Certificateless Cryptography with KGC Trust Level 3 Revisited

Fei Li[1,2,3], Wei Gao[1,2,3](✉), Dongqing Xie[1], and Chunming Tang[1]

[1] School of Mathematics, Guangzhou University, Guangzhou 510000, China
miss_lifei@163.com, mygaowei@163.com, dongqing_xie@hotmail.com,
ctang@gzhu.edu.cn
[2] Nanjing University of Information Science and Technology, Nanjing 210044, China
[3] School of Mathematics and Statistics, Ludong University, Yantai 264025, China

Abstract. This paper revisits the issue of obtaining KGC (Key Generator Center) trust level 3 in certificateless cryptography. The AP (Al-Riyami-Paterson) binding technique can modularly construct the certificateless encryption/signature scheme with trust level 3 from that with trust level 2. However, its security proof has been an open problem. Yang and Tan improved the AP framework by adding extra cryptographic tools: random oracles for security proof in the random oracle model or trapdoor hash functions for security proof in the standard model. This paper aims to prove secure the original AP binding technique. The basic technique for achieving this security proof depends on the improved security model for certificateless encryption (or signature) schemes. As an application example, one key dependent certificateless encryption scheme with both authority trust level 3 and provable security in the standard model is modularly constructed by applying the AP binding framework to one conventional certificateless encryption scheme.

Keywords: Certificateless cryptography · Provable security · Trust level

1 Introduction

Certificateless cryptography (CLC) proposed by Al-Riyami and Paterson [1] lies between identity based cryptography (IBC) and conventional public key cryptography (PKC) based on PKI. It tries to solve both the key escrow issue in IBC and the certificate management issue in PKC. In CLC, the entity's full private key fsk is computed from the partial private key ppk generated by Key Generation Centre (KGC) and the user self-generated secret key usk (also known as secret value) which corresponds to the user's public key upk. The ciphertext is generated based on the identity and the entity public key. There exists no certificate to authenticate the entity public key in CL-PKC. As a result, an adversary in CLC can use any public key of its choice to replace the original public key and certificateless cryptosystems should be secure against such an attack. Attackers for CLC are usually divided into two types in the security model [1].

X. Sun et al. (Eds.): ICCCS 2017, Part II, LNCS 10603, pp. 292–304, 2017.
https://doi.org/10.1007/978-3-319-68542-7_24

The attacker of Type I is used to capture an ordinary adversary that can replace the user public keys. The attacker of Type II is used to capture an honest-but-curious KGC that knows the master secret key but never replaces public keys. In fact, the KGC can impersonate any user once if it can arbitrarily change the public key. Since the notion of certificateless cryptography was proposed, there are some important improvements in the security models. In [10], to prevent the Denial-of-Decryption attacks, the concept of self-generated-certificate public key cryptography was proposed. The attack model called the malicious-but-passive KGC in certificateless cryptosystems was developed by Au et al. [2], and further studied in [8,15]. In [7], Hu et al. formalized the KGC trust level 3 security. In [14], for reaching KGC trust level 3, the notion of the key dependent certificateless encryption/signature schemes was developed and the generic construction for key dependent CLE and CLS schemes was presented.

Next, we focus on the issue of KGC trust level 3. The trust hierarchy for public key cryptography is divided into 3 levels by Girault [6]. (1) For level 1, as in identity-based cryptography [3], the trusted authority knows the private key of any user; (2) For level 2, as in conventional certificateless cryptography, the authority cannot figure out the secret key, but can first generate false guarantees and then impersonate the user. (3) For level 3, as in public key cryptography based on PKI, the authority cannot figure out the secret key, and it has to face the proof for generating false guarantees of the user once if it does such fraud. To address the KGC trust level problem in CLC, Yang and Tan [14] developed the concept called Key Dependent Certificateless encryption/signature (KD-CLE and KD-CLS) which naturally obtain the KGC trust level and proposed the way for transforming any conventional certificateless cryptosystem into its key dependent counterparts. In fact, as early as in the first paper on certificateless cryptography, the authors had noticed the issue of KGC trust level and introduced the AP (Al-Riyami-Paterson) binding technique which can lift the authority trust level of any certificateless encryption/signature scheme from 2 to 3. However, they did not provide the formal security proof. This AP binding technique looks very reasonable and has been informally showed secure. However, whether it can be formally prove secure has been an open problem [14]. Yang and Tan [14] explained why the security proof for the AP binding method can not be obtained, then bypassed the security proof for the original AP binding technique, and turned to expand the AP binding technique by involving extra cryptographic tools: random oracles for obtaining security proof in the random oracle model or trapdoor hash functions for obtaining security proof in the standard model.

Our Work. Following Yang and Tan's work [14], this paper tries to solve the above open problem. As the starting point, we improve the security model of conventional certificateless encryption and key dependent certificateless encryption. In fact, the improved points in our new security model focus on replaced public keys. In previous security models such as those in [1,14], for the replaced public key, the relative secret key (also known as secret value in [1,14]) is thought to be known by the attacker and unknown by the challenger. In contrast, we find some contrary cases where the secret key relative to the replaced public key is unknown by the attacker and known by the challenger. In other words, in our improved security model, "being replaced" for public keys will not be used as the criterion for "being known" for the secret key as before. With this discovery, we improve the security model for conventional/key dependent certificateless encryption. With this improved security model, the AP binding technique is formally proved to be one secure generic framework which can generate one secure KD-CLE scheme if the underlying CLE is assumed to be secure. As an application example, one key dependent certificateless encryption scheme with both authority trust level 3 and provable security in the standard model is modularly constructed from the existing conventional certificateless encryption scheme due to Dent et al. [4] through the AP binding framework. We also show that our results can also be naturally extended to other certificateless cryptosystems such as certificateless signatures.

Paper Organization. In Sect. 2, we present the improved security model for conventional CLE and key dependent CLE and review the AP binding technique. In Sect. 3, we show that the AP binding technique is provably secure in our improved security model. In Sect. 4, one key dependent certificateless encryption scheme is modularly constructed by applying the AP binding technique. Some observations on our result are presented in Sect. 5. The conclusion is in Sect. 6.

2 Preliminaries

2.1 Syntax Definitions for CLE and KD-CLE

Following previous works [1,5], we present the syntax definition as follows.

Definition 1. A ("Key Dependent" or "Conventional") certificateless encryption scheme has the following seven algorithms. Here note that conventional CLE and KD-CLE are defined in one syntax framework with the differences pointed out by the notation $[\cdot]_{KD}$.

- $\mathsf{Setup}(1^k) \rightarrow (mpk, msk)$, run by the KGC, takes as input the security parameter 1^k, and then returns a master public/secret key pair (mpk, msk).
- $\mathsf{GenUSK}(mpk) \rightarrow usk$, run by the user, takes mpk as inputs, and returns a user secret key usk. In some previous works, the "user secret key" is also called "secret value".
- $\mathsf{GenUPK}(mpk, usk) \rightarrow upk$, run by the user, takes as input the master public key mpk and the user secret key usk and then returns a user public key upk.
- $\mathsf{GenPPK}(msk, ID, [upk]_{KD}) \rightarrow ppk$, run by KGC, takes as input the master secret key msk, the *optional* user public key upk and the entity identity ID, and then returns the partial private key ppk for the user.

Remark 1: The notation $[\cdot]_{KD}$ means that: (1) upk is not taken as input for "*conventional*" certificateless encryption schemes, but taken as input for "*key dependent*" certificateless encryption schemes (KD-CLE).

- $\mathsf{GenFPK}(mpk, usk, ppk) \rightarrow fpk$, run by the entity, takes as input the master public key mpk, the entity secret key usk, and the partial private key ppk, and then returns the full private key fpk.
- $\mathsf{Encrypt}(mpk, ID, upk, m) \rightarrow c$, run by the sender, takes as input the master public key mpk, the identity ID, the public key upk and one message m, and then outputs the ciphertext c.
- $\mathsf{Decrypt}(mpk, fpk, c) \rightarrow m$, run by the receiver, takes the master public key mpk, the full private key fpk, and a ciphertext c as input, and then outputs the plaintext m.

2.2 Security Models for CLE and KD-CLE

The following steps constitute the common game framework to define CCA (Chosen Ciphertext Attack) security of certificateless encryption schemes (CLE or KD-CLE) for the attacker $\mathcal{A}$ being of Type I or II. After this game, respectively for Type I and Type II attackers, the oracle sets and the oracle query restrictions will be further described in details.

1. $(mpk, msk) \leftarrow \mathsf{Setup}(1^k)$.
   The setup algorithm $\mathsf{Setup}(1^k)$ is run the challenger.
2. $(ID^*, upk^*_{ID^*}, m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}}(mpk)$.
   With the master public key and the oracle access to all oracles in the set $\mathcal{O}$, the attacker $\mathcal{A}$ finally sends the target identity with the public key and message pair $(ID^*, upk^*_{ID^*}, m_0, m_1)$ to the challenger. Different instantiation of the oracle set $\mathcal{O}$, being $\mathcal{O}_I$ or $\mathcal{O}_{II}$, defines the attack type being of Type I or II. The set of oracles ($\mathcal{O}_I$ and $\mathcal{O}_{II}$) will be further described after this game.
3. $c^* \leftarrow \mathsf{Encrypt}(mpk, ID^*, upk^*_{ID^*}, m_b)$, where $b \xleftarrow{R} \{0, 1\}$.
   The challenger chooses a random bit $b \in \{0, 1\}$ and sends ciphertext $c^* \leftarrow \mathsf{Encrypt}(mpk, ID^*, upk^*_{ID^*}, m_b)$ to the attacker. Here note that $upk^*_{ID^*}$ refers to ID^*'s *current* public key.
4. $b' \leftarrow \mathcal{A}^{\mathcal{O}}(c^*)$.
   Given the challenged ciphertext c^*, with access to the oracle set $\mathcal{O}$ restrictively as will be described below, the attacker finally returns a guessed bit b' for b.

The adversary advantage for winning the game is defined to be

$$\mathrm{Adv}^{CCA}_{CLE,\mathcal{A}} = |\Pr[b = b'] - 1/2|.$$

A certificateless encryption scheme is said to be CCA (chosen ciphertext attack) secure, if $Adv^{CCA}_{CLE,\mathcal{A}}(k)$ is negligible in the parameter k for both cases: (1)$\mathcal{A}$ is of Type I, denoted by $\mathcal{A}_I$; (2) $\mathcal{A}$ is of type II, denoted by $\mathcal{A}_{II}$. According to the oracle sets and oracle query restrictions, $\mathcal{A}_{II}$ and $\mathcal{A}_I$ are determined as follows. Oracle set $\mathcal{O}$, instantiated with $\mathcal{O}_I$ for type I attackers or $\mathcal{O}_{II}$ for Type II attackers, consists of some oracles as below:

- $O^{MSK}(mpk)$ presents the master secret key msk corresponding to the master public key. In previous security models [1], the presence of msk of the challenger to the adversary is directly expressed. For expression convenience and without essential difference, here the presence of msk is formalized by this specific oracle.
- $O^{UPK}(ID)$ returns ID's original public key upk_{ID}. The set of all original public keys provided by the challenger for the adversary is denoted by $\mathcal{OPK}$.
- $O^{RPK}(ID, upk)$ changes ID's public key into the value upk. Without loss of generality, for the input ID, the oracle query $O^{UPK}(ID)$ is assumed to be made previously.

- $O^{PPK}(ID, [upk]_{KD})$ outputs the partial private key ppk_{ID} for the identity ID. Just as before mentioned in the syntax definition, only for KD-CLE schemes, the optional parameter of public key upk should be included.
- $O^{FPK}(ID, upk)$ outputs the user's full private key fpk_{ID} for the identity ID and the public key upk. Without loss of generality, upk is assumed to be the current public key of the input identity ID. We also assume that this oracle O^{FPK} will first make the oracle query $O^{RPK}(ID, upk)$ to set upk as ID's user public key, if ID's current public key is not upk.
- $O^{DEC}(ID, upk, c)$ outputs the decryption of c for the identity ID and the current public key upk. This oracle O^{DEC} is assumed to first make the oracle query $O^{RPK}(ID, upk)$ to set upk as ID's user public key, if ID's current public key is not upk.

Remark 2. For the two oracles of O^{FPK} and O^{DEC}, in previous works [1,14], the current public key upk is usually not explicitly included as input. For convenience of expression and without essential difference, here upk is explicitly taken as input.

For Type I adversaries, the oracle set $\mathcal{O}$ is instantiated with

$$\mathcal{O}_I = \{O^{PPK}, O^{FPK}, O^{UPK}, O^{RPK}, O^{DEC}\}$$

under the following restrictions:

I.1 $\mathcal{A}$ cannot make the full private key query $O^{FPK}(ID, upk)$ for $upk \notin \mathcal{OPK}$, since there is no way for the challenger to know the corresponding indispensable secret key.

I.2 $\mathcal{A}$ cannot make the full private key query $O^{FPK}(ID^*, upk^*_{ID^*})$, since this will help $\mathcal{A}$ to trivially succeed.

I.3 $\mathcal{A}$ cannot make the decryption query $O^{DEC}(ID^*, upk^*_{ID^*}, c^*)$, since this will help $\mathcal{A}$ to trivially succeed.

I.4 $\mathcal{A}$ can not make the query for the partial private key $O^{PPK}(ID^*, upk^*_{ID^*})$ for $upk^*_{ID^*} \notin \mathcal{OPK}$, since this will help $\mathcal{A}$ to know the partial private key and the secret key together, and then trivially succeed.

For Type II adversaries, the oracle set $\mathcal{O}$ is instantiated with

$$\mathcal{O}_{II} = \{O^{MSK}, O^{FPK}, O^{UPK}, O^{RPK}, O^{DEC}\}$$

under the following restrictions:

II.1, II.2, II.3 are the same to the above restriction rules I.1, I.2, I.3 respectively.
II.4 $upk^*_{ID^*} \in \mathcal{OPK}$. Otherwise, the Type II adversary, which is able to compute the partial private key by itself, can further get known the target user secret key, and hence can trivially succeed in computing the target full private key.

Remark 3. In the above restriction I.1, for one identity ID and its "current" public key upk, the condition $upk \notin \mathcal{OPK}$ means that the current user secret key usk is known by the adversary and not known by the challenger. In contrast, in previous works [1], the corresponding restriction usually requires that, $\mathcal{A}$ is not allowed to query the full private key of any identity if the corresponding public key has ever been replaced. Similar analysis works for the condition $upk^*_{ID^*} \notin \mathcal{OPK}$ in the restriction I.4 and $upk^*_{ID^*} \in \mathcal{OPK}$ in the restriction II.4.

2.3 Al-Riyami-Paterson Binding Technique

Al-Riyami-Paterson binding method is as follows: for a traditional certificateless encryption scheme $\mathcal{CLE}$ = (Setup, GenUSK, GenUPK, GenPPK, GenFPK, Encrypt, Decrypt), construct a key dependent CLE scheme $\mathcal{KD}$-$\mathcal{CLE}$ = (Setup', GenUSK', GenUPK', GenPPK', GenFPK', Encrypt', Decrypt') as follows:

- (Setup', GenUSK', GenPPK', GenFPK', Decrypt') are same as (Setup, GenUSK, GenUPK, GenFPK, Decrypt) respectively.
- GenPPK'(msk, ID, upk): let $ID' = ID||upk$, run $usk \leftarrow$ GenPPK(msk, ID') and return usk.
- Encrypt'(mpk, ID, upk, m): let $ID' = ID||upk$, run $c \leftarrow$ Encrypt(mpk, ID', upk, m) and return c.

In [14], Yang and Tan analyzed difficulties in developing a generic proof for the above AP binding technique. To get provable security, they proposed two modified versions of the AP binding technique: in stead of using the original AP binding $ID' = ID||upk$, they used (1) $ID' = H(ID||upk)$ for provable security in random oracle model, where $H(\cdot)$ is a cryptographic hash function taken as one random oracle; or (2) $ID' = H_{pk}(ID||upk)$ in the standard model, where $H_{pk}(\cdot)$ is a trapdoor hash function. At the end, they proposed the open problem that, whether the Al-Riyami-Paterson binding technique can be proved to be secure as a generic transformation.

3 Security Proof for AP Binding Technique

Theorem 1. If the conventional certificateless encryption scheme $\mathcal{CLE}$ is CCA secure against PPT adversaries of type I, then the corresponding key dependent certificateless encryption scheme $\mathcal{KD}$-$\mathcal{CLE}$ is also CCA secure against PPT adversaries of type I.

Proof. We prove it by contradiction. We assume that there is an adversary $\mathcal{A}$ (against $\mathcal{KD}$-$\mathcal{CLE}$) who wins a non-negligible advantage. We try to construct one adversary $\mathcal{B}$ (against $\mathcal{CLE}$) whose advantage is non-negligible. $\mathcal{B}$ simulates the game for $\mathcal{A}$. $\mathcal{B}$ first passes mpk to $\mathcal{A}$ and answers $\mathcal{A}$'s queries as below. Here note that, without loss of generality, each oracle for $\mathcal{A}$ against $\mathcal{KD}$-$\mathcal{CLE}$ and its counterpart oracle for $\mathcal{B}$ against $\mathcal{CLE}$ use the same notation. For example, the decryption oracle for $\mathcal{KD}$-$\mathcal{CLE}$ and the decryption oracle for $\mathcal{CLE}$ both use the notation $O^{DEC}(\cdot)$.

- $O^{UPK}(ID)$: If the user ID has ever been queried by $\mathcal{A}$, then the corresponding original public key is returned according to the below recording list L_{opk}. Otherwise, $\mathcal{B}$ selects a random valid CLE identity string ID'', makes a $O^{UPK}(ID'')$ query to its own oracle, and returns the original public key $upk_{ID''}$ of ID'' as the answer for $O^{UPK}(ID)$. To record the case that ID and ID'' has the same original public key, adds (ID, ID'') to the initially empty list L_{opk}.
- $O^{RPK}(ID, upk)$: For $ID' = ID||upk$, $\mathcal{B}$ makes a query $O^{RPK}(ID', upk)$ to its own challenger. After this oracle query, ID for $\mathcal{KD}$-$\mathcal{CLE}$ and ID' for $\mathcal{CLE}$ have the same current public key and full private key.
- $O^{PPK}(ID, upk)$: For $ID' = ID||upk$, $\mathcal{B}$ makes a query $O^{PPK}(ID')$ to its own challenger, and transfers the answer to $\mathcal{A}$.
- $O^{FPK}(ID, upk)$: For $ID' = ID||upk$, $\mathcal{B}$ first makes a public key replacing oracle query $O^{RPK}(ID', upk)$ to ensure that the current public key of ID' is upk, and then issues a query $O^{FPK}(ID', upk)$ to the challenger of itself, and transfers the answer to $\mathcal{A}$.
- $O^{DEC}(ID, upk, c)$: For $ID' = ID||upk$, $\mathcal{B}$ first makes a public key replacing oracle query $O^{RPK}(ID', upk)$ to ensure that the current public key of ID' is upk, and then makes a $O^{DEC}(ID', upk, c)$ query to its own challenger, and transfers the answer to $\mathcal{A}$.

During these oracle queries, when $\mathcal{A}$ outputs $(ID^*, upk^*_{ID^*}, m_0, m_1)$ as the challenge, $\mathcal{B}$ first makes a public key replacing oracle query $O^{RPK}(ID^{*\prime}, upk^*_{ID^*})$ to ensure that the current public key of $ID^{*\prime}$ is $upk^*_{ID^*}$, and then transfers $(ID^{*\prime}$, $upk^*_{ID^*}, m_0, m_1)$ to its own challenger for $ID^{*\prime} = ID^*||upk^*_{ID^*}$. After the target ciphertext c^* is returned from the challenger, $\mathcal{B}$ then transfers c^* to $\mathcal{A}$ as the challenging ciphertext. Then $\mathcal{B}$ answers the oracle queries as before. At the end, when $\mathcal{A}$ gives a bit b' to $\mathcal{B}$, $\mathcal{B}$ transfers b' to its own challenger.

Next, we analyze the restrictions on oracle access.

I.1 For $\mathcal{A}$'s full private query $O^{FPK}(ID, upk)$, according to the restriction, upk should be an original public key of one certain identity generated by its challenger (simulated by $\mathcal{B}$). However, $\mathcal{B}$ never generates any public key by itself. Hence upk must be ever gotten from $\mathcal{B}$'s challenger and then transferred to $\mathcal{A}$. Hence $\mathcal{B}$ also never violate restriction I.1.

I.2 According to $\mathcal{B}$'s simulation of O^{FPK}, since $\mathcal{A}$ don't violates this restriction to make the query $O^{FPK}(ID^*, pk^*_{ID^*})$, $\mathcal{B}$ will not make the query $O^{FPK}(ID^{*\prime}, pk^*_{ID^*}) = O^{FPK}(ID^* \, || pk^*_{ID^*}, pk^*_{ID^*})$. Hence, $\mathcal{B}$ will not violates this rule.

I.3 According to $\mathcal{B}$'s simulation of O^{DEC}, since $\mathcal{A}$ don't violates this rule to make the query $O^{DEC}(ID^*, pk^*_{ID^*}, c^*)$, $\mathcal{B}$ will not make the query $O^{DEC}(ID^{*\prime}, pk^*_{ID^*}, c^*) = O^{DEC}(ID^* \, || pk^*_{ID^*}, pk^*_{ID^*}, c^*)$. Hence, $\mathcal{B}$ will not violates this rule.

I.4 According to $\mathcal{B}$'s simulation of O^{PPK}, since $\mathcal{A}$ don't violates this rule to make a partial private key query $O^{PPK}(ID^*,\, upk^*_{ID^*})$ for $upk^*_{ID^*} \notin \mathcal{OPK}$ (This means $upk^*_{ID^*}$ is generated by $\mathcal{A}$) and $\mathcal{B}$ never generates any public key by itself. Hence, $\mathcal{B}$ will not violates this rule.

Based on the above description, it can be seen that $\mathcal{B}$ makes the successful simulation for $\mathcal{A}$ without violating any restriction and wins the game only if $\mathcal{A}$ succeeds. Additionally, $\mathcal{B}$'s running time is equal to that of $\mathcal{A}$ without considering some trivial differences. Now, the proof is completed.

Theorem 2. If the conventional certificateless encryption scheme $\mathcal{CLE}$ is CCA secure against PPT adversaries of type II, then the corresponding key dependent certificateless encryption scheme $\mathcal{KD}$-$\mathcal{CLE}$ is CCA secure against PPT adversaries of type II.

Proof. We prove it by contradiction. Assume we have a Type II adversary $\mathcal{A}$ (against $\mathcal{KD}$-$\mathcal{CLE}$) whose advantage is non-negligible, we try to construct one Type II adversary $\mathcal{B}$ (against $\mathcal{CLE}$) whose advantage is also non-negligible. $\mathcal{B}$ simulates the game for $\mathcal{A}$. $\mathcal{B}$ first passes mpk to $\mathcal{A}$ and answers $\mathcal{A}$'s queries as below.

- $O^{MSK}(mpk)$: $\mathcal{B}$ makes the query $O^{MSK}(mpk)$ to its own challenger, and transfer the answer to $\mathcal{A}$.
- $O^{UPK}(ID)$, $O^{RPK}(ID, upk)$, $O^{FPK}(ID, upk)$, $O^{DEC}(ID, upk, c)$: $\mathcal{B}$ simulates these oracle queries just as in the proof for Theorem 1.

During these oracle queries, when $\mathcal{A}$ outputs $(ID^*, upk^*_{ID^*}, m_0, m_1)$ as the target, $\mathcal{B}$ sends $(ID^{*\prime},\, upk^*_{ID^*}, m_0, m_1)$ to its challenger for $ID^{*\prime} = ID^* || upk^*_{ID^*}$. After the target ciphertext c^* is received from the challenger, B transfers c^* to $\mathcal{A}$ as the target ciphertext. B then answers the oracle queries as before. At the end, when $\mathcal{A}$ provides a bit b' to $\mathcal{B}$, $\mathcal{B}$ transfers b' to its own challenger.

Just as analyzed in the proof for Theorem 1, $\mathcal{B}$ will not violates the restriction rules II.1, II.2, II.3. According to the restriction rule II.4, $upk^*_{ID^*}$ should be generated by $\mathcal{A}$'s challenger (here simulated by $\mathcal{B}$). However during the whole simulation process, $\mathcal{B}$ never generated any public key by itself. Hence $upk^{*}_{ID^{*}}$ is not generated by $\mathcal{A}$ or $\mathcal{B}$, but be generated by $\mathcal{B}$'s challenger and then given to $\mathcal{B}$. Hence $\mathcal{B}$ does not violates the restriction rule II.4.

Based on the above description, it can be seen that $\mathcal{B}$ makes the successful simulation for $\mathcal{A}$ without violating any restriction and wins the game only if $\mathcal{A}$ succeeds. Additionally, $\mathcal{B}$'s running time is same to that of $\mathcal{A}$ without considering some trivial differences. Now, the proof is completed.

4 KD-CLE Scheme Secure in the Standard Model

When the AP binding technique is applied to the Dent-Libert-Paterson conventional CLE scheme, the following KD-CLE scheme is constructed. The KD-CLE scheme works as follows.

First, we present bilinear pairing which is used for our KD-CLE construction. Let $\mathbb{G}, \mathbb{G}_T$ be the prime q-order groups and let g be $\mathbb{G}$'s generator, where $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicatively represented. A map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is called a bilinear pairing, if the below conditions all hold: (1) e is bilinear: $e(g^a, g^b) = e(g, g)^{ab}$ where $a, b \in \mathbb{Z}_q^*$; (2) e is non-degenerate: $e(g, g) \neq 1$, where 1 is $\mathbb{G}_T$'s identity; (3) e can be efficiently computed.

- $\mathsf{Setup}(1^k)$: Let $(\mathbb{G}, \mathbb{G}_T)$ be bilinear pairing groups of order $q > 2^k$ and let g a generator of $\mathbb{G}$. Set $g_1 = g^{\gamma}$, where γ is randomly chosen from $\mathbb{Z}_p^*$, and randomly pick $g_2, u', u_1, u_2, \ldots, u_n, v', v_1, v_2, \ldots, v_n \in \mathbb{G}$. For $i = i_1 i_2 \ldots i_n$ and $w = w_1 w_2 \ldots w_n$, the two functions are as below.

$$F_u(i) = u' \prod_{j=1}^{n} u_j^{i_j} \text{ and } F_v(w) = v' \prod_{j=1}^{n} v_j^{w_j}$$

  The hash function $H : \{0,1\}^* \leftarrow \{0,1\}^n$ is collision resistant (here note, H will not be assumed as the random oracle in the security proof). Let the master public key be

$$mpk \leftarrow (g, g_1, g_2, u', u_1, \ldots, u_n, v', v_1, \ldots, v_n)$$

  and the master secret key be $msk \leftarrow g_2^{\gamma}$.
- $\mathsf{GenUSK}(mpk)$: Return the user secret key x_{ID} which is randomly chosen from $\mathbb{Z}_p^*$.
- $\mathsf{GenUPK}(x_{ID}, mpk)$: Return $upk_{ID} = (X, Y) = (g^{x_{ID}}, g_1^{x_{ID}})$.
- $\mathsf{GenPPK}(mpk, \gamma, ID, upk_{ID})$: Pick $r \leftarrow \mathbb{Z}_p^*$ and return $d_{ID} = (d_1, d_2) = (g_2^{\gamma} \cdot F_u(i)^r, g^r)$, where $i = H(ID||upk_{ID})$.
- $\mathsf{GenFPK}(x_{ID}, d_{ID}, mpk)$: Randomly choose r' from $\mathbb{Z}_q^*$ and set the private key as

$$sk_{ID} = (s_1, s_2) = (d_1^{x_{ID}} \cdot F_u(i)^{r'}, d_2^{x_{ID}} \cdot g^{r'}) = (g_2^{\gamma x_{ID}} \cdot F_u(i)^t, g^t)$$

  where $i = H(ID||upk_{ID}), t = rx_{ID} + r'$.

- Encrypt(m, upk_{ID}, ID, mpk): If upk_{ID} is correctly shaped, randomly chooses s from $\mathbb{Z}_q^*$ and then computes

$$C = (C_0, C_1, C_2, C_3) = (m \cdot e(Y, g_2)^s, g^s, F_u(i)^s, F_v(w)^s),$$

  where $i = H(ID||upk_{ID}), w \leftarrow H(C_0, C_1, C_2, ID, pk_{ID})$.
- Decrypt(C, sk_{ID}, mpk): If

$$e(C_1, F_u(i) \cdot F_v(w)) = e(g, C_2 \cdot C_3), \text{where } i = H(ID||upk_{ID}),$$

  then return $m \leftarrow C_0 \cdot \frac{e(C_2,s_2)}{e(C_1,s_1)}$.

Lemma 1. The Dent-Libert-Paterson conventional CLE scheme [4] is CCA secure in our improved security model, if the 3-DDH assumption holds in the group $\mathbb{G}$.

Proof. In [4], for Type I attackers and Type II attackers, Theorems 2 and 3 respectively described the CCA provable security of the Dent-Libert-Paterson conventional CLE scheme in their security model. Now, we show that their security proof can be modified into the security proof in our new security model. First, revisit Remark 3 which explains the differences between our security model and that in [4]: with the *same* purpose to formally capture the case that the secret key of one identity ID is unknown by the adversary (or known by the challenger), we require that the public key should be the original one of **ANY** identity (ID or $ID' \neq ID$), while the public key should be the original (never replaced) one of **ONLY** ID in the security model of [4]. In other words, as said in Remark 3, for the case that the current secret key is known by the challenger, the previous security model ignored the subcase that the current public key of ID is not the original public key of itself, but is the original public key of the other identity ID'. Following this observation on security model differences, during checking and modifying the original security proof in [4], we only need to focus on the steps where the security proof aims to use the case that the challenger (or adversary) knows (or does not know) the secret key for the current public key, but uses the criterion whether the public key is the replaced one. We modifies these places by using the new criterion whether the public key has been original one of any identity, to decide whether the challenger (or adversary) knows (or does not know) the secret key for the current public key. In fact, at these places in the security proof, the relative logic is based on not how to ensure that the challenger (or adversary) knows (or unknows) the current secret key, but the ultimate result that the challenger (or adversary) knows (or unknows) the current secret key. For example, the case $c_{mode} = 1$ in Game 9 in the security proof of Theorem 2 in [4] only associates with whether the secret key is unknown by adversary (or known by the challenger). Hence, these modifications do not affect the logics of the original security proof in [4], but use the more reasonable criterion to fit the logics of the security proof. By this analysis, the security proof in [4] can be directly used as that for the above lemma with a few trivial modifications. Here we omit the detailed the proof.

Following Theorems 1, 2 and Lemma 1, the following corollary can be directly obtained.

Theorem 3. The above KD-CLE scheme is CCA secure if the 3-DDH problem is assumed to be intractable in the group $\mathbb{G}$.

5 Further Discussions

Now we point out the following observations. Firstly, many existing CLE schemes, provably secure in their "old" security model, can remain provably secure in our new security model. For example, as shown in Lemma 1, the Dent-Libert-Paterson conventional CLE scheme [4] remains provably secure in our new security model, by slightly or even trivially modifying the previous proof in the "old" security model. In the security proof of Lemma 1, the reasons why these modifications work has been concretely explained.

Secondly, many efficient KD-CLE schemes can be modularly constructed from existing conventional CLE schemes. In fact, the security proof in our new security model helps to "revive" the "perfect" and "old" generic transforming method from conventional CLE to KD-CLE due to Al-Rayami and Paterson. "Perfect" means that, unlike the results in [14], the AP transformation can construct KD-CLE from conventional CLE with almost no cost.

Thirdly, similar results for key-dependent certificateless signatures can be obviously obtained [11,14], following the results on modularly transforming conventional CLE into KD-CLE. Since this further extension is trivial, we omitted these detailed descriptions for KD-CLS. Additionally, following the results for key dependent encryption and signatures, other certificateless primitives, such as signcryption and authentication [13] in the key dependent sense can also be trivially obtained.

At last, our improved security model is more comprehensive and may be technically significant in basic theory for certificateless cryptography. In fact, there are many works on improving the security model for certificateless primitives [2,5,9,10,12,14,16]. However, to capture "whether the current secret key is unknown by the adversary" (simultaneously known by the challenger), all previous security models use the standard "whether the public key has been replaced" while our security model uses the standard "whether the public key has been generated by the challenger (denoted by $upk \in \mathcal{OPK}$)".

6 Conclusion

We positively answered the open problem whether the AP binding technique [1,14] is provably secure. This result means that any provably secure conventional CLE/CLS scheme with KGC trust level 2 can be almost directly transformed into the corresponding provably secure key dependent CLE/CLS scheme with KGC trust level 3. As a example, we modularly constructed one key dependent CLE scheme with KGC trust level 3 and provable security in the stand model.

Although the improved CLE security is proposed for proving security for the AP binding technique, it may have independent significance in certificateless cryptography.

Acknowledgements. This work is partially supported by National Natural Science Foundation of China (Nos. 61202475, 61133014, 61472114), the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD, Nanjing University of Information Science & Technology), Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET, Nanjing University of Information Science & Technology), and the open foundation of Key Laboratory of Information Security, School of Mathematics and Information Science.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003). doi:10.1007/978-3-540-40061-5_29
2. Au, M.H., Chen, J., Liu, J.K., Mu, Y., Wong, D.S., Yang, G.: Malicious KGC attack in certificateless cryptography. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, pp. 302–311. ACM (2007)
3. Dan, B., Franklin, M.: Identity-based encryption from the weil pairing. SIAM J. Comput. **32**(3), 213–229 (2003)
4. Dent, A.W., Libert, B., Paterson, K.G.: Certificateless encryption schemes strongly secure in the standard model. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 344–359. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78440-1_20
5. Dent, A.W.: A survey of certificateless encryption schemes and security models. Int. J. Inf. Secur. **7**(5), 349–377 (2008)
6. Girault, M.: Self-certified public keys. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 490–497. Springer, Heidelberg (1991). doi:10.1007/3-540-46416-6_42
7. Hu, B.C., Wong, D.S., Zhang, Z., Deng, X.: Certificateless signature: a new security model and an improved generic construction. Des. Codes Cryptogr. **42**(2), 109–126 (2007)
8. Huang, Q., Wong, D.S.: Generic certificateless encryption secure against malicious-but-passive kgc attacks in the standard model. J. Comput. Sci. Technol. **25**(4), 807–826 (2010)
9. Huang, X., Mu, Y., Susilo, W., Wong, D.S., Wu, W.: Certificateless signatures: new schemes and security models. Comput. J. **55**(4), 457–474 (2012)
10. Liu, J.K., Au, M.H., Susilo, W.: 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, pp. 273–283. ACM (2007)
11. Pang, L., Hu, Y., Liu, Y., Xu, K., Li, H.: Efficient and secure certificateless signature scheme in the standard model. Theoret. Comput. Sci. **30**(5), 3025–3041 (2017)
12. Wang, X.A., Huang, X., Yang, X.: Further observations on certificateless public key encryption. In: Yung, M., Liu, P., Lin, D. (eds.) Inscrypt 2008. LNCS, vol. 5487, pp. 217–239. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01440-6_18

13. Xiong, H.: Cost-effective scalable and anonymous certificateless remote authentication protocol. IEEE Trans. Inf. Forensics Secur. **9**(12), 2327–2339 (2014)
14. Yang, G., Tan, C.H.: Certificateless cryptography with KGC trust level 3. Theor. Comput. Sci. **412**(39), 5446–5457 (2011)
15. Yang, W., Zhang, F., Shen, L.: Efficient certificateless encryption withstanding attacks from malicious KGC without using random oracles. Secur. Commun. Netw. **7**(2), 445–454 (2014)
16. Zhang, F., Shen, L., Wu, G.: Notes on the security of certificateless aggregate signature schemes. Inf. Sci. **287**, 32–37 (2014)