_{k \in \mathcal{U}} \tilde{u}_i, \omega_2\right) \prod_{i \in I} e\left(v_1' \prod_{k \in \mathcal{V}_{1,i}} \tilde{v}_{1,k}, t_{i,3}^{s_i}\right) e\left(v_2' \prod_{k \in \mathcal{V}_{2,i}} \tilde{v}_{2,k}, t_{i,4}^{s_i}\right)$$

holds or not. Output 1 if the equality holds; otherwise output 0.

6. DISCUSSION

This study introduces a novel certificateless signature scheme and demonstrates its security against Super adversaries in the STM. While previous research has proposed certificateless signature schemes in the STM, no scheme has been proven secure against Super adversaries in the STM. Additionally, we extend the structure of the proposed certificateless signature scheme to develop a CLCA scheme, which is also provably secure against Super adversaries in the STM. As far as we are aware, no existing schemes offer a similar level of security.

Based on our experimental results, although the efficiency of our scheme has not yet reached that of the most advanced schemes, the overhead is still within an acceptable range. Future work will focus on improving efficiency while maintaining the same level of security, such as by incorporating blockchain technology[38] to reduce computational and storage overhead. Furthermore, the scheme can be deployed as a component in systems such as Verifiable Query Layer (VQL)[39], enhancing system functionality and security, which presents a promising direction for further research.

7. CONCLUSION

To the best of our knowledge, no certificateless signature scheme has been proposed in the literature that is secure against Super adversaries without random oracles. In this paper, we introduce a certificateless signature scheme against Super adversaries based on the CDH problem. We then employ a similar technique to present a CLCA scheme with the same level of security. Our primary approach combines Water's signature scheme[29] with Paterson's IBS scheme[31], which is akin to the methodology used by Huang *et al*[20].

DECLARATIONS

Acknowledgments

We would like to express our sincere gratitude to the editor and anonymous reviewers for their helpful and professional comments and guidance in improving our manuscript.

Authors' contributions

Made substantial contributions to the design and proof of the proposed scheme: S. Yao; Provided administrative support and supervision: G. Wu; Performed a substantial review of the proposed scheme, along with editorial work and proofreading: X. Liu.

Availability of data and materials

The rata data supporting the findings of this study are available within this Article and its Supplementary Material. Further data are available from the corresponding authors upon request.

Financial support and sponsorship

This paper is supported by the National Natural Science Foundation of China (No.62372103, No.62002058) and the Natural Science Foundation of Jiangsu Province (No.BK20200391).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Diffie W, Hellman ME. New directions in cryptography. *IEEE Trans Inf Theory* 1976;22:644–54. DOI
2. Shamir A. identity-based cryptosystems and signature schemes. In: Blakley GR, Chaum D, editors. Proceedings of the Annual International Cryptology Conference-CRYPTO; 1984 Aug 19-22; Santa Barbara, USA. Springer; 1984. pp. 47–53. DOI
3. Al-Riyami SS, Paterson KG. Certificateless public key cryptography. In: Laih C, editor. Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security-ASIACRYPT; 2003 Nov 30 - Dec 4; Taipei, Taiwan. Springer; 2003. pp. 452–73. DOI
4. Huang X, Susilo W, Mu Y, Zhang F. on the security of certificateless signature schemes from asiacrypt 2003. In: Desmedt Y, Wang H, Mu Y, Li Y, editors. Proceedings of the 4th International Conference on Cryptology and Network Security-CANS; 2005 Dec 14-16; Xiamen, China. Springer; 2005. pp. 13–25. DOI
5. Huang X, Mu Y, Susilo W, Wong DS, Wu W. Certificateless signatures: new schemes and security models. *Comput J* 2012;55:457–74. DOI
6. Zhang Z, Wong DS, Xu J, Feng D. certificateless public-key signature: security model and efficient construction. In: Zhou J, Yung M, Bao F, editors. Proceedings of the 4th International Conference on Applied Cryptography and Network Security-ACNS; 2006 Jun 6-9; Singapore, Singapore. Springer; 2006. pp. 293–308. DOI
7. Huang X, Mu Y, Susilo W, Wong DS, Wu W. Certificateless signature revisited. In: Pieprzyk J, Ghodosi H, Dawson E, editors. Proceedings of the 12th Australasian Conference on Information Security and Privacy-ACISP; 2007 Jul 2-4; Townsville, Australia. Springer; 2007. pp. 308–22. DOI
8. Choi KY, Park JH, Hwang JY, Lee DH. Efficient certificateless signature schemes. In: Katz J, Yung M, editors. Proceedings of the 5th International Conference on Applied Cryptography and Network Security-ACNS; 2007 Jun 5-8; Zhuhai, China. Springer; 2007. pp. 443–58. DOI
9. Tso R, Yi X, Huang X. Efficient and short certificateless signature. In: Franklin MK, Hui LCK, Wong DS, editors. Proceedings of the 7th International Conference on Cryptology and Network Security-CANS; 2008 Dec 2-4; Hong-Kong, China. Springer; 2008. pp. 64–79. DOI
10. Zhang L, Zhang F, Zhang F. New efficient certificateless signature scheme. In: Denko MK, Shih C, Li K, et al., editors. Proceedings of the Emerging Directions in Embedded and Ubiquitous Computing-EUC; 2007 Dec 17-20; Taipei, Taiwan. Springer; 2007. pp. 692–703. DOI
11. Hu BC, Wong DS, Zhang Z, Deng X. Key replacement attack against a generic construction of certificateless signature. In: Batten LM, Safavi-Naini R, editors. Proceedings of the 11th Australasian Conference on Information Security and Privacy-ACISP; 2006 Jul 3-5; Melbourne, Australia. Springer; 2006. pp. 235–46. DOI
12. Chen Y, Tso R, Horng G, Fan C, Hsu R. Strongly secure certificateless cignature: cryptanalysis and improvement of two schemes. *J*

Inf Sci Eng 2015;31:297–314. Available from: http://www.iis.sinica.edu.tw/page/jise/2014/2015/201501_16.html.[Last accessed 29 Oct 2024].
13. Canetti R, Goldreich O, Halevi S. The random oracle methodology, revisited (preliminary version). In: Vitter JS, editor. Proceedings of the 30th Annual ACM Symposium on Theory of Computing-STOC; 1998 May 23-26; Dallas, USA. ACM; 1998. pp. 209–18. DOI
14. Liu JK, Au MH, Susilo W. Self-generated-certificate public key cryptography and certificateless signature / encryption scheme in the standard model. In: Deng R, Samarati P, Baoand F, Miller S, editors. Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security-ASIACCS; 2007 Mar 20-22; Singapore, Singapore. ACM; 2007. pp. 273–83. DOI
15. Xiong H, Qin Z, Li F. An improved certificateless signature scheme secure in the standard model. *Fundam Informaticae* 2008;88:193–206. Available from: http://content.iospress.com/articles/fundamenta-informaticae/fi88-1-2-09.[Last accessed 29 Oct 2024].
16. Xia Q, Xu C, Yu Y. Key replacement attack on two certificateless signature schemes without random oracles. *Key Eng Mat* 2010;439-440:1606–11. DOI
17. Yuan H, Zhang F, Huang X, et al. Certificateless threshold signature scheme from bilinear maps. *Inf Sci* 2010;180:4714–28. DOI
18. Yu Y, Mu Y, Wang G, Xia Q, Yang B. Improved certificateless signature scheme provably secure in the standard model. *IET Inf Secur* 2012;6:102–10. DOI
19. Cheng L, Wen Q. Provably secure and efficient certificateless signature in the standard model. *Int J Inf Commun Technol* 2015;7:287–301. DOI
20. Hung Y, Huang S, Tseng Y, Tsai T. Certificateless signature with strong unforgeability in the standard model. *Informatica* 2015;26:663–84. Available from: http://content.iospress.com/articles/informatica/inf1073.[Last accessed 29 Oct 2024].
21. Yang W, Weng J, Luo W, Yang A. Strongly unforgeable certificateless signature resisting attacks from malicious-but-passive KGC. *Secur Commun Networks* 2017;2017:1–8. DOI
22. Pang L, Hu Y, Liu Y, Xu K, Li H. Efficient and secure certificateless signature scheme in the standard model. *Int J Commun Syst* 2017;30. DOI
23. Wang F, Xu L. Strongly secure certificateless signature scheme in the standard model with resisting malicious-but-passive KGC attack ability. *J Inf Sci Eng* 2017;33:873–89. Available from: https://jise.iis.sinica.edu.tw/JISESearch/pages/View/PaperView.jsf?keyId=157_2046.[Last accessed 29 Oct 2024].
24. Shim K. A new certificateless signature scheme provably secure in the standard model. *IEEE Syst J* 2019;13:1421–30. DOI
25. Wu C, Huang H, Zhou K, Xu C. Cryptanalysis and improvement of a new certificateless signature scheme in the standard model. *China Commun* 2021;18:151–60. DOI
26. Tseng Y, Fan C, Chen C. Top-level secure certificateless signature scheme in the standard model. *IEEE Syst J* 2019;13:2763–74. DOI
27. Rastegari P, Susilo W. On random-oracle-Free Top-Level Secure Certificateless Signature Schemes. *Comput J* 2022;65:3049–61. DOI
28. Yang X, Wen H, Liu L, Ren N, Wang C. Blockchain-enhanced certificateless signature scheme in the standard model. *Math Biosci Eng* 2023;20:1271–73. DOI
29. Waters B. Efficient identity-based encryption without random oracles. In: Cramer R, editor. Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques-EUROCRYPT; 2005 May 22-26; Aarhus, Denmark. Springer; 2005. pp. 114–27. DOI
30. Hu BC, Wong DS, Zhang Z, Deng X. Certificateless signature: a new security model and an improved generic construction. *Des Codes Cryptogr* 2007;42:109–26. DOI
31. Paterson KG, Schuldt JCN. Efficient identity-based signatures secure in the standard model. In: Batten LM, Safavi-Naini R, editors. Proceedings of the 11th Australasian Conference on Information Security and Privacy-ACISP; 2006 Jul 3-5; Melbourne, Australia. Springer; 2006. pp. 207–22. DOI
32. Ateniese G, Burns RC, Curtmola R, et al. Provable data possession at untrusted stores. In: Ning P, di Vimercati SDC, Syverson PF, editors. Proceedings of the 2007 ACM Conference on Computer and Communications Security-CCS; 2007 Oct 28-31; Alexandria, USA. ACM; 2007. pp. 598–609. DOI
33. Juels A, Jr BSK. Pors: proofs of retrievability for large files. In: Ning P, di Vimercati SDC, Syverson PF, editors. Proceedings of the 2007 ACM Conference on Computer and Communications Security-CCS; 2007 Oct 28-31; Alexandria, USA. ACM; 2007. pp. 584–97. DOI
34. Ma M, Weber J, van den Berg J. Secure public-auditing cloud storage enabling data dynamics in the standard model. In: Proceedings of the Third International Conference on Digital Information, Data Mining, and Wireless Communications-DIPDMWC; 2016 July 6-8; Moscow, Russia. IEEE; 2016. pp. 170–75. DOI
35. Zhang J, Li P, Mao J. IPad: ID-based public auditing for the outsourced data in the standard model. *Clust Comput* 2016;19:127–38. DOI
36. Deng L, Wang B, Wang T, Feng S, Li S. Certificateless provable data possession scheme with provable security in the standard model suitable for cloud storage. *IEEE Trans Serv Comput* 2023;16:3986–98. DOI
37. Yang G, Han L, Bi J, Wang F. A collusion-resistant certificateless provable data possession scheme for shared data with user revocation. *Clust Comput* 2024;27:2165–79. DOI
38. Xu Y, Ren J, Zhang Y, Zhang C, Shen B, et al. Blockchain empowered arbitrable data auditing scheme for network storage as a service. *IEEE Trans Serv Comput* 2020;13:289–300. DOI
39. Wu H, Peng Z, Guo S, Yang Y, Xiao B. VQL: Efficient and verifiable cloud query services for blockchain systems. *IEEE Trans Parallel Distributed Syst* 2022;33:1393–406. DOI