Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 728591, 16 pages
http://dx.doi.org/10.1155/2014/728591

Research Article

Secure Certificateless Signature with Revocation in the Standard Model

Tung-Tso Tsai, Sen-Shan Huang, and Yuh-Min Tseng

Department of Mathematics, National Changhua University of Education, Chang-Hua City 500, Taiwan

Correspondence should be addressed to Yuh-Min Tseng; ymtseng@cc.ncue.edu.tw

Received 6 May 2014; Revised 3 September 2014; Accepted 3 October 2014; Published 19 November 2014

Academic Editor: Kwok-Wo Wong

Certificateless public key cryptography is very attractive in solving the key escrow problem which is inherent in identity- (ID-) based public key cryptography. In the past, a large number of certificateless cryptographic schemes and protocols were presented, but a secure certificateless signature in the standard model (without random oracles) is still not accessible until now. To the best of our knowledge, all the previously proposed certificateless signature schemes were insecure under a considerably strong security model in the sense that they suffered from outsiders' key replacement attacks or the attacks from the key generation center (KGC). In this paper, we propose a certificateless signature scheme without random oracles. Moreover, our scheme is secure under the strong security model and provides a public revocation mechanism, called revocable certificateless signature (RCLS). Under the standard computational Diffie-Hellman assumption, we formally demonstrate that our scheme possesses existential unforgeability against adaptive chosen-message attacks.

1. Introduction

In 2001, Boneh and Franklin [1], led by Shamir's idea [2], realized a practical construction of an identity- (ID-) based public key system (ID-PKS) by using bilinear pairings, such as Weil, Tate, and Ate pairings. Subsequently, the study of ID-based cryptographic mechanisms using bilinear pairings has received a great attention from researchers. A large number of literatures, such as [3–7], have been presented. An ID-based public key system consists of two roles, namely, a trusted private key generator (PKG) and users. A user's identity information, such as social security number, e-mail address, and IP address, is viewed as her/his public key and the corresponding private key is computed and issued secretly by the trusted PKG. Evidently, the PKG owns all the users' private keys so that it can decrypt ciphertexts or sign messages on behalf of any user as it wishes. Hence, an ID-based public key system always inherits an imperfection—the key escrow problem.

To overcome the key escrow problem in ID-based public key systems, Gentry [8] introduced the concept of certificate-based public key system, which contains two roles, namely, a key generation center (KGC) and users. Here, we roughly describe Gentry's system. A user first generates the public key and then sends it to the KGC. The KGC generates a certificate for the user's public key and then sends it to the user. The certificate also acts as a decryption or signing key of the user.

In 2003, Al-Riyami and Paterson [9] proposed a new paradigm for public key cryptography by combining the traditional public key system (PKS) with ID-PKS, termed certificateless public key system. As a result, certificates are no longer needed, but the two roles, namely, a KGC and users, remain in their certificateless public key system. A user's private key consists of two components: a partial private key and a secret key. They are generated by the KGC and the user, independently. In this case, the KGC does not have access to the user's full private key due to the lack of the secret key generated by the user. Hence, this certificateless design resolves the key escrow problem in ID-based public key systems. Subsequently, a numer of certificateless cryptographic schemes and protocols have been studied [10–22].

Al-Riyami and Paterson [9] presented a security model for certificateless public key cryptography. It consists of two types of adversaries, namely, outsiders (Type I adversary) and the KGC (Type II adversary). They also presented a concrete certificateless signature (CLS) scheme but did not offer security notion for this kind of scheme. Two years later, Huang et al. [11] first provided formal security notion for

CLS schemes and pointed out that Al-Riyami and Paterson's scheme above suffers from key replacement attacks from outsiders in the sense that, by replacing a user's public key, an outsider can forge valid signatures without the knowledge of the user's partial private key. In 2004, Yum and Lee [23] gave a generic construction of CLS schemes by combining an ID-based signature scheme with a signature scheme based on traditional public key systems. In 2006, Hu et al. [12] presented a considerably strong security model for CLS schemes and pointed out a security drawback in Yum and Lee's construction. Since then, Hu et al.'s security model is considered promising and is generally adopted to formalize the security of CLS schemes.

To improve the efficiency of CLS schemes, several constructions of CLS were built and discussed such as [24–27]. In addition, for providing relatively short signature, Huang et al. [14] proposed a certificateless short signature scheme, which was shown to suffer from key replacement attacks by Shim [28]. Recently, Chen et al. [29] wrote a survey article on CLS schemes, in which they also presented a secure certificateless short signature scheme.

The security of all the CLS schemes mentioned above was based on the random oracle model [30]. Although these schemes offer better performance, they could be insecure when random oracles are instantiated with some particular hash functions such as SHA-1. In order to compensate this situation, a secure CLS scheme in the standard model (without random oracles) must be constructed. Based on the ID-based signature scheme presented by Paterson and Schuldt [31], Liu et al. [16] proposed the first CLS scheme in the standard model. However, Xiong et al. [17] pointed out that Liu et al.'s scheme is insecure against KGC attacks. To withstand the KGC attacks, two CLS schemes [18, 21] were independently proposed but later shown to be insecure [32, 33]. To our best knowledge, a secure CLS in the standard model (without random oracles) is still not accessible. All the previously proposed CLS schemes in the standard model were insecure under Hu et al.'s strong security model in [12] in the sense that they suffered from key replacement attacks from outsiders (Type I adversary) or attacks from the key generation center (Type II adversary).

A public key system should provide an efficient revocation mechanism to revoke misbehaving/compromised users. In traditional public key systems, users are able to know the revoked public keys by querying the certificate revocation list (CRL) [34] which contains all the revoked public keys. Evidently, the CRL approach is no longer suited for the certificateless public key system due to the lack of the usage of certificates. So, revoking misbehaving/compromised users in certificateless public key system has received attention from cryptographic researchers. Recently, Tsai and Tseng [22] and Shen et al. [35], independently, used the revocation technique presented by Tseng and Tsai [36] to propose a revocable certificateless public-key encryption (RCL-PKE) scheme. However, there was little work in constructing a revocable CLS scheme without random oracles.

Our Construction. In this paper, we will propose a CLS scheme with revocation in the standard model, called *revocable certificateless signature* (RCLS). This scheme includes two merits.

(1) Under the standard computational Diffie-Hellman assumption, our scheme is secure under Hu et al.'s strong security model. We will formally demonstrate that our scheme possesses existential unforgeability against adaptive chosen-message attacks under the standard computational Diffie-Hellman assumption.

(2) Our CLS scheme provides a public revocation mechanism to revoke misbehaving/compromised users.

The rest of the paper is organized as follows. We give some preliminaries in Section 2. In Section 3, we review the framework and security notion for RCLS schemes. A concrete RCLS scheme in the standard model is proposed in Section 4. We analyze the security of the proposed scheme in Section 5. In Section 6, we present discussions and comparisons between some existing schemes and ours. Finally, a concluding remark is drawn in Section 7.

2. Preliminaries

In this section, we will review some fundamentals required in this sequel, namely, the concept of bilinear pairings and the related assumption.

2.1. Bilinear Pairings. In the sequel, $\mathbb{G}_1$ and $\mathbb{G}_2$ denote two multiplicative cyclic groups of large prime order p. A map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is called an admissible bilinear map if it satisfies the following properties.

(1) The map $\hat{e}$ is bilinear: given $g, h \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$, we have $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$.

(2) The map $\hat{e}$ is nondegenerate: there exist $g, h \in \mathbb{G}_1$ such that $\hat{e}(g, h) \neq 1$.

(3) The map $\hat{e}$ is efficiently computable.

We refer the reader to previous literature, such as [1, 9], for a more comprehensive description of groups, maps, and other parameters.

2.2. Related Mathematical Assumption. Here, we introduce the computational Diffie-Hellman (CDH) problem and define its corresponding security assumption (CDH assumption) on which our scheme is based.

Definition 1. Let g be a generator of $\mathbb{G}_1$. Given $\langle g, g^a, g^b \rangle$ with unknown a, b randomly chosen from $\mathbb{Z}_p^*$, the CDH problem is to compute g^{ab}. The associated CDH assumption states that there exists no probabilistic polynomial-time adversary $\mathcal{A}$ that can solve the CDH problem with nonnegligible probability. The successful probability (advantage) of the adversary $\mathcal{A}$ is presented as

$$\mathrm{Adv}_{\mathcal{A}} = \Pr\left[\mathcal{A}\left(\left\langle g, g^a, g^b \right\rangle\right) = g^{ab} : g \in \mathbb{G}_1, a, b \in \mathbb{Z}_p^*\right], \quad (1)$$

where the probability is over the random choice consumed by the adversary $\mathcal{A}$.

3. Framework and Security Notion of Revocable Certificateless Signature

In this section, we will define the framework and security notion for our RCLS scheme. In 2003, Al-Riyami and Paterson [9] presented a concrete certificateless signature (CLS) scheme, but did not give formal security notion for such schemes. Later, Yum and Lee [23] and Huang et al. [11] defined formal security notion for CLS schemes. Furthermore, Hu et al. [12] enhanced the definitions in [11, 23] to permit stronger queries. Since then, Hu et al.'s security model is generally adopted to formalize the security of CLS schemes. By modifying Hu et al.'s framework and security notion for CLS schemes [12], we present a framework of our RCLS scheme by adding *time key update algorithm*. Under this new framework, a user's partial private key consists of two components, namely, an initial secret key (fixed since being issued) and a time update key (altered every period of time). As a result, the issue on time update key queries must be addressed in the security notion for CLS schemes.

Definition 2. A revocable certificateless signature (RCLS) is specified by six algorithms, namely, the system setup, the initial key extract, the time key update, the user key generation, the signing, and the verification.

(i) *System setup*: this algorithm takes security parameter l and the total number z of periods as input and returns a system secret key, a time secret key, and the public parameters *Params*. We assume that *Params* are publicly and authentically available in all the following algorithms.

(ii) *Initial key extract*: this algorithm takes the system secret key and a user's identity ID as input and returns the user's initial secret key D_{ID} to the user via a secure channel.

(iii) *Time key update*: this algorithm takes the time secret key, a user's identity ID, and a period t as input and returns the user's time update key $T_{\mathrm{ID},t}$ to the user via a public channel.

(iv) *User key generation*: this algorithm takes the public parameters *Params* and ID as input and outputs the secret key $\mathrm{SK}_{\mathrm{ID}}$ and public key $\mathrm{PK}_{\mathrm{ID}}$.

(v) *Signing*: this algorithm takes a user's initial secret key D_{ID}, the user's time update key $T_{\mathrm{ID},t}$, the user's secret key $\mathrm{SK}_{\mathrm{ID}}$, and a message M as input and returns a signature σ.

(vi) *Verification*: taking a signature σ, a message M, a user identity ID, a period t, and the user's public key $\mathrm{PK}_{\mathrm{ID}}$ as input; the algorithm outputs either "accept" or "reject."

Remark 3. The *system setup*, the *initial key extract*, and the *time key update* algorithms are run by the KGC, while the *user key generation* algorithm is run by the user.

There are three types of adversaries in our security notion for RCLS schemes. In the following, we will employ a security game to model security notion for RCLS schemes. The security game describes the interaction between a challenger and an adversary. Before introducing the security game, we first describe capabilities of Type I, Type II, and Type III adversaries, respectively.

(i) Type I adversary (outsider): an adversary of this type does not have access to the system secret key and time secret key, but she/he can replace the public key of any entity with another of her/his own choice.

(ii) Type II adversary (KGC): an adversary of this type is the KGC who knows the system secret key and time secret key. The KGC can produce the initial secret key and time update key of arbitrary identity, but it is forbidden to replace the public key of any identity at any time.

(iii) Type III adversary (revoked user): an adversary of this type used to be a member of the system before being revoked by the system. A revoked user still owns the initial secret key although the system stops issuing the current time update key for her/him.

Definition 4. A RCLS scheme is existential unforgeable against chosen message attack if there is no probabilistic polynomial-time (PPT) adversary $\mathscr{A}$ which has non-negligible advantage in the following security game.

(i) *Setup*. A challenger $\mathscr{B}$ takes a security parameter l and runs the *system setup* algorithm. The challenger $\mathscr{B}$ gives the public parameters *Params* to the adversary $\mathscr{A}$. Meanwhile, if $\mathscr{A}$ is of Type II adversary, $\mathscr{B}$ gives the system secret key and time secret key to $\mathscr{A}$. Otherwise, the challenger $\mathscr{B}$ keeps them for itself.

(ii) Phase 1. The adversary $\mathscr{A}$ may issue the *public key retrieve*, *public key replace*, *initial key extract*, *time key update*, *secret key extract*, and *signing queries* as follows.

(a) *Public key retrieve query (ID)*. When $\mathscr{A}$ issues this query with an identity ID, the challenger $\mathscr{B}$ returns the corresponding public key $\mathrm{PK}_{\mathrm{ID}}$ to $\mathscr{A}$.

(b) *Public key replace query* (ID, PK'_{ID}). In such a query, the adversary replaces the public key of the identity ID with $\mathrm{PK}'_{\mathrm{ID}}$. The challenger records the replacement.

(c) *Initial key extract query (ID)*. When $\mathscr{A}$ issues this query with an identity ID, the challenger $\mathscr{B}$ runs the *initial key extract* algorithm to return the initial secret key D_{ID}.

(d) *Time key update query* (ID, t). When $\mathscr{A}$ issues this query with an identity ID in a period t, the challenger $\mathscr{B}$ runs the *time key update* algorithm to return the time update key $T_{\mathrm{ID},t}$.

(e) *Secret key extract query (ID)*. When $\mathscr{A}$ issues this query with an identity ID, the challenger $\mathscr{B}$ returns the secret key $\mathrm{SK}_{\mathrm{ID}}$. Here, the query is forbidden if the identity ID has already appeared in the *public key replace query*.

(f) *Signing query* (ID, t, M). When $\mathscr{A}$ issues this query with (ID, t, M), the challenger $\mathscr{B}$ uses

the initial secret key D_{ID}, the time update key $T_{\mathrm{ID},t}$, and the secret key $\mathrm{SK}_{\mathrm{ID}}$ to run the *signing* algorithm to obtain a signature σ on the message M. The challenger $\mathcal{B}$ then returns σ to $\mathcal{A}$ no matter whether the identity ID appears in the *public key replace query* or not.

Note that a Type I adversary can issue all types of queries except the *initial key extract query* on ID^*; a Type II adversary can issue all types of queries except the queries on public key replace and the *secret key extract query* on ID^*; a Type III adversary (revoked user) is allowed to issue all types of queries except the *time update key query* on (ID^*, t^*).

(iii) *Forgery*. The adversary $\mathcal{A}$ outputs $(\mathrm{ID}^*, t^*, M^*, \sigma^*)$. We say that the adversary $\mathcal{A}$ wins this game if the following conditions are satisfied.

(1) The response of the *verification* algorithm on $(\mathrm{ID}^*, t^*, M^*, \sigma^*)$ is "accept."

(2) $(\mathrm{ID}^*, t^*, M^*)$ has never been submitted during the *signing query* process.

Remark 5. Since a Type II adversary owns the system secret key and the time secret key, it is able to generate any initial secret key and time update key. Hence, a Type II adversary does not need to issue the queries on initial secret key and time update key.

4. Concrete Revocable Certificateless Signature Scheme

In this section, we describe a concrete revocable certificateless signature scheme that consists of six algorithms.

(i) *System setup*: the KGC takes a security parameter l as input and returns $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$, where $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is an admissible bilinear map and $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic groups of sufficiently large prime order p. Then, the KGC sets five collision-resistant hash functions $H_u : \{0,1\}^* \to \{0,1\}^{n_u}$, $H_t : \{0,1\}^* \to \{0,1\}^{n_t}$, $H_\zeta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\zeta}$, $H_\eta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\eta}$, and $H_w : \{0,1\}^* \to \{0,1\}^{n_w}$, where n_u, n_t, n_ζ, n_η, and n_w are fixed. Furthermore, the KGC randomly chooses five values u', t', ζ', η', $w' \in \mathbb{G}_1$ and five vectors $\overrightarrow{U} = (u_i)$, $\overrightarrow{T} = (t_j)$, $\overrightarrow{\zeta} = (\zeta_r)$, $\overrightarrow{\eta} = (\eta_s)$, $\overrightarrow{W} = (w_k)$, where $u_i, t_j, \zeta_r, \eta_s, w_k \in \mathbb{G}_1$ for $i = 1, 2, \ldots, n_u$, $j = 1, 2, \ldots, n_t$, $r = 1, 2, \ldots, n_\zeta$, $s = 1, 2, \ldots, n_\eta$, and $k = 1, 2, \ldots, n_w$. A generator g of $\mathbb{G}_1$ and two values $\alpha, \beta \in \mathbb{Z}_p^*$ are picked. The KGC computes $g_1 \in g^{\alpha+\beta} \in \mathbb{G}_1$ and randomly chooses $g_2 \in \mathbb{G}_1$ to set the system secret key as g_2^α and the time secret key as g_2^β. Then, the public parameters are presented as $Params = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_u, H_t, H_\zeta, H_\eta, H_w, g, g_1, g_2, u', \overrightarrow{U}, t', \overrightarrow{T}, \zeta', \overrightarrow{\zeta}, \eta', \overrightarrow{\eta}, w', \overrightarrow{W}\}$.

(ii) *Initial key extract*: given a user's identity $\mathrm{ID} \in \{0,1\}^*$, the KGC computes a bit string $v = H_u(\mathrm{ID}) = (v_1, v_2, \ldots, v_{n_u})$. The KGC then chooses a random value $r_v \in \mathbb{Z}_p^*$ and uses the system secret key to compute the user's initial secret key $D_{\mathrm{ID}} = (D_1, D_2) = (g_2^\alpha (u' \prod_{i=1}^{n_u} u_i^{v_i})^{r_v}, g^{r_v})$. The KGC transmits D_{ID} to the user via a secure channel.

(iii) *Time update key*: given a user's identity $\mathrm{ID} \in \{0,1\}^*$ and a period t, the KGC computes a bit string $vt = H_t(\mathrm{ID}, t) = (vt_1, vt_2, \ldots, vt_{n_t})$. The KGC then chooses a random value $r_t \in \mathbb{Z}_p^*$ and uses the time secret key to compute the user's time update key $T_{\mathrm{ID},t} = (T_1, T_2) = (g_2^\beta (t' \prod_{j=1}^{n_t} t_j^{vt_j})^{r_t}, g^{r_t})$. The KGC transmits $T_{\mathrm{ID},t}$ to the user via a public channel.

(iv) *User key generation*: given a user's identity ID, this algorithm selects two secret values $\lambda_1, \lambda_2 \in \mathbb{Z}_p^*$ and computes the user's public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2) = (g^{\lambda_1}, g^{\lambda_2})$. Then, the algorithm first computes two bit strings $vu = H_\zeta = (\mathrm{PK}_1, \mathrm{PK}_2) = (vu_1, vu_2, \ldots, vu_{n_\zeta})$ and $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2) = (vs_1, vs_2, \ldots, vs_{n_\eta})$. Finally, the algorithm uses the selected two secret values to compute the user's secret key $\mathrm{SK}_{\mathrm{ID}} = g_2^{\lambda_1} (\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r})^{\lambda_1} (\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s})^{\lambda_2}$.

(v) *Signing*: given a message $M \in \{0,1\}^*$, a signer computes a bit string $vm = H_w(M) = (vm_1, vm_2, \ldots, vm_{n_w})$. The signer chooses a random value $r_m \in \mathbb{Z}_p^*$ and computes g^{r_m}. The signer uses her/his initial secret key $D_{\mathrm{ID}} = (D_1, D_2)$, time update key $T_{\mathrm{ID},t} = (T_1, T_2)$, and secret key $\mathrm{SK}_{\mathrm{ID}}$ to compute a signature on the message M as follows:

$$
\begin{aligned}
\sigma &= (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \\
&= \left(D_1 \cdot T_1 \cdot \mathrm{SK}_{\mathrm{ID}} \left(w' \prod_{k=1}^{n_w} w_k^{vm_k} \right)^{r_m}, D_2, T_2, g^{r_m} \right) \\
&= \left(g_2^\alpha \left(u' \prod_{i=1}^{n_u} u_i^{v_i} \right)^{r_v} g_2^\beta \left(t' \prod_{j=1}^{n_t} t_j^{vt_j} \right)^{r_t} g_2^{\lambda_1} \left(\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r} \right)^{\lambda_1} \right. \\
&\quad \left. \cdot \left(\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s} \right)^{\lambda_2} \left(w' \prod_{k=1}^{n_w} w_k^{vm_k} \right)^{r_m}, g^{r_v}, g^{r_t}, g^{r_m} \right).
\end{aligned} \tag{2}
$$

(vi) *Verification*: given a period t, a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on a message M for a signer with an identity ID and the signer's public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2)$, a verifier first computes five bit strings $v = H_u(\mathrm{ID})$, $vt = H_t(\mathrm{ID}, t)$, $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2)$, and $vm = H_w(M)$. Finally, the verifier checks the following equation:

$$
\hat{e}(g, \sigma_1) = \hat{e}(g_1, g_2)\, \hat{e}\left(\sigma_2, u' \prod_{i=1}^{n_u} u_i^{v_i} \right) \hat{e}\left(\sigma_3, t' \prod_{j=1}^{n_t} t_j^{vt_j} \right)
$$

$$\cdot \widehat{e}\left(\mathrm{PK}_1, g_2\left(\zeta' \prod_{r=1}^{n_u} \zeta_r^{vu_r}\right)\right) \widehat{e}\left(\mathrm{PK}_2, \eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s}\right) \cdot \widehat{e}\left(\sigma_3, w' \prod_{k=1}^{n_w} w_k^{vm_k}\right). \tag{3}$$

5. Security Analysis

In this section, we establish three theorems to demonstrate that, under the CDH assumption, the proposed RCLS scheme offers existential unforgeability against adaptive chosen-message attacks for Type I, Type II, and Type III adversaries, respectively.

Theorem 6. *Under the CDH assumption, the proposed RCLS scheme is secure against Type I adversary (outsider). Concretely, if there is a Type I adversary that has an advantage ϵ against the proposed scheme within a running time τ, then we can construct an algorithm to solve the CDH problem with an advantage*

$$\epsilon' \geq \frac{\epsilon}{16 q_K q_S (q_E + q_S)(n_u + 1)(n_\zeta + 1)(n_w + 1)} \tag{4}$$

within a running time $\tau' = \tau + O((n_u \cdot q_E + n_t \cdot q_U + (n_\zeta + n_\eta) \cdot q_K + (n_u + n_t + n_\zeta + n_\eta + n_w) \cdot q_S)\tau_1 + (q_E + q_U + q_S + q_K)\tau_2)$, in which q_E, q_U, and q_S are the numbers of queries on initial key extract, time key update, and signing, respectively, q_K is the sum of the numbers of queries on public key replace and secret key extract, and τ_1 and τ_2 denote the executing time of a multiplication and an exponentiation in $\mathbb{G}_1$, respectively.

Proof. Assume that a Type I adversary $\mathscr{A}$ can forge a valid signature for the proposed RCLS scheme. We will construct an algorithm $\mathscr{B}$ that solves the CDH problem as follows. Without loss of generality, we assume that there exists a tuple $\langle \mathbb{G}_1, \mathbb{G}_2, \widehat{e} \rangle$, as mentioned in Section 2, and the algorithm $\mathscr{B}$ is given $g, g^a, g^b \in \mathbb{G}_1$, where a and b are unknown to $\mathscr{B}$. In order to compute g^{ab}, the algorithm $\mathscr{B}$ simulates a challenger in the following game.

(i) *Setup*. A challenger (algorithm) $\mathscr{B}$ first sets five collision-resistant hash functions $H_u : \{0,1\}^* \to \{0,1\}^{n_u}$, $H_t : \{0,1\}^* \to \{0,1\}^{n_t}$, $H_\zeta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\zeta}$, $H_\eta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\eta}$, and $H_w : \{0,1\}^* \to \{0,1\}^{n_w}$, where $n_u, n_t, n_\zeta, n_\eta$, and n_w are fixed. Note that the employed collision-resistant hash functions are not viewed as random oracles in our security proofs. The challenger $\mathscr{B}$ then sets $l_v = 2(q_E + q_S)$, $l_u = q_K$, and $l_m = 2q_S$ and chooses three integers k_v, k_u, and k_m at random, where $0 \leq k_v \leq n_u$, $0 \leq k_u \leq n_\zeta$, and $0 \leq k_m \leq n_w$. We assume that $l_v(n_u + 1) < p$, $l_u(n_\zeta + 1) < p$, and $l_m(n_w + 1) < p$ for the given values of q_E, q_K, q_S, n_u, n_ζ and n_w. The challenger $\mathscr{B}$ randomly selects the integers $x', x_1, \ldots, x_{n_u} \in \mathbb{Z}_{l_v}$, $y', y_1, \ldots, y_{n_u} \in \mathbb{Z}_p$, $z', z_1, \ldots, z_{n_t} \in \mathbb{Z}_p$, $a', a_1, \ldots, a_{n_\zeta} \in \mathbb{Z}_{l_u}$, $b', b_1, \ldots, b_{n_\eta} \in \mathbb{Z}_p$, $c', c_1, \ldots, c_{n_w} \in \mathbb{Z}_{l_m}$, and $d', d_1, \ldots, d_{n_w} \in \mathbb{Z}_p$.

Now, the challenger $\mathscr{B}$ constructs a set of public parameters as follows. The challenger $\mathscr{B}$ chooses a value $\beta \in \mathbb{Z}_p$ as the time secret key. The challenger $\mathscr{B}$ sets $g_1 = g^a g^\beta$ and $g_2 = g^b$. Furthermore, $\mathscr{B}$ computes $u' = g_2^{-l_v k_v + x'} g^{y'}$ and a vector $\overrightarrow{U} = (u_i)$, where $u_i = g_2^{x_i} g^{y_i}$ for $1 \leq i \leq n_u$; $t' = g^{z'}$ and a vector $\overrightarrow{T} = (t_j)$, where $t_j = g^{z_j}$ for $1 \leq j \leq n_t$; $\zeta' = g_2^{-1 - l_u k_u + a'}$ and a vector $\overrightarrow{\zeta} = (\zeta_r)$, where $\zeta_r = g_2^{a_r}$ for $1 \leq r \leq n_\zeta$; $\eta' = g^{b'}$ and a vector $\overrightarrow{\eta} = (\eta_s)$, where $\eta_s = g^{b_s}$ for $1 \leq s \leq n_\eta$; $w' = g_2^{-l_m k_m + c'} g^{d'}$ and a vector $\overrightarrow{W} = (w_k)$, where $w_k = g_2^{c_k} g^{d_k}$ for $1 \leq k \leq n_w$. Now, the challenger $\mathscr{B}$ has constructed a set of public parameters as $Params = \{\mathbb{G}_1, \mathbb{G}_2, \widehat{e}, H_u, H_t, H_\zeta, H_\eta, H_w, g, g_1, g_2, u', \overrightarrow{U}, t', \overrightarrow{T}, \zeta', \overrightarrow{\zeta}, \eta', \overrightarrow{\eta}, w', \overrightarrow{W}\}$.

Before performing *Queries* and *Forgery* between the adversary $\mathscr{A}$ and the challenger $\mathscr{B}$, we define seven functions J, E, Q, L, F, R, and K by

$$J(v) = y' + \sum_{i=1}^{n_u} v_i y_i; \qquad E(vt) = z' + \sum_{j=1}^{n_t} vt_j z_j;$$

$$Q(vs) = b' + \sum_{s=1}^{n_\eta} vs_r b_s; \qquad L(vm) = d' + \sum_{k=1}^{n_w} vm_k d_k;$$

$$F(v) = -l_v k_v + x' + \sum_{i=1}^{n_u} v_i x_i; \tag{5}$$

$$R(vu) = -l_u k_u + a' + \sum_{r=1}^{n_\zeta} vu_r a_r;$$

$$K(vm) = -l_m k_m + c' + \sum_{k=1}^{n_w} vm_k c_k.$$

Here, as before, $v = H_u(\mathrm{ID}) = (v_1, v_2, \ldots, v_{n_u})$ for an identity ID, $vt = H_t(\mathrm{ID}, t) = (vt_1, vt_2, \ldots, v_{n_t})$ for an identity ID in a period t, $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2) = (vu_1, vu_2, \ldots, vu_{n_\zeta})$ and $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2) = (vs_1, vs_2, \ldots, vs_{n_\eta})$ for a public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2)$, and $vm = H_w(M) = (vm_1, vm_2, \ldots, vm_{n_w})$ for a message M.

Finally, for the cumbersome notations defined above, we conclude with five relations to which will be referred frequently in the sequel; namely,

$$u' \prod_{i=1}^{n_u} u_i^{v_i} = g_2^{F(v)} g^{J(v)}; \qquad t' \prod_{j=1}^{n_t} t_j^{vt_j} = g^{E(vt)};$$

$$\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r} = g_2^{R(vu)-1}; \qquad \eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s} = g^{Q(vs)};$$
$$w' \prod_{k=1}^{n_w} w_k^{vm_k} = g_2^{K(vm)} g^{L(vm)}. \tag{6}$$

(ii) *Queries.* The challenger $\mathcal{B}$ maintains a list L of tuples of the form $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$. Initially the list is empty. The adversary $\mathcal{A}$ may make a number of queries in an adaptive manner as follows.

(a) *Public key retrieve query (ID)*: upon receiving a query for the public key of an identity ID, the challenger $\mathcal{B}$ responds to the query as follows.

(1) If ID appears in the list L, the challenger $\mathcal{B}$ responds with the corresponding $\mathrm{PK}_{\mathrm{ID}}$.

(2) If ID does not appear in the list L, the challenger $\mathcal{B}$ selects two secret values $\lambda_1, \lambda_2 \in \mathbb{Z}_p^*$, sets the public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2) = (g^{\lambda_1}, g^{\lambda_2})$, and then computes $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2) = (vu_1, vu_2, \ldots, vu_{n_\zeta})$, $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2) = (vs_1, vs_2, \ldots, vs_{n_\eta})$, and the secret key $\mathrm{SK}_{\mathrm{ID}} = g_2^{\lambda_1} (\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r})^{\lambda_1} (\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s})^{\lambda_2}$. The challenger $\mathcal{B}$ adds the tuple $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$ in the list L and returns $\mathrm{PK}_{\mathrm{ID}}$ as the query output.

(b) *Public key replace query (ID, PK'_{ID})*: upon receiving a query to replace the public key of an identity ID, the challenger $\mathcal{B}$ accesses the tuple $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$ in the list L. If ID appears in the list L, the challenger $\mathcal{B}$ replaces $\mathrm{PK}_{\mathrm{ID}}$ with $\mathrm{PK}'_{\mathrm{ID}}$. If ID does not appear in the list L, the challenger adds the tuple $\langle \mathrm{ID}, \perp, \perp, \mathrm{PK}'_{\mathrm{ID}}, \perp \rangle$ in the list L.

(c) *Initial key extract query (ID)*: upon receiving a query for the initial secret key of an identity ID, the challenger $\mathcal{B}$ first sets $v = H_u(\mathrm{ID})$, and then computes $F(v)$ and $J(v)$. If $F(v) = 0$, the challenger $\mathcal{B}$ aborts. Otherwise, the challenger $\mathcal{B}$ chooses a random value $r_v \in \mathbb{Z}_p$ and responds with the initial secret key D_{ID} generated by

$$D_{\mathrm{ID}} = (D_1, D_2) = \left((g^a)^{-J(v)/F(v)} \left(g_2^{F(v)} g^{J(v)} \right)^{r_v}, (g^a)^{-1/F(v)} g^{r_v} \right). \tag{7}$$

Here, $D_{\mathrm{ID}} = (D_1, D_2)$ is indeed a valid initial secret key since, by the first equality in (6),

$$D_1 = (g^a)^{-J(v)/F(v)} \left(g_2^{F(v)} g^{J(v)} \right)^{a/F(v)} \cdot \left(g_2^{F(v)} g^{J(v)} \right)^{r_v - a/F(v)} = (g^a)^{-J(v)/F(v)} \left(g_2^{F(v)} g^{J(v)} \right)^{a/F(v)} \cdot \left(u' \prod_{i=1}^{n_u} u_i^{v_i} \right)^{r_v - a/F(v)} = g_2^a \left(u' \prod_{i=1}^{n_u} u_i^{v_i} \right)^{r'_v};$$
$$D_2 = (g^a)^{-1/F(v)} g^{r_v} = g^{r_v - a/F(v)} = g^{r'_v}, \tag{8}$$

where $r'_v = r_v - a/F(v)$.

(d) *Time key update query (ID, t)*: upon receiving a query for the time update key of an identity ID in a period t, the challenger $\mathcal{B}$ first sets $vt = H_t(\mathrm{ID}, t)$ and then computes $E(vt)$. The challenger $\mathcal{B}$ chooses a random $r_t \in \mathbb{Z}_p$ and uses the time secret key β to compute the time update key as follows:

$$T_{\mathrm{ID},t} = (T_1, T_2) = \left(g_2^\beta \left(g^{E(vt)} \right)^{r_t}, g^{r_t} \right) = \left(g_2^\beta \left(t' \prod_{j=1}^{n_t} t_j^{vt_j} \right)^{r_t}, g^{r_t} \right). \tag{9}$$

(e) *Secret key extract query (ID)*: upon receiving a query for the secret key of an identity ID, the challenger $\mathcal{B}$ accesses the tuple $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$ in the list L. If ID appears in the list L, the challenger $\mathcal{B}$ responds with $\mathrm{SK}_{\mathrm{ID}}$. If ID does not appear in the list L, the challenger $\mathcal{B}$ runs *user key generation* algorithm to add the tuple $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$ in the list L and responds with $\mathrm{SK}_{\mathrm{ID}}$.

(f) *Signing query (ID, t, M, PK_{ID})*: consider a query for a message M, an identity ID, a period t, and a public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2)$. The challenger $\mathcal{B}$ first sets $v = H_u(\mathrm{ID})$, $vt = H_t(\mathrm{ID}, t)$, $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2)$, and $vm = H_w(M)$. $\mathcal{B}$ then computes $F(v)$, $J(v)$, $E(vt)$, $R(vu)$, $Q(vs)$, $K(vm)$, and $L(vm)$. If $K(vm) = 0$, the challenger $\mathcal{B}$ reports failure and terminates. Otherwise, the challenger $\mathcal{B}$ considers the following two cases.

Case 1: assume that the identity ID has previously appeared in the *public key replace query*. If $F(v) \neq 0$, the challenger $\mathcal{B}$ can compute the initial secret key $D_{\mathrm{ID}} = (D_1, D_2)$ as in the *initial key extract query*. In addition, the challenger $\mathcal{B}$ computes the time update key $T_{\mathrm{ID},t} = (T_1, T_2)$ as in the *time key update query*. The challenger $\mathcal{B}$ then chooses a random value $r_m \in \mathbb{Z}_p^*$ and responds whit the signature

$$\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) = \left(D_1 T_1 (\mathrm{PK}_1)^{(-L(vm)/K(vm))R(vu)} \right.$$

$$\cdot \left(\mathrm{PK}_2\right)^{Q(vs)} \times \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m}, D_2, T_2, \left(\mathrm{PK}_1\right)^{-R(vu)/K(vm)} g^{r_m}\Big). \tag{10}$$

Note that σ is indeed a valid signature since $K(vm) \neq 0$ and, by the equalities in (6),

$$\begin{aligned}\sigma_1 &= D_1 T_1 \left(g^{\lambda_1}\right)^{(-L(vm)/K(vm))R(vu)} \cdot \left(g^{\lambda_2}\right)^{Q(vs)} \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m} \\ &= D_1 T_1 g_2^{\lambda_1} \left(g_2^{R(vu)-1}\right)^{\lambda_1} \left(g^{Q(vs)}\right)^{\lambda_2} \cdot \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m - \lambda_1 R(vu)/K(vm)} \\ &= D_1 T_1 g_2^{\lambda_1} \left(\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r}\right)^{\lambda_1} \left(\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s}\right)^{\lambda_2} \cdot \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r'_m}; \\ \sigma_2 &= D_2; \qquad \sigma_3 = T_2; \\ \sigma_4 &= g^{-\lambda_1 R(vu)/K(vm)} g^{r_m} = g^{r'_m},\end{aligned} \tag{11}$$

where $r'_m = r_m - \lambda_1 R(vu)/K(vm)$.

On the other hand, if $F(v) = 0$, the challenger $\mathscr{B}$ first computes the time update key $T_{\mathrm{ID},t} = (T_1, T_2)$ as in the *time key update query*. Then, the challenge $\mathscr{B}$ chooses two random values $r_v, r_m \in \mathbb{Z}_p^*$ and responds with the signature

$$\begin{aligned}\sigma &= (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \\ &= \Big(\left(g^a\right)^{-L(vm)/K(vm)} \left(g_2^{F(v)} g^{J(v)}\right)^{r_v} \times T_1 \left(\mathrm{PK}_1\right)^{(-L(vm)/K(vm))R(vu)} \cdot \left(\mathrm{PK}_2\right)^{Q(vs)} \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m}, \\ &\quad g^{r_v}, T_2, g_1^{-1/K(vm)} \left(\mathrm{PK}_1\right)^{-R(vu)/K(vm)} g^{r_m}\Big).\end{aligned} \tag{12}$$

Note that σ is also a valid signature since, by (6),

$$\begin{aligned}\sigma_1 &= \left(g^a\right)^{-L(vm)/K(vm)} \left(g_2^{F(v)} g^{J(v)}\right)^{r_v} \times T_1 \left(g^{\lambda_1}\right)^{(-L(vm)/K(vm))R(vu)} \cdot \left(g^{\lambda_2}\right)^{Q(vs)} \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m} \\ &= g_2^a \left(g_2^{F(v)} g^{J(v)}\right)^{r_v} T_1 g_2^{\lambda_1} \left(g_2^{R(vu)-1}\right)^{\lambda_1} \left(g^{Q(vs)}\right)^{\lambda_2} \cdot \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m - (a+\lambda_1 R(vu))/K(vm)} \\ &= g_2^a \left(u' \prod_{i=1}^{n_u} u_i^{v_i}\right)^{r_v} T_1 g_2^{\lambda_1} \left(\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r}\right)^{\lambda_1} \cdot \left(\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s}\right)^{\lambda_2} \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r'_m}; \\ \sigma_2 &= g^{r_v}; \qquad \sigma_3 = T_2; \\ \sigma_4 &= \left(g^a\right)^{-1/K(vm)} \left(g^{\lambda_1}\right)^{-R(vu)/K(vm)} g^{r_m} = g^{r_m - (a+\lambda_1 R(vu))/K(vm)} = g^{r'_m},\end{aligned} \tag{13}$$

where $r'_m = r_m - (a + \lambda_1 R(vu))/K(vm)$.

Case 2: assume that the identity ID has not previously appeared in the *public key replace query*. The challenger $\mathscr{B}$ computes the time update key $T_{\mathrm{ID},t} = (T_1, T_2)$ as in the *time key update query*. If $F(v) \neq 0$, the challenger $\mathscr{B}$ can compute the initial secret key (D_1, D_2) as in the *initial key extract query* and accesses the list L to obtain the corresponding secret key $\mathrm{SK}_{\mathrm{ID}}$. The challenger $\mathscr{B}$ chooses a random value $r_m \in \mathbb{Z}_p^*$ and then responds with the signature

$$\sigma = \left(D_1 T_1 \mathrm{SK}_{\mathrm{ID}} \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m}, D_2, T_2, g^{r_m}\right). \tag{14}$$

If $F(v) = 0$, then $\mathscr{B}$ chooses two random values $r_v, r_m \in \mathbb{Z}_p^*$ and responds with the signature

$$\sigma = \left(\left(g^a\right)^{-L(vm)/K(vm)} \left(u' \prod_{i=1}^{n_u} u_i^{v_i}\right)^{r_v} T_1 \mathrm{SK}_{\mathrm{ID}} \cdot \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m}, g^{r_v}, T_2, g_1^{-1/K(vm)} g^{r_m}\right). \tag{15}$$

(iii) *Forgery.* Assume that the adversary $\mathscr{A}$ generates a valid signature $\sigma^* = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ for ID^* on M^* in t^*, where ID^*, t^*, and M^* are the target identity, period, and message, respectively. The challenger $\mathscr{B}$ first accesses to the list L to obtain $\mathrm{PK}_{\mathrm{ID}^*} = (\mathrm{PK}_1, \mathrm{PK}_2)$. The challenger $\mathscr{B}$ then computes $v^* = H_u(\mathrm{ID}^*)$, $vt^* = H_t(\mathrm{ID}^*, t^*)$, $vu^* = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vs^* = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vm^* = H_w(M^*)$, $F(v^*)$, $J(v^*)$, $E(vt^*)$, $R(vu^*)$, $Q(vs^*)$, $L(vm^*)$, and $K(vm^*)$. If $F(v^*) \neq 0$, $R(vu^*) \neq 0$, or $K(vm^*) \neq 0$, the challenger $\mathscr{B}$ aborts. Otherwise, that is, when $F(v^*) = R(vu^*) = K(vm^*) = 0$, the challenger $\mathscr{B}$, by using (6), computes g^{ab} as follows:

$$\begin{aligned}&\frac{\sigma_1}{\left(\sigma_2^{J(v^*)}\right)\left(\sigma_3^{E(vt^*)}\right)\left(\mathrm{PK}_2^{Q(vs^*)}\right)\left(\sigma_4^{L(vm^*)}\right) g_2^\beta} \\ &= \frac{g_2^a \left(u' \prod_{i=1}^{n_u} u_i^{v_i}\right)^{r_v} g_2^\beta \left(t' \prod_{j=1}^{n_t} t_j^{vt_j}\right)^{r_t} g_2^{\lambda_1} \left(\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r}\right)^{\lambda_1}}{\left(g^{r_v}\right)^{J(v^*)} \left(g^{r_t}\right)^{E(vt^*)} \left(g^{\lambda_2}\right)^{Q(vs^*)} \left(g^{r_m}\right)^{L(vm^*)} g_2^\beta}\end{aligned}$$

$$\cdot \left(\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s}\right)^{\lambda_2} \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m}$$

$$= \frac{g_2^a \left(g_2^{F(v^*)} g^{J(v^*)}\right)^{r_v} \left(g^{E(vt^*)}\right)^{r_t} g_2^{\lambda_1} \left(g_2^{R(vu^*)-1}\right)^{\lambda_1}}{g^{r_v J(v^*)} g^{r_t E(vt^*)} g^{\lambda_2 Q(vs^*)} g^{r_m L(vm^*)}}$$

$$\cdot \left(g^{Q(vs^*)}\right)^{\lambda_2} \left(g_2^{K(vm^*)} g^{L(vm^*)}\right)^{r_m}$$

$$= g_2^a \quad (\text{since } F(v^*) = Q(vs^*) = K(vm^*) = 0)$$

$$= g^{ab}. \tag{16}$$

Thus, the challenger $\mathcal{B}$ resolves the computational Diffie-Hellman (CDH) problem.

Next, we proceed to the probability analysis for the simulation above. Note that if the simulation aborts, then the CDH problem will be unable to be computed by the challenger. In *Forge* phase, the adversary can generate a signature if the challenger can correctly response the adversary's queries in Phase 1 and do not abort in the security game. If the signature is valid, the challenger can use it to solve the CDH problem. Thus, the probability of the challenger not aborting in the security game is equal to the probability of solving the CDH problem. Hence, we list the events that the challenger $\mathcal{B}$ does not abort during the simulation process.

(1) In the phase of *initial key extract query*: if $F(v) \neq 0$, the challenger $\mathcal{B}$ can correctly answer queries without aborting.

(2) In the phase of *signing query*: if $K(vm) \neq 0$, the challenger $\mathcal{B}$ can correctly respond queries without aborting.

(3) In the phase of *forgery*: if $F(v^*) = R(vu^*) = K(vm^*) = 0$, the challenger $\mathcal{B}$ can perform the simulation without aborting.

Let q_I be the number of identities appearing in either *initial key extract queries* or *signing queries* not involving the challenge identity. In addition, let q_T be the number of identities with the period appearing in the *time key update queries* and let q_M be the number of messages in the *signing queries* involving the challenge identity. Clearly, we will have $q_I < q_E + q_S$, $q_T < q_U$ and $q_M < q_S$. To simplify the analysis, we define the events $A_i : F(v) \neq 0 \bmod l_v$ for the ith query $(1 \leq i \leq q_I)$, $A^* : F(v^*) = 0 \bmod p$, $B_k : K(vm) \neq 0 \bmod l_m$ for the kth query $(1 \leq k \leq q_M)$, $B^* : K(vm^*) = 0 \bmod p$, and $C^* : R(vu^*) = 0 \bmod p$. From the above analysis, the probability of the challenger $\mathcal{B}$ not aborting is

$$\Pr[\neg\text{abort}] \geq \Pr\left[\bigwedge_{i=1}^{q_I} A_i \wedge A^* \wedge \bigwedge_{k=1}^{q_M} B_k \wedge B^* \wedge C^*\right]$$

$$= \Pr[A^*] \Pr\left[\bigwedge_{i=1}^{q_I} A_i \mid A^*\right] \Pr[B^*] \times \Pr\left[\bigwedge_{k=1}^{q_M} B_k \mid B^*\right] \Pr[C^*]. \tag{17}$$

Note that, if $F(v) = 0 \bmod l_v$, there will be a unique choice of k_v with $0 \leq k_v \leq n_u$ such that $F(v) = 0$. Since $k_v, x', x_1, \ldots, x_{n_u}$ are chosen randomly, we obtain that

$$\begin{aligned}\Pr[A^*] &= \Pr[F(v^*) = 0] \\ &\geq \Pr[F(v^*) = 0 \wedge F(v^*) = 0 \bmod l_v] \\ &= \Pr[F(v^*) = 0 \bmod l_v] \\ &\quad \cdot \Pr[F(v^*) = 0 \mid F(v^*) = 0 \bmod l_v] \\ &= \frac{1}{l_v} \frac{1}{n_u + 1}.\end{aligned} \tag{18}$$

By similar arguments, we have

$$\begin{aligned}\Pr[B^*] &= \Pr[K(vm^*) = 0] \geq \frac{1}{l_m} \frac{1}{n_w + 1}, \\ \Pr[C^*] &= \Pr[R(vu^*) = 0] \geq \frac{1}{l_u} \frac{1}{n_\zeta + 1}.\end{aligned} \tag{19}$$

We also have that

$$\begin{aligned}\Pr\left[\bigwedge_{i=1}^{q_I} A_i \mid A^*\right] &= 1 - \Pr\left[\bigvee_{i=1}^{q_I} \neg A_i \mid A^*\right] \\ &\geq 1 - \sum_{i=1}^{q_I} \Pr[\neg A_i \mid A^*] \\ &= 1 - \frac{q_I}{l_v} \geq 1 - \frac{q_E + q_S}{l_v}\end{aligned} \tag{20}$$

and, similarly,

$$\Pr\left[\bigwedge_{k=1}^{q_M} B_k \mid B^*\right] = 1 - \frac{q_M}{l_m} \geq 1 - \frac{q_S}{l_m}. \tag{21}$$

Hence, we can obtain that

$$\begin{aligned}\Pr\left[\bigwedge_{i=1}^{q_I} A_i \wedge A^*\right] &= \Pr[A^*] \cdot \Pr\left[\bigwedge_{i=1}^{q_I} A_i \mid A^*\right] \\ &\geq \left(\frac{1}{l_v} \frac{1}{n_u + 1}\right)\left(1 - \frac{q_E + q_S}{l_v}\right),\end{aligned} \tag{22}$$

$$\Pr\left[\bigwedge_{k=1}^{q_M} B_k \wedge B^*\right] \geq \left(\frac{1}{l_m} \frac{1}{n_w + 1}\right)\left(1 - \frac{q_S}{l_m}\right).$$

We have set $l_v = 2(q_E + q_S)$, $l_u = q_K$, and $l_m = 2q_S$ in the setup at the beginning of the proof, and so the resulting probability of the challenger $\mathcal{B}$ not aborting is

$$\Pr[\neg\text{abort}] \geq \Pr\left[\bigwedge_{i=1}^{q_I} A_i \wedge A^* \wedge \bigwedge_{k=1}^{q_M} B_k \wedge B^* \wedge C^*\right]$$

$$= \Pr[A^*] \Pr\left[\bigwedge_{i=1}^{q_I} A_i \mid A^*\right] \Pr[B^*] \times \Pr\left[\bigwedge_{k=1}^{q_M} B_k \mid B^*\right] \Pr[C^*] \geq \frac{1}{4(q_E+q_S)(n_u+1)\,4q_S(n_w+1)\,q_K(n_\zeta+1)} = \frac{1}{16q_Kq_S(q_E+q_S)(n_u+1)(n_\zeta+1)(n_w+1)}. \quad (23)$$

Since the adversary $\mathscr{A}$ that has an advantage ϵ against the proposed certificateless signature scheme, the challenger $\mathscr{B}$ has an advantage

$$\epsilon' \geq \frac{\epsilon}{16q_Kq_S(q_E+q_S)(n_u+1)(n_\zeta+1)(n_w+1)} \quad (24)$$

to solve the CDH problem.

For the queries, we observe that there are $O(n_u)$ multiplications and $O(1)$ exponentiations in the *initial key extract queries*. There are $O(n_t)$ multiplications and $O(1)$ exponentiations in the *time update key queries*. There are $O(n_\zeta)$ multiplications and $O(1)$ exponentiations in both the *public key retrieve* and the *secret key extract queries*. Moreover, there are $O(n_u + n_t + n_\zeta + n_\eta + n_w)$ multiplications and O(1) exponentiations in the *signing queries*. So we have $\tau' = \tau + O((n_u \cdot q_E + n_t \cdot q_U + (n_\zeta + n_\eta) \cdot q_K + (n_u + n_t + n_\zeta + n_\eta + n_w) \cdot q_S) \cdot \tau_1 + (q_E + q_U + q_K + q_S) \cdot \tau_2)$, where τ_1 and τ_2 denote the executing time of a multiplication in $\mathbb{G}_1$ and an exponentiation in $\mathbb{G}_1$, respectively. □

Theorem 7. *Under the CDH assumption, the proposed RCLS scheme is secure against Type II adversary. Concretely, if there is a Type II adversary (KGC) that has an advantage ϵ against the proposed scheme within a running time τ, then we can construct an algorithm to solve the CDH problem with an advantage*

$$\epsilon' \geq \frac{\epsilon}{4q_Kq_S(n_w+1)} \quad (25)$$

within a running time $\tau' = \tau + O(((n_\zeta + n_\eta) \cdot q_K + (n_u + n_t + n_\zeta + n_\eta + n_w) \cdot q_S) \cdot \tau_1 + (q_K + q_S) \cdot \tau_2)$, in which q_K and q_S are the numbers of queries on secret key extract and signing, respectively, and τ_1 and τ_2 denote the executing time of a multiplication and an exponentiation in $\mathbb{G}_1$, respectively.

Proof. We assume that a Type II adversary $\mathscr{A}$ can forge a valid signature for the proposed RCLS scheme. We will construct an algorithm $\mathscr{B}$ that solves the CDH problem as follows. Without loss of generality, we assume that there exists a tuple $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$, as mentioned in Section 2, and the algorithm $\mathscr{B}$ is given $g, g^a, g^b \in \mathbb{G}_1$, where a and b are unknown to $\mathscr{B}$. In order to compute g^{ab}, the algorithm $\mathscr{B}$ simulates a challenger in the following game.

(i) *Setup*. A challenger (algorithm) $\mathscr{B}$ first sets five collision-resistant hash functions $H_u : \{0,1\}^* \to \{0,1\}^{n_u}$, $H_t : \{0,1\}^* \to \{0,1\}^{n_t}$, $H_\zeta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\zeta}$, $H_\eta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\eta}$, and $H_w : \{0,1\}^* \to \{0,1\}^{n_w}$, where $n_u, n_t, n_\zeta, n_\eta$, and n_w are fixed. Note that the employed collision-resistant hash functions are not viewed as random oracles in our security proofs. The challenger $\mathscr{B}$ then sets $l_m = 2q_S$ and chooses an integer k_m at random, where $0 \leq k_m \leq n_w$. We assume that $l_m(n_w + 1) < p$ for the given values of q_S and n_w. The challenger $\mathscr{B}$ randomly selects the integers: $z', z_1, \ldots, z_{n_u} \in \mathbb{Z}_p$, $x', x_1, \ldots, x_{n_t} \in \mathbb{Z}_p$, $a', a_1, \ldots, a_{n_\zeta} \in \mathbb{Z}_p$, $b', b_1, \ldots, b_{n_\eta} \in \mathbb{Z}_p$, $c', c_1, \ldots, c_{n_w} \in \mathbb{Z}_{l_m}$, and $d', d_1, \ldots, d_{n_w} \in \mathbb{Z}_p$.

Now, the challenger $\mathscr{B}$ constructs a set of public parameters as follows. The challenger $\mathscr{B}$ chooses two values $\alpha, \beta \in \mathbb{Z}_p$. The challenger $\mathscr{B}$ sets $g_1 = g^{\alpha+\beta}$ and $g_2 = g^b$. Furthermore, $\mathscr{B}$ computes and sends the system secret key g_2^α and time secret key g_2^β to the adversary $\mathscr{A}$. Also, $\mathscr{B}$ computes $u' = g^{z'}$ and a vector $\overrightarrow{U} = (u_i)$, where $u_i = g^{z_i}$ for $1 \leq i \leq n_u$; $t' = g^{x'}$ and a vector $\overrightarrow{T} = (t_j)$, where $t_j = g^{x_j}$ for $1 \leq j \leq n_t$; $\zeta' = g_2^{a'}$ and a vector $\overrightarrow{\zeta} = (\zeta_r)$, where $\zeta_r = g_2^{a_r}$ for $1 \leq r \leq n_\zeta$; $\eta' = g^{b'}$ and a vector $\overrightarrow{\eta} = (\eta_s)$, where $\eta_s = g^{b_s}$ for $1 \leq s \leq n_\eta$; $w' = g_2^{-l_m k_m + c'} g^{d'}$ and a vector $\overrightarrow{W} = (w_k)$, where $w_k = g_2^{c_k} g^{d_k}$ for $1 \leq k \leq n_w$. Now, the challenger $\mathscr{B}$ has constructed a set of public parameters as $Params = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_u, H_t, H_\zeta, H_\eta, H_w, g, g_1, g_2, u', \overrightarrow{U}, t', \overrightarrow{T}, \zeta', \overrightarrow{\zeta}, \eta', \overrightarrow{\eta}, w', \overrightarrow{W}\}$.

Before performing *Queries* and *Forgery* between the adversary $\mathscr{A}$ and the challenger $\mathscr{B}$, we define six functions E, F, R, Q, K, and L by

$$E(v) = z' + \sum_{i=1}^{n_u} v_i z_i; \qquad F(vt) = x' + \sum_{j=1}^{n_t} vt_j x_j;$$

$$R(vu) = a' + \sum_{r=1}^{n_\zeta} vu_r a_r; \qquad Q(vs) = b' + \sum_{s=1}^{n_\eta} vs_r b_s; \quad (26)$$

$$K(vm) = -l_m k_m + c' + \sum_{k=1}^{n_w} vm_k c_k;$$

$$L(vm) = d' + \sum_{k=1}^{n_w} vm_k d_k.$$

Here, as before, $v = H_u(\mathrm{ID}) = (v_1, v_2, \ldots, v_{n_u})$ for an identity ID, $vt = H_t(\mathrm{ID}, t) = (vt_1, vt_2, \ldots, v_{n_t})$ for an identity ID in a period t, $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2) = (vu_1, vu_2, \ldots, vu_{n_\zeta})$ and $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2) = (vs_1, vs_2, \ldots, vs_{n_\eta})$ for a public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2)$, and $vm = H_w(M) = (vm_1, vm_2, \ldots, vm_{n_w})$ for a message M.

Finally, for the cumbersome notations defined above, we conclude with four relations to which will be referred frequently in the sequel; namely,

$$u'\prod_{i=1}^{n_u} u_i^{v_i} = g^{E(v)}; \qquad t'\prod_{j=1}^{n_t} t_j^{vt_j} = g^{F(vt)};$$
$$\zeta'\prod_{r=1}^{n_\zeta} \zeta_r^{vu_r} = g_2^{R(vu)}; \qquad \eta'\prod_{s=1}^{n_\eta} \eta_s^{vs_s} = g^{Q(vs)}; \tag{27}$$
$$w'\prod_{k=1}^{n_w} w_k^{vm_k} = g_2^{K(vm)} g^{L(vm)}.$$

(ii) *Queries.* The challenger $\mathscr{B}$ maintains a list L of tuples of the form $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$. Initially the list is empty. First, without loss of generality, the challenger $\mathscr{B}$ picks a target identity ID', selects a secret value $\lambda_2 \in \mathbb{Z}_p^*$, and sets the public key $\mathrm{PK}_{\mathrm{ID}'} = (\mathrm{PK}_1, \mathrm{PK}_2) = (g^a, g^{\lambda_2})$. Then, the challenger $\mathscr{B}$ adds the tuple $\langle \mathrm{ID}', \perp, \lambda_2, \mathrm{PK}_{\mathrm{ID}'}, \perp \rangle$ in the list L. The adversary $\mathscr{A}$ may make a number of queries in an adaptive manner as follows.

(a) *Public key retrieve query (ID)*: upon receiving a query for the public key of an identity ID, the challenger $\mathscr{B}$ responds to the query as follows.

(1) If ID appears in the list L, the challenger $\mathscr{B}$ responds with the corresponding $\mathrm{PK}_{\mathrm{ID}}$.

(2) If ID does not appear in the list L, the challenger $\mathscr{B}$ selects two secret values $\lambda_1, \lambda_2 \in \mathbb{Z}_p^*$, sets the public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2) = (g^{\lambda_1}, g^{\lambda_2})$, and then computes $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2) = (vu_1, vu_2, \ldots, vu_{n_\zeta})$, $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2) = (vs_1, vs_2, \ldots, vs_{n_\eta})$ and the secret key $\mathrm{SK}_{\mathrm{ID}} = g_2^{\lambda_1}(\zeta'\prod_{r=1}^{n_\zeta}\zeta_r^{vu_r})^{\lambda_1}(\eta'\prod_{s=1}^{n_\eta}\eta_s^{vs_s})^{\lambda_2}$. The challenger $\mathscr{B}$ adds the tuple $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$ in the list L and returns $\mathrm{PK}_{\mathrm{ID}}$ as the query output.

(b) *Secret key extract query (ID)*: consider such a query along with an identity ID. If $\mathrm{ID} = \mathrm{ID}'$, the challenger $\mathscr{B}$ aborts. If $\mathrm{ID} \neq \mathrm{ID}'$ and ID appears in the list L, the challenger $\mathscr{B}$ accesses the tuple $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$ in the list L and responds with $\mathrm{SK}_{\mathrm{ID}}$. If $\mathrm{ID} \neq \mathrm{ID}'$ and ID does not appear in the list L, the challenger $\mathscr{B}$ runs the *user key generation* algorithm to add the tuple $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$ in the list L and then responds with $\mathrm{SK}_{\mathrm{ID}}$ so obtained.

(c) *Signing query (ID, t, M, PK_{ID})*: consider a query for a message M, an identity ID, a period t, and a public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2)$. The challenger $\mathscr{B}$ first sets $v = H_u(\mathrm{ID})$, $vt = H_t(\mathrm{ID}, t)$, $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2)$, and $vm = H_w(M)$. $\mathscr{B}$ then computes $E(v)$, $F(vt)$, $R(vu)$, $Q(vs)$, $K(vm)$, and $L(vm)$. If $K(vm) = 0$, the challenger $\mathscr{B}$ reports failure and terminates. Otherwise, the challenger $\mathscr{B}$ considers the following two cases.

Case 1: if $\mathrm{ID} = \mathrm{ID}'$, the challenger $\mathscr{B}$ runs the *initial key extract* and *time key update* algorithms to obtain the initial secret key $D_{\mathrm{ID}} = (D_1, D_2)$ and the time update key $T_{\mathrm{ID},t} = (T_1, T_2)$. The challenger $\mathscr{B}$ chooses a random value $r_m \in \mathbb{Z}_p^*$ and then responds with the signature

$$\begin{aligned}\sigma &= (\sigma_1, \sigma_2, \sigma_3, \sigma_4)\\ &= \Big(D_1 T_1 (\mathrm{PK}_1)^{(-L(vm)/K(vm))(1+R(vu))}\\ &\quad \cdot (\mathrm{PK}_2)^{Q(vs)} \big(g_2^{K(vm)} g^{L(vm)}\big)^{r_m},\\ &\quad D_2, T_2, (\mathrm{PK}_1)^{-(1+R(vu))/K(vm)} g^{r_m}\Big).\end{aligned} \tag{28}$$

Note that σ is indeed a valid signature since $K(vm) \neq 0$ and, by the equalities in (27),

$$\begin{aligned}\sigma_1 &= D_1 T_1 (g^a)^{(-L(vm)/K(vm))(1+R(vu))} \big(g^{\lambda_2}\big)^{Q(vs)}\\ &\quad \cdot \big(g_2^{K(vm)} g^{L(vm)}\big)^{r_m}\\ &= D_1 T_1 g_2^a \big(g_2^{R(vu)}\big)^a \big(g^{Q(vs)}\big)^{\lambda_2}\\ &\quad \cdot \big(g_2^{K(vm)} g^{L(vm)}\big)^{r_m - a(1+R(vu))/K(vm)}\\ &= D_1 T_1 g_2^a \left(\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r}\right)^a \left(\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s}\right)^{\lambda_2}\\ &\quad \cdot \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r'_m};\\ \sigma_2 &= D_2; \qquad \sigma_3 = T_2;\\ \sigma_4 &= (g^a)^{-(1+R(vu))/K(vm)} g^{r_m} = g^{r'_m},\end{aligned} \tag{29}$$

where $r'_m = r_m - a(1 + R(vu))/K(vm)$.

Case 2: if $\mathrm{ID} \neq \mathrm{ID}'$, the challenger $\mathscr{B}$ runs the *initial key extract* and *time key update* algorithms to obtain the initial secret key $D_{\mathrm{ID}} = (D_1, D_2)$ and the time update key $T_{\mathrm{ID},t} = (T_1, T_2)$. Then, $\mathscr{B}$ accesses to the list L to obtain the secret key $\mathrm{SK}_{\mathrm{ID}}$. The challenger $\mathscr{B}$ chooses a random value $r_m \in \mathbb{Z}_p^*$ and then responds with the signature

$$\begin{aligned}\sigma &= (\sigma_1, \sigma_2, \sigma_3, \sigma_4)\\ &= \left(D_1 T_1 \mathrm{SK}_{\mathrm{ID}} \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m}, D_2, T_2, g^{r_m}\right).\end{aligned} \tag{30}$$

(iii) *Forgery.* Assume that the adversary $\mathscr{A}$ generates a valid signature $\sigma^* = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ for ID^* on M^*

in t^*, where ID^*, t^*, and M^* are the target identity, period, and message, respectively. If $\mathrm{ID} \neq \mathrm{ID}'$, the challenger $\mathscr{B}$ reports failure and terminates. If $\mathrm{ID} = \mathrm{ID}'$, the challenger $\mathscr{B}$ first accesses to the list L to obtain $\mathrm{PK}_{\mathrm{ID}^*} = (\mathrm{PK}_1, \mathrm{PK}_2)$. The challenger $\mathscr{B}$ then computes $v^* = H_u(\mathrm{ID}^*)$, $vt^* = H_t(\mathrm{ID}^*, t^*)$, $vu^* = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vs^* = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vm^* = H_w(M^*)$, $E(v^*)$, $F(vt^*)$, $R(vu^*)$, $Q(vs^*)$, $L(vm^*)$, and $K(vm^*)$. If $K(vm^*) \neq 0$, the challenger $\mathscr{B}$ aborts. Otherwise, that is, when $K(vm^*) = 0$, the challenger $\mathscr{B}$, by using (27), computes $(g^{ab})^{1+R(vu^*)}$ as follows:

$$\begin{aligned}
V &= \frac{\sigma_1}{g_2^{\alpha+\beta}\left(\sigma_2^{E(v^*)}\right)\left(\sigma_3^{F(vt^*)}\right)\left(\mathrm{PK}_2^{R(vu^*)}\right)\left(\sigma_4^{L(vm^*)}\right)} \\
&= \frac{g_2^{\alpha+\beta}\left(u'\prod_{i=1}^{n_u} u_i^{v_i}\right)^{r_v}\left(t'\prod_{j=1}^{n_t} t_j^{vt_j}\right)^{r_t} g_2^{a}\left(\zeta'\prod_{r=1}^{n_\zeta}\zeta_r^{vu_r}\right)^{a}}{g_2^{\alpha+\beta}\left(g^{r_v}\right)^{E(v^*)}\left(g^{r_t}\right)^{F(vt^*)}\left(g^{\lambda_2}\right)^{Q(vs^*)}\left(g^{r_m}\right)^{L(vm^*)}} \\
&\quad \cdot \left(\eta'\prod_{s=1}^{n_\eta}\eta_s^{vs_s}\right)^{\lambda_2}\left(w'\prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m} \\
&= \frac{\left(g^{E(v^*)}\right)^{r_v}\left(g^{F(vt^*)}\right)^{r_t} g_2^{a}\left(g_2^{R(vu^*)}\right)^{a}}{g^{r_v E(v^*)} g^{r_t F(vt^*)} g^{\lambda_2 Q(vs^*)} g^{r_m L(vm^*)}} \\
&\quad \cdot \left(g^{Q(vs^*)}\right)^{\lambda_2}\left(g_2^{K(vm^*)} g^{L(vm^*)}\right)^{r_m} \\
&= \left(g_2^{a}\right)^{1+R(vu^*)} \quad (\text{since } K(vm^*) = 0) \\
&= \left(g^{ab}\right)^{1+R(vu^*)},
\end{aligned} \tag{31}$$

where α and β have been chosen in the setup at the beginning of the proof. Finally, by computing $V^{(1+R(vu^*))^{-1}}$, we obtain the value g^{ab}. Thus, the challenger $\mathscr{B}$ resolves the computational Diffie-Hellman (CDH) problem.

Next, we proceed to the probability analysis for the simulation above. For convenience, we list the events that the challenger $\mathscr{B}$ does not abort during the simulation process.

(1) In the phase of *signing query*: if $K(vm) \neq 0$, the challenger $\mathscr{B}$ can correctly respond to queries without aborting.

(2) In the phase of *forgery*: if $\mathrm{ID}^* = \mathrm{ID}'$ and $K(vm^*) = 0$, the challenger $\mathscr{B}$ can perform the simulation without aborting.

Let q_M be the number of messages in the *signing queries* involving the challenge identity. Clearly, we will have $q_M < q_S$. To simplify the analysis, we define the events $B_k : K(vm) \neq 0$ for the kth query $(1 \leq k \leq q_M)$, $B^* : K(vm^*) = 0$, $C^* : \mathrm{ID}^* = \mathrm{ID}'$. From the analysis above, the probability of the challenger $\mathscr{B}$ not aborting is

$$\begin{aligned}
\Pr[\neg\text{abort}] &\geq \Pr\left[\bigwedge_{k=1}^{q_M} B_k \wedge B^* \wedge C^*\right] \\
&= \Pr[B^*] \cdot \Pr\left[\bigwedge_{k=1}^{q_M} B_k \mid B^*\right] \cdot \Pr[C^*].
\end{aligned} \tag{32}$$

The analysis of probability above is similar to Theorem 6. Here, we can obtain that

$$\begin{aligned}
\Pr\left[\bigwedge_{k=1}^{q_M} B_k \wedge B^*\right] &= \Pr[B^*] \cdot \Pr\left[\bigwedge_{k=1}^{q_M} B_k \mid B^*\right] \\
&\geq \left(\frac{1}{l_m}\frac{1}{n_w+1}\right)\left(1 - \frac{q_S}{l_m}\right).
\end{aligned} \tag{33}$$

Since we have set $l_m = 2q_S$, the resulting probability of the challenger $\mathscr{B}$ not aborting is

$$\begin{aligned}
\Pr[\neg\text{abort}] &\geq \Pr[B^*] \cdot \Pr\left[\bigwedge_{k=1}^{q_M} B_k \mid B^*\right] \cdot \Pr[C^*] \\
&\geq \frac{1}{4q_K q_S (n_w+1)},
\end{aligned} \tag{34}$$

by noting that $\Pr[C^*] = 1/q_K$.

Since the adversary $\mathscr{A}$ has an advantage ϵ against the proposed certificateless signature scheme, the challenger $\mathscr{B}$ has an advantage

$$\epsilon' \geq \frac{\epsilon}{4q_K q_S (n_w+1)} \tag{35}$$

to solve the CDH problem.

For the queries, we observe that there are $O(n_\zeta + n_\eta)$ multiplications and $O(1)$ exponentiations in *secret key extract queries*. Moreover, there are $O(n_u + n_t + n_\zeta + n_\eta + n_w)$ multiplications and $O(1)$ exponentiations in the *signing queries*. So we have $\tau' = \tau + O(((n_\zeta + n_\eta)\cdot q_K + (n_u + n_t + n_\zeta + n_\eta + n_w)\cdot q_S)\tau_1 + (q_K + q_S)\tau_2)$, where τ_1 and τ_2 denote the executing time of a multiplication and an exponentiation in $\mathbb{G}_1$, respectively. □

Theorem 8. *Under the CDH assumption, the proposed RCLS scheme is secure against Type III adversary (revoked user). Concretely, if there is a revoked user that has an advantage ϵ against the proposed scheme within a running time τ, then we can construct an algorithm to solve the CDH problem with an advantage*

$$\epsilon' \geq \frac{\epsilon}{16q_K q_S (q_U + q_S)(n_t+1)(n_\zeta+1)(n_w+1)} \tag{36}$$

within a running time $\tau' = \tau + O((n_u\cdot q_E + n_t\cdot q_U + (n_\zeta + n_\eta)\cdot q_K + (n_u + n_t + n_\zeta + n_\eta + n_w)\cdot q_S)\tau_1 + (q_E + q_U + q_S + q_K)\tau_2)$, in which q_E, q_U, and q_S are the numbers of queries on initial key extract, time key update, and signing, respectively, q_K is the sum of the numbers of queries on public key replace and secret key extract, and τ_1 and τ_2 denote the executing time of a multiplication and an exponentiation in $\mathbb{G}_1$, respectively.

Proof. Assume that a revoked user $\mathscr{A}$ can forge a valid signature for the proposed RCLS scheme. We will construct an algorithm $\mathscr{B}$ that solves the CDH problem as follows. Without loss of generality, we assume that there exists a tuple $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle$, as mentioned in Section 2, and the algorithm $\mathscr{B}$ is given $g, g^a, g^b \in \mathbb{G}_1$, where a and b are unknown to $\mathscr{B}$. In order to compute g^{ab}, the algorithm $\mathscr{B}$ simulates a challenger in the following game.

(i) *Setup*. A challenger (algorithm) $\mathscr{B}$ first sets five collision-resistant hash functions $H_u : \{0,1\}^* \to \{0,1\}^{n_u}$, $H_t : \{0,1\}^* \to \{0,1\}^{n_t}$, $H_\zeta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\zeta}$, $H_\eta : \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^{n_\eta}$, and $H_w : \{0,1\}^* \to \{0,1\}^{n_w}$, where $n_u, n_t, n_\zeta, n_\eta$, and n_w are fixed. Note that the employed collision-resistant hash functions are not viewed as random oracles in our security proofs. The challenger $\mathscr{B}$ then sets $l_t = 2(q_U + q_S)$, $l_u = q_K$ and $l_m = 2q_S$, and chooses three integers k_t, k_u, and k_m at random, where $0 \le k_t \le n_t$, $0 \le k_u \le n_\zeta$, and $0 \le k_m \le n_w$. We assume that $l_t(n_t+1) < p, l_u(n_\zeta+1) < p$, and $l_m(n_w + 1) < p$ for the given values of q_U, q_K, q_S, n_t, n_ζ, and n_w. The challenger $\mathscr{B}$ randomly selects the integers: $z', z_1, \ldots, z_{n_u} \in \mathbb{Z}_p$, $x', x_1, \ldots, x_{n_t} \in \mathbb{Z}_{l_t}$, $y', y_1, \ldots, y_{n_t} \in \mathbb{Z}_p$, $a', a_1, \ldots, a_{n_\zeta} \in \mathbb{Z}_{l_u}$, $b', b_1, \ldots, b_{n_\eta} \in \mathbb{Z}_p$, $c', c_1, \ldots, c_{n_w} \in \mathbb{Z}_{l_m}$, and $d', d_1, \ldots, d_{n_w} \in \mathbb{Z}_p$.

Now, the challenger $\mathscr{B}$ constructs a set of public parameters as follows. The challenger $\mathscr{B}$ chooses a value $\alpha \in \mathbb{Z}_p$ as the system secret key. The challenger $\mathscr{B}$ sets $g_1 = g^\alpha g^a$ and $g_2 = g^b$. Furthermore, $\mathscr{B}$ computes $u' = g^{z'}$ and a vector $\overrightarrow{U} = (u_i)$, where $u_i = g^{z_i}$ for $1 \le i \le n_u$; $t' = g_2^{-l_t k_t + x'} g^{y'}$ and a vector $\overrightarrow{T} = (t_j)$, where $t_j = g_2^{x_i} g^{y_i}$ for $1 \le j \le n_t$; $\zeta' = g_2^{-1-l_u k_u + a'}$ and a vector $\overrightarrow{\zeta} = (\zeta_r)$, where $\zeta_r = g_2^{a_r}$ for $1 \le r \le n_\zeta$; $\eta' = g^{b'}$ and a vector $\overrightarrow{\eta} = (\eta_s)$, where $\eta_s = g^{b_s}$ for $1 \le s \le n_\eta$; $w' = g_2^{-l_m k_m + c'} g^{d'}$ and a vector $\overrightarrow{W} = (w_k)$, where $w_k = g_2^{c_k} g^{d_k}$ for $1 \le k \le n_w$. Now, the challenger $\mathscr{B}$ has constructed a set of public parameters as $Params = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_u, H_t, H_\zeta, H_\eta, H_w, g, g_1, g_2, u', \overrightarrow{U}, t', \overrightarrow{T}, \zeta', \overrightarrow{\zeta}, \eta', \overrightarrow{\eta}, w', \overrightarrow{W}\}$.

Before performing *Queries* and *Forgery* between the adversary $\mathscr{A}$ and the challenger $\mathscr{B}$, we define seven functions E, J, Q, L, F, R, and K by

$$E(v) = z' + \sum_{i=1}^{n_u} v_i z_i; \qquad J(vt) = y' + \sum_{j=1}^{n_t} vt_j y_j.$$

$$Q(vs) = b' + \sum_{s=1}^{n_\eta} vs_r b_s; \qquad L(vm) = d' + \sum_{k=1}^{n_w} vm_k d_k;$$

$$F(vt) = -l_t k_t + x' + \sum_{j=1}^{n_t} vt_j x_j; \tag{37}$$

$$R(vu) = -l_u k_u + a' + \sum_{r=1}^{n_\zeta} vu_r a_r;$$

$$K(vm) = -l_m k_m + c' + \sum_{k=1}^{n_w} vm_k c_k.$$

Here, as before, $v = H_u(\mathrm{ID}) = (v_1, v_2, \ldots, v_{n_u})$ for an identity ID, $vt = H_t(\mathrm{ID}, t) = (vt_1, vt_2, \ldots, v_{n_t})$ for an identity ID in a period t, $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2) = (vu_1, vu_2, \ldots, vu_{n_\zeta})$ and $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2) = (vs_1, vs_2, \ldots, vs_{n_\eta})$ for a public key $\mathrm{PK}_{\mathrm{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2)$, and $vm = H_w(M) = (vm_1, vm_2, \ldots, vm_{n_w})$ for a message M.

Finally, for the cumbersome notations defined above, we conclude with four relations to which will be referred frequently in the sequel; namely,

$$u' \prod_{i=1}^{n_u} u_i^{v_i} = g^{E(v)}; \qquad t' \prod_{j=1}^{n_t} t_j^{vt_j} = g_2^{F(vt)} g^{J(vt)};$$

$$\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r} = g_2^{R(vu)-1}; \qquad \eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s} = g^{Q(vs)}; \tag{38}$$

$$w' \prod_{k=1}^{n_w} w_k^{vm_k} = g_2^{K(vm)} g^{L(vm)}.$$

(ii) *Queries*. The challenger $\mathscr{B}$ maintains a list L of tuples of the form $\langle \mathrm{ID}, \lambda_1, \lambda_2, \mathrm{PK}_{\mathrm{ID}}, \mathrm{SK}_{\mathrm{ID}} \rangle$. Initially the list is empty. The adversary $\mathscr{A}$ may make a number of queries in an adaptive manner as follows.

(a) *Public key retrieve query (ID)*: to respond to the queries, the challenger $\mathscr{B}$ responds to the queries as in Theorem 6.

(b) *Public key replace query (ID, PK'_{ID})*: to respond to the queries, the challenger $\mathscr{B}$ responds to the queries as in Theorem 6.

(c) *Initial key extract query (ID)*: upon receiving a query for the initial secret key of an identity ID, the challenger $\mathscr{B}$ first sets $v = H_u(\mathrm{ID})$ and then computes $E(v)$. The challenger $\mathscr{B}$ chooses a random $r_v \in \mathbb{Z}_p$ and uses the system secret key α to compute the initial secet key as follows:

$$D_{\mathrm{ID}} = (D_1, D_2) = \left(g_2^\alpha \left(g^{E(v)}\right)^{r_v}, g^{r_v}\right) = \left(g_2^\alpha \left(u' \prod_{i=1}^{n_u} u_i^{v_i}\right)^{r_v}, g^{r_v}\right). \tag{39}$$

(d) *Time key update query (ID, t)*: upon receiving a query for the time update key of an identity ID in a period t, the challenger $\mathscr{B}$ first sets $vt = H_t(\mathrm{ID}, t)$, and then computes $F(vt)$ and $J(vt)$. If $F(vt) = 0$, the challenger $\mathscr{B}$ aborts. Otherwise, the challenger $\mathscr{B}$ chooses a random value $r_t \in \mathbb{Z}_p$ and responds with the time update key $T_{\mathrm{ID},t}$ generated by

$$T_{\mathrm{ID},t} = (T_1, T_2) = \left(\left(g^a\right)^{-J(vt)/F(vt)} \left(g_2^{F(vt)} g^{J(vt)}\right)^{r_t}, \left(g^a\right)^{-1/F(vt)} g^{r_t}\right). \tag{40}$$

Here, $T_{\mathrm{ID},t} = (T_1, T_2)$ is indeed a valid time update key since, by the first equality in (38),

$$\begin{aligned} T_1 &= (g^a)^{-J(vt)/F(vt)} \left(g_2^{F(vt)} g^{J(vt)}\right)^{a/F(vt)} \\ &\quad \cdot \left(g_2^{F(vt)} g^{J(vt)}\right)^{r_t - a/F(vt)} \\ &= (g^a)^{-J(vt)/F(vt)} \left(g_2^{F(vt)} g^{J(vt)}\right)^{a/F(vt)} \\ &\quad \cdot \left(t' \prod_{j=1}^{n_t} t_j^{vt_j}\right)^{r_t - a/F(vt)} \\ &= g_2^a \left(t' \prod_{j=1}^{n_t} t_j^{vt_j}\right)^{r_t'}; \end{aligned} \tag{41}$$

$$D_2 = (g^a)^{-1/F(vt)} g^{r_t} = g^{r_t - a/F(vt)} = g^{r_t'},$$

where $r_t' = r_t - a/F(vt)$.

(e) *Secret key extract query (ID)*: to respond to the queries, the challenger $\mathscr{B}$ responds to the queries as in Theorem 6.

(f) *Signing query* (ID, t, M, PK_{ID}): consider a query for a message M, an identity ID, a period t, and a public key $\mathrm{PK_{ID}} = (\mathrm{PK}_1, \mathrm{PK}_2)$. The challenger $\mathscr{B}$ first sets $v = H_u(\mathrm{ID})$, $vt = H_t(\mathrm{ID}, t)$, $vu = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vs = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2)$, and $vm = H_w(M)$. $\mathscr{B}$ then computes $E(v)$, $F(vt)$, $J(vt)$, $R(vu)$, $Q(vs)$, $K(vm)$, and $L(vm)$. If $K(vm) = 0$, the challenger $\mathscr{B}$ reports failure and terminates. Otherwise, the challenger $\mathscr{B}$ considers the following two cases.

Case 1: assume that the identity ID has previously appeared in the *public key replace query*. If $F(vt) \neq 0$, the challenger $\mathscr{B}$ can compute the time update key $T_{\mathrm{ID},t} = (T_1, T_2)$ as in the *time key update query*. In addition, the challenger $\mathscr{B}$ computes the initial secret key $D_{\mathrm{ID}} = (D_1, D_2)$ as in the *initial key extract query*. The challenger $\mathscr{B}$ then chooses a random value $r_m \in \mathbb{Z}_p^*$ and responds with the signature

$$\begin{aligned} \sigma &= (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \\ &= \Big(D_1 T_1 (\mathrm{PK}_1)^{(-L(vm)/K(vm))R(vu)} \\ &\quad \cdot (\mathrm{PK}_2)^{Q(vs)} \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m}, \\ &\quad D_2, T_2, (\mathrm{PK}_1)^{-R(vu)/K(vm)} g^{r_m}\Big). \end{aligned} \tag{42}$$

Note that σ is indeed a valid signature since $K(vm) \neq 0$ and, by the equalities in (38),

$$\begin{aligned} \sigma_1 &= D_1 T_1 \left(g^{\lambda_1}\right)^{(-L(vm)/K(vm))R(vu)} \\ &\quad \cdot \left(g^{\lambda_2}\right)^{Q(vs)} \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m} \\ &= D_1 T_1 g_2^{\lambda_1} \left(g_2^{R(vu)-1}\right)^{\lambda_1} \left(g^{Q(vs)}\right)^{\lambda_2} \\ &\quad \cdot \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m - \lambda_1 R(vu)/K(vm)} \\ &= D_1 T_1 g_2^{\lambda_1} \left(\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r}\right)^{\lambda_1} \left(\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s}\right)^{\lambda_2} \\ &\quad \cdot \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m'}; \end{aligned}$$

$$\sigma_2 = D_2; \qquad \sigma_3 = T_2;$$

$$\sigma_4 = g^{-\lambda_1 R(vu)/K(vm)} g^{r_m} = g^{r_m'}, \tag{43}$$

where $r_m' = r_m - \lambda_1 R(vu)/K(vm)$.

On the other hand, if $F(vt) = 0$, the challenger $\mathscr{B}$ first computes the initial secret key $D_{\mathrm{ID}} = (D_1, D_2)$ as in the *initial key extract query*. Then, the challenge $\mathscr{B}$ chooses two random values $r_t, r_m \in \mathbb{Z}_p^*$ and responds with the signature

$$\begin{aligned} \sigma &= (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \\ &= \Big(D_1 (g^a)^{-L(vm)/K(vm)} \left(g_2^{F(vt)} g^{J(vt)}\right)^{r_t} \\ &\quad \times (\mathrm{PK}_1)^{(-L(vm)/K(vm))R(vu)} \\ &\quad \cdot (\mathrm{PK}_2)^{Q(vs)} \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m}, \\ &\quad D_2, g^{r_t}, g_1^{-1/K(vm)} (\mathrm{PK}_1)^{-R(vu)/K(vm)} g^{r_m}\Big). \end{aligned} \tag{44}$$

Note that σ is also a valid signature since, by (38),

$$\begin{aligned} \sigma_1 &= D_1 (g^a)^{-L(vm)/K(vm)} \left(g_2^{F(vt)} g^{J(vt)}\right)^{r_t} \\ &\quad \times \left(g^{\lambda_1}\right)^{(-L(vm)/K(vm))R(vu)} \\ &\quad \cdot \left(g^{\lambda_2}\right)^{Q(vs)} \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m} \\ &= D_1 g_2^a \left(g_2^{F(vt)} g^{J(vt)}\right)^{r_t} g_2^{\lambda_1} \left(g_2^{R(vu)-1}\right)^{\lambda_1} \left(g^{Q(vs)}\right)^{\lambda_2} \\ &\quad \cdot \left(g_2^{K(vm)} g^{L(vm)}\right)^{r_m - (a + \lambda_1 R(vu))/K(vm)} \\ &= D_1 g_2^a \left(t' \prod_{j=1}^{n_t} t_j^{vt_j}\right)^{r_t} g_2^{\lambda_1} \left(\zeta' \prod_{r=1}^{n_\zeta} \zeta_r^{vu_r}\right)^{\lambda_1} \\ &\quad \cdot \left(\eta' \prod_{s=1}^{n_\eta} \eta_s^{vs_s}\right)^{\lambda_2} \left(w' \prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m'}; \end{aligned}$$

$$\sigma_2 = D_2; \qquad \sigma_3 = g^{r_t};$$

$$\sigma_4 = \left(g^a\right)^{-1/K(vm)}\left(g^{\lambda_1}\right)^{-R(vu)/K(vm)} g^{r_m} = g^{r_m-(a+\lambda_1 R(vu))/K(vm)} = g^{r'_m}, \tag{45}$$

where $r'_m = r_m - (a + \lambda_1 R(vu))/K(vm)$.

Case 2: assume that the identity ID has not previously appeared in the *public key replace query*. The challenger $\mathcal{B}$ computes the initial secret key $D_{\mathrm{ID}} = (D_1, D_2)$ as in the *initial key extract query*. If $F(vt) \neq 0$, the challenger $\mathcal{B}$ can compute the time update key $T_{\mathrm{ID},t} = (T_1, T_2)$ as in the *initial key extract query* and accesses the list L to obtain the corresponding secret key $\mathrm{SK}_{\mathrm{ID}}$. The challenger $\mathcal{B}$ chooses a random value $r_m \in \mathbb{Z}_p^*$ and then responds with the signature

$$\sigma = \left(D_1 T_1 \mathrm{SK}_{\mathrm{ID}}\left(w'\prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m}, D_2, T_2, g^{r_m}\right). \tag{46}$$

If $F(vt) = 0$, then $\mathcal{B}$ chooses two random values $r_t, r_m \in \mathbb{Z}_p^*$ and responds with the signature

$$\sigma = \left(D_1\left(g^a\right)^{-L(vm)/K(vm)}\left(t'\prod_{j=1}^{n_t} t_j^{vt_j}\right)^{r_t} \mathrm{SK}_{\mathrm{ID}} \cdot \left(w'\prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m}, D_2, g^{r_t}, g_1^{-1/K(vm)} g^{r_m}\right). \tag{47}$$

(iii) *Forgery.* Assume that the adversary $\mathcal{A}$ generates a valid signature $\sigma^* = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ for ID^* on M^* in t^*, where ID^*, t^*, and M^* are the target identity, period, and message, respectively. The challenger $\mathcal{B}$ first accesses the list L to obtain $\mathrm{PK}_{\mathrm{ID}^*} = (\mathrm{PK}_1, \mathrm{PK}_2)$. The challenger $\mathcal{B}$ then computes $v^* = H_u(\mathrm{ID}^*)$, $vt^* = H_t(\mathrm{ID}^*, t^*)$, $vu^* = H_\zeta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vs^* = H_\eta(\mathrm{PK}_1, \mathrm{PK}_2)$, $vm^* = H_w(M^*)$, $E(v^*)$, $F(vt^*)$, $J(vt^*)$, $R(vu^*)$, $Q(vs^*)$, $L(vm^*)$, and $K(vm^*)$. If $F(vt^*) \neq 0$, $R(vu^*) \neq 0$, or $K(vm^*) \neq 0$, the challenger $\mathcal{B}$ aborts. Otherwise, that is, when $F(vt^*) = R(vu^*) = K(vm^*) = 0$, the challenger $\mathcal{B}$, by using (38), computes g^{ab} as follows:

$$\begin{aligned}
&\frac{\sigma_1}{\left(\sigma_2^{E(v^*)}\right)\left(\sigma_3^{J(vt^*)}\right)\left(\mathrm{PK}_2^{Q(vs^*)}\right)\left(\sigma_4^{L(vm^*)}\right) g_2^{\alpha}} \\
&= \frac{g_2^{\alpha}\left(u'\prod_{i=1}^{n_u} u_i^{v_i}\right)^{r_v} g_2^{a}\left(t'\prod_{j=1}^{n_t} t_j^{vt_j}\right)^{r_t} g_2^{\lambda_1}\left(\zeta'\prod_{r=1}^{n_\zeta}\zeta_r^{vu_r}\right)^{\lambda_1}}{\left(g^{r_v}\right)^{E(v^*)}\left(g^{r_t}\right)^{J(vt^*)}\left(g^{\lambda_2}\right)^{Q(vs^*)}\left(g^{r_m}\right)^{L(vm^*)} g_2^{\alpha}} \\
&\quad \cdot \left(\eta'\prod_{s=1}^{n_\eta}\eta_s^{vs_s}\right)^{\lambda_2}\left(w'\prod_{k=1}^{n_w} w_k^{vm_k}\right)^{r_m} \\
&= \frac{\left(g_2^{E(v^*)}\right)^{r_v} g_2^{a}\left(g^{F(vt^*)} g^{J(vt^*)}\right)^{r_t} g_2^{\lambda_1}\left(g_2^{R(vu^*)-1}\right)^{\lambda_1}}{g^{r_v E(v^*)} g^{r_t J(vt^*)} g^{\lambda_2 Q(vs^*)} g^{r_m L(vm^*)}} \\
&\quad \cdot \left(g^{Q(vs^*)}\right)^{\lambda_2}\left(g_2^{K(vm^*)} g^{L(vm^*)}\right)^{r_m} \\
&= g_2^{a} \quad \left(\text{since } F\left(vt^*\right) = Q\left(vs^*\right) = K\left(vm^*\right) = 0\right) \\
&= g^{ab}.
\end{aligned} \tag{48}$$

Thus, the challenger $\mathcal{B}$ resolves the computational Diffie-Hellman (CDH) problem.

The analysis is similar to Theorem 6. The probability of the challenger $\mathcal{B}$ not aborting is $\Pr[\neg\mathrm{abort}] \geq 1/16 q_K q_S (q_E + q_S)(n_u + 1)(n_\zeta + 1)(n_w + 1)$. Then, the successful probability (advantage) of the challenger $\mathcal{B}$ who can solve the CDH problem is at least $\varepsilon/16 q_K q_S (q_E + q_S)(n_u + 1)(n_\zeta + 1)(n_w + 1)$. The executing time is $\tau + O((n_u \cdot q_E + n_t \cdot q_U + (n_\zeta + n_\eta)\cdot q_K + (n_u + n_t + n_\zeta + n_\eta + n_w)\cdot q_S)\cdot \tau_1 + (q_E + q_U + q_K + q_S)\cdot \tau_2)$, where τ_1 and τ_2 denote the executing time of a multiplication in $\mathbb{G}_1$ and an exponentiation in $\mathbb{G}_1$, respectively. □

6. Discussions and Comparisons

Here, we compare our RCLS scheme with the previously proposed CLS schemes in the standard model [16–18, 21]. Table 1 presents the comparisons between those schemes and ours in terms of computational cost and security property.

In the signature procedure, our scheme requires only $2T_{\mathrm{exp}}$ operations and the other schemes require at least $6T_{\mathrm{exp}}$ operations, where T_{exp} is the time for executing an exponentiation operation in $\mathbb{G}_1$. In the verification procedure, our scheme requires $7TG_e$ operations, where TG_e denotes the time for executing a pairing operation $\hat{e}$. Our scheme requires more pairing operations than the others but it provides the security against both Type I and Type II adversaries. As for the security analysis, as mentioned in Section 1, all the existing CLS schemes in the standard model suffer from outsiders' key replacement attacks (Type I adversary) or KGC attacks (Type II adversary). We emphasize that our scheme provides a full security against both Type I and Type II adversaries.

In addition, our scheme provides a public revocation mechanism, while the existing CLS schemes did not. As for security assumptions, our scheme is based on the standard CDH assumption, while the others are based on nonstandard assumptions, such as the non-pairing-based generalized bilinear Diffie-Hellman (NGBDH), many Diffie-Hellman (Many-DH), augmented computational Diffie-Hellman (AC-DH), and 2-many Difffie-Hellman (2-Many-DH) assumptions.

Furthermore, to reduce the KGC's computational cost, one may adopt the technique in [37] by employing a delegated revocation authority (DRA) to generate users' time update keys. In Figure 1, we illustrate how the DRA assists the KGC in revoking misbehaving/compromised users. First, the KGC generates the public parameters *params*, the system secret key g_2^{α} and the time secret key g_2^{β}. Secondly, the KGC transmits the time secret key g_2^{β} to the DRA by using a secure channel, and then the DRA can use it to generate users' time update keys and send them to users via a public channel. Finally, in order to revoke some misbehaving/compromised users, it only stops issuing the current time update keys to those users.

Table 1: Comparisons between our scheme and the previously proposed schemes.

	Liu et al.'s scheme [16]	Xiong et al.'s scheme [17]	Yuan et al.'s scheme [18]	Yu et al.'s scheme [21]	Our scheme
Computational cost for signature	$6T_{\text{exp}}$	$6T_{\text{exp}}$	$9T_{\text{exp}}$	$7T_{\text{exp}}$	$2T_{\text{exp}}$
Computational cost for verification	$6TG_e$	$3TG_e$	$6TG_e$	$5TG_e$	$7TG_e$
Against Type I adversary	Yes	No	No	No	Yes
Against Type II adversary	No	Yes	Yes	No	Yes
Revocation mechanism	No	No	No	No	Yes
Against Type III adversary	—	—	—	—	Yes
Security assumption	NGBDH Many-DH	NGBDH Many-DH	AC-DH 2-Many-DH	NGBDH Many-DH	CDH

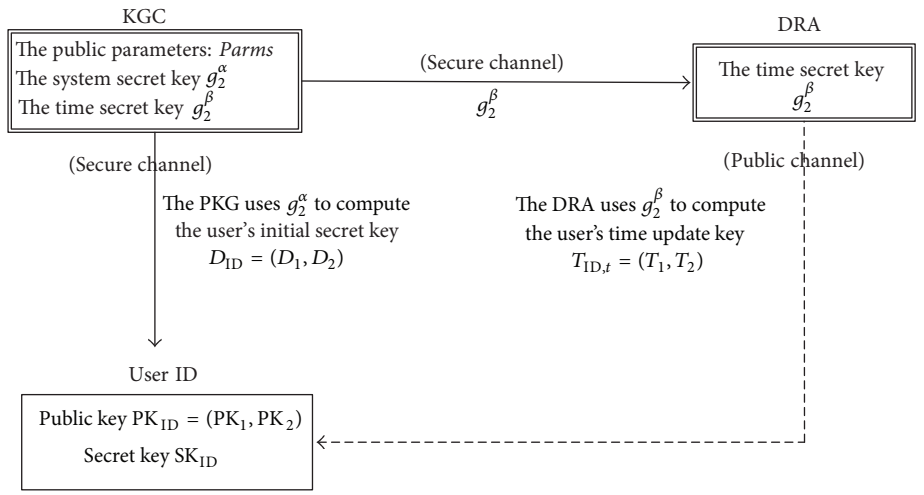

Figure 1: The generation of a user's full signing key.

7. Conclusions

In this article, we proposed the first secure revocable certificateless signature scheme in the standard model under an extended model of Hu et al.'s. We formally demonstrated that our scheme possesses existential unforgeability against adaptive chosen-message attacks from Type I, Type II, and Type III adversaries under the standard computational Diffie-Hellman assumption. Moreover, Table 1 indicates that our scheme owns better security than the others under consideration. Finally, an interesting and nontrivial issue, namely, constructing a strongly unforgeable certificateless signature scheme, is worth studying. We leave it as future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank anonymous referees for their valuable comments and constructive suggestions. This research was partially supported by the Ministry of Science and Technology, Taiwan, under contract no. MOST103-2221-E-018-022-MY2.

References

[1] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of the 21st Annual International Cryptology Conference (Crypto '01)*, pp. 213–229, Santa Barbara, Calif, USA, August 2001.

[2] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology: Proceedings of CRYPTO '84*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Berlin, Germany, 1985.

[3] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Public Key Cryptography—PKC 2003*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 18–30, 2003.

[4] B. Waters, "Efficient identity-based encryption without random oracles," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt '05)*, pp. 1–33, Aarhus, Denmark, May 2005.

[5] Y.-M. Tseng, T.-Y. Wu, and J.-D. Wu, "An efficient and provably secure id-based signature scheme with batch verifications," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 11, pp. 3911–3922, 2009.

[6] Y. Ren, D. Gu, S. Wang, and X. Zhang, "New fuzzy identity-based encryption in the standard model," *Informatica*, vol. 21, no. 3, pp. 393–407, 2010.

[7] J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Transactions on Computers*, 2013.

[8] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Advances in Cryptology—EUROCRYPT '03*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 272–293, 2003.

[9] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT 2003*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, Berlin, Germany, 2003.

[10] S. S. Al-Riyami and K. G. Paterson, "CBE from CL-PKE: a generic construction and efficient schemes," in *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC '05)*, pp. 398–415, Les Diablerets, Switzerland, January 2005.

[11] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from Asiacrypt 2003," in *Proceedings of the Cryptology and Network Security (CANS '05)*, pp. 13–25, 2005.

[12] B. Hu, D. Wong, Z. Zhang, and X. Deng, "Key replacement attack against a generic construction of certificateless signature," in *Information Security and Privacy*, vol. 4058 of *Lecture Notes in Computer Science*, pp. 235–246, Springer, Berlin, Germany, 2006.

[13] B. Libert and J. J. Quisquater, "On constructing certificateless cryptosystems from identity based encryption," in *Public Key Cryptography—PKC '06*, vol. 3958 of *Lecture Notes in Computer Science*, pp. 474–490, 2006.

[14] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signature revisited," in *Proceedings of the 12th Australasian Conference (ACISP '07)*, pp. 308–322, Townsville, Australia, July 2007.

[15] Y. H. Hwang, J. K. Liu, and S. S. Chow, "Certificateless public key encryption secure against malicious KGC attacks in the standard model," *Journal of Universal Computer Science*, vol. 14, no. 3, pp. 463–480, 2008.

[16] J. K. Liu, M. H. Au, and W. Susilo, "Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS '07)*, pp. 273–283, March 2007.

[17] H. Xiong, Z. Qin, and F. Li, "An improved certificateless signature scheme secure in the standard model," *Fundamenta Informaticae*, vol. 88, no. 1-2, pp. 193–206, 2008.

[18] Y. Yuan, D. Li, L. Tian, and H. Zhu, "Certificateless signature scheme without random oracles," in *Proceedings of the Advances in Information Security and Assurance (ISA '09)*, pp. 31–40, 2009.

[19] D. Fiore, R. Gennaro, and N. P. Smart, "Constructing certificateless encryption and ID-based encryption from ID-based key agreement," in *Pairing-Based Cryptography—Pairing '10*, vol. 6487 of *Lecture Notes in Computer Science*, pp. 167–186, 2010.

[20] G. Yang and C. H. Tan, "Strongly secure certificateless key exchange without pairing," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS '11)*, pp. 71–79, March 2011.

[21] Y. Yu, Y. Mu, G. Wang, Q. Xia, and B. Yang, "Improved certificateless signature scheme provably secure in the standard model," *IET Information Security*, vol. 6, no. 2, pp. 102–110, 2012.

[22] T.-T. Tsai and Y.-M. Tseng, "Revocable certificateless public key encryption," *IEEE Systems Journal*, 2013.

[23] D. Yum and P. Lee, "Generic construction of certificateless encryption," in *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA '04)*, pp. 802–811, 2004.

[24] M. C. Gorantla and A. Saxena, "An efficient certificateless signature scheme," in *Proceedings of the International Conference (CIS '05)*, pp. 110–116, Xi'an, China, December 2005.

[25] X. Cao, K. G. Paterson, and W. Kou, "An attack on a certificateless signature scheme," Tech. Rep. 2006/367, Cryptology ePrint Archive, 2006, http://eprint.iacr.org/2006/367.

[26] J. Zhang and J. Mao, "Security analysis of two signature schemes and their improved schemes," in *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA '07)*, pp. 589–602, 2007.

[27] L. Zhang and F. Zhang, "A new provably secure certificateless signature scheme," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 1685–1689, May 2008.

[28] K.-A. Shim, "Breaking the short certificateless signature scheme," *Information Sciences*, vol. 179, no. 3, pp. 303–306, 2009.

[29] Y. C. Chen, R. Tso, W. Susilo, X. Huang, and G. Horng, "Certificateless signatures: structural extensions of security models and new provably secure schemes," Tech. Rep. 2013/193, Cryptology ePrint Archive, 2013, http://eprint.iacr.org/2013/193.

[30] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS '93)*, pp. 62–73, November 1993.

[31] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Information Security and Privacy*, vol. 4058 of *Lecture Notes in Computer Science*, pp. 207–222, Springer, Berlin, Germany, 2006.

[32] Q. Xia, C. Xu, and Y. Yu, "Key replacement attack on two certificateless signature schemes without random oracles," *Key Engineering Materials*, vol. 439-440, pp. 1606–1611, 2010.

[33] L. Cheng, Q. Wen, Z. P. Jin, and H. Zhang, "On the security of a certificateless signature scheme in the standard model," Cryptology ePrint Archive Report 2013/153, 2013, https://eprint.iacr.org/2013/153.

[34] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," RFC 3280, IETF, 2002.

[35] L. Shen, F. Zhang, and Y. Sun, "Efficient revocable certificateless encryption secure in the standard model," *The Computer Journal*, vol. 57, no. 4, pp. 592–601, 2014.

[36] Y.-M. Tseng and T.-T. Tsai, "Efficient revocable ID-based encryption with a public channel," *Computer Journal*, vol. 55, no. 4, pp. 475–486, 2012.

[37] T.-T. Tsai, Y.-M. Tseng, and T.-Y. Wu, "RHIBE: constructing revocable hierarchical ID-based encryption from HIBE," *Informatica*, vol. 25, no. 2, pp. 299–326, 2014.