Generic Construction of Certificate Based Encryption from Certificateless Encryption Revisited

Wei Gao[a,b], Guilin Wang[c], Kefei Chen[a], Xueli Wang[d]

[a] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[b] *School of Mathematics and Information, Ludong University, Yantai 264025, China*
[c] *School of Computer Science & Software Engineering, University of Wollongong, NSW 2522, Australia*
[d] *School of Mathematics, South China Normal University, Guangzhou 510631, China*

Abstract

Certificateless public key encryption (CLE) and certificate based encryption (CBE) are two novel public key cryptographic primitives requiring no authenticity verification of the recipient's public key. Both of them are motivated to simultaneously solve the heavy certificate management problem inherent in the traditional public key encryption (PKE) and the key escrow problem inherent in the identity-based encryption (IBE). It is an attractive cryptographic task to formally explore the relation between CBE and CLE. In 2005, Al-Riyami and Paterson proposed one general conversion from CLE to CBE. Shortly later, Kang and Park pointed out a flaw in the security proof of Al-Riyami-Paterson conversion. In 2012, Wu et al. proposed another generic conversion from CLE to CBE. Compared with Al-Riyami-Paterson conversion, Wu et al.'s method can be proved secure, but it has to additionally involve collision resistant hash functions. It remains an open problem whether the generic conversion due to Al-Riyami and Paterson, which is very neat, is provably secure. We aim to solve this open problem. First, we formalize CLE's new security model, featured by introducing a new security property overlooked by previous security models. With this new security model as the basic technique, we succeed in proving that the Al-Riyami-Paterson generic

Email addresses: sdgaowei@gmail.com (Wei Gao), guilin@uow.edu.au (Guilin Wang), kfchen@sjtu.edu.cn (Kefei Chen), wangxuyuyan@gmail.com (Xueli Wang)

[1]This work is completed when the first author visited University of Wollongong (2011-2012)

conversion from CLE to CBE is secure, if the CLE scheme is secure in our new security model. A concrete provably secure CBE scheme is presented to demonstrate the application of our result.

Keywords: certificateless encryption, certificate based encryption, identity based encryption, provable security.

---

1. Introduction

1.1. Backgroud

Traditional PKE. For traditional public-key encryption (PKE), each user has a pair of public key (assumed to be publicly available) and private key (only known to the owner). The sender uses the public key of the intended recipient to encrypt messages, while the recipient uses his private key to decrypt ciphertexts. In this way, there is no secure channel needed for the sender to transfer encryption keys, since the public keys are publicly known. However, it is an issue how to assure the sender that a claimed public key belongs to a specific recipient. In current practice, the authenticity of public keys is guaranteed by certificates that are issued by a trusted third party called Certification Authority (CA). The management of certificates, which includes certification, storage, revocation and distribution, is the main bottleneck for PKE in deployment.

Identity-based PKE. In 1984, Shamir [1] proposed the concept of identity-based PKE (IBE), which aims to ease the public key mangagement by using a user's well-known identity as its public key. On the one hand, since the identity information is publicly known, the need of certification can be eliminated in IBE. On the other hand, because each user's private key is generated by a trusted party called Private Key Generation (PKG), the PKG knows all users' private keys for decryption. Consequently, IBE suffers from the main drawback of being inherently key escrowed, which limits the applicability of IBE. The first practical IBE scheme was proposed by Boneh and Franklin in 2001 [2], which exploits bilinear maps on elliptic curves and is proven secure in the random oracle model.

Certificate-based encryption. Motivated to overcome the inherent key escrow problem in IBE, the concept of certificate-based encryption (CBE) was introduced by Gentry in [3]. In this paradigm, the decryption key of each user consists of indispensable parts, i.e., the private key selected by the user and the up-to-date certificate issued by the CA using an IBE. The certificate

acts not only one part of the decryption key but also a traditional public key certificate. In this way, there is no need for the sender to check the existence of certificates, since he knows that the ciphertext cannot be decrypted if there is no corresponding certificate. Hence, CBE greatly lessens the task for certificate management. For example, CRL (Certificate Revocation List) [4] is not needed any more. At the same time, there is no key-escrow problem in CBE, since the private key, one of the two indispensable parts for decryption, is generated by the user itself and then not known by the PKG. Additionally, different from IBE there is no secret key distribution problem in CBE as each user's certificate needs not be kept secret.

Certificateless public key encryption. With the same motivation as CBE, the concept of certificateless public key encryption (CLE) was introduced by Al-Riyami and Paterson independently in [5]. In CLE, each user has two secrets: a secret value SV chosen by the user and a partial private key PPK generated by the PKG. The full private key is the output of a function by taking SV and PPK as the input, and hence can be known only by the user. As a result, on one hand, unlike IBE, CLE does not suffer from key escrow, since PKG does not have access to the user's secret value SV. On the other hand, unlike traditional PKE, CLE does not require certificates to guarantee the authenticity of public keys, because any attacker other than the PKG is not able to figure out the partial private key PPK for an authentic or fake public key. For the survey on CLE, please refer to [6].

In CBE and CLE, due to the lack of checking certificates, malicious parties can replace an entity's public key with a false key, and other entities may be duped to use the false key in encryption, which is known as *key replacement attack*. To define security of CBE and CLE, the attackers are divided into two types, i.e., Type I attackers and Type II attackers. Type I attackers are malicious outsiders who are allowed to replace public keys but don't know the PKG's master key. Type I attacker is the malicious or compromised PKG with the master key for generating certificates (for CBE) or partial private key (for CLE) but it is not allowed to replace public keys. We do need to trust the PKG as an authority so that it never replaces any user's public key. Otherwise, no IBE or CLE can be secure as the authority can trivially generate the decryption key by first generating a new private/public key pair and then issue the certificate (for CBE) or the partial private key (for CLE). More discussions on trust levels in CLE were given in [7].

1.2. Related Works

Related Works. On the one hand, although CBE and CLE were developed independently, both of them can be conceptually seen as intermediates between traditional PKE and IBE, seeking to simplify certificate management while avoiding the key escrow property of identity-based cryptography. On the other hand, they are obviously different as CBE has public key certificates but CLE does not. So a natural issue is to ask the cryptographic relation between the two concepts. This idea motivated the work by Yum and Lee [8, 9], in which they tried to show a formal equivalence among IBE, CBE and CLE. In particular, their intention was to show that IBE implies both CBE and CLE by giving a generic conversion from IBE to those primitives. However, Galindo et al. (2006) pointed out that a dishonest authority could break the security of their generic constructions [10].

In 2005, Al-Riyami and Paterson in [11] proposed the conversion from a secure CLE scheme to a secure CBE scheme and proved its security. Shortly, Kang and Park [13] pointed out one critical flaw during the security proof of Al-Riiyami-Paterson conversion. The flaw occurs in the security proof against Type II attackers, while the security proof can go through for Type I attackers. Without being able to further solve this security problem, they left the problem open: whether this conversion is provably secure, i.e., how to prove its security or break it. Very recently, Wu et al. [14] proposed another generic conversion from CLE to CBE with provable security. Compared with Al-Riyami-Paterson conversion, this new method is more complicated for involving additional collision resistant hash functions. As noted in [15], however, collision resistant hash functions may result in additional assumptions, lower computation efficiency and looser tightness in proof security.

1.3. Motivation and Contributions

Motivation. As mentioned above, two generic methods have been proposed for converting CLE into CBE: Al-Riyami-Paterson conversion is optimal in efficiency and natural to understand, but its provable security is an open problem; Wu et al.'s method achieves provable security, but it involves additional cryptographic primitives, i.e. collision resistant hash functions. The paper due to Kang and Park analyzed why the security proof fails for Al-Riyami-Paterson conversion but did not give any clue to solve this issue. We realize that public key replacement plays a core role in the security proof of Al-Riyami-Paterson conversion. By deeply exploring public

key replacement, we are motivated to make further on the security of Al-Riyami-Paterson conversion: (1) Whether this intuitive and neat conversion is secure; (2) If yes, how to proves its security; (3) If no, how to show this by presenting attacks or disproof.

CONTRIBUTIONS. By confirming the provable security of Al-Riyami-Paterson conversion, the contributions of this paper are two-fold. In practice, it means that many concrete CBE schemes can be constructed in batch from the existing secure CLE. In theory, it implies that the notion of CLE can be formally implicit in that of CBE. We now briefly explain how this is achieved.

First, we formalize a novelly extended security model for CLE. Its key feature is to consider the Type-II attacker more flexible than those in the existing security models. Our model *partially* forbids Type-II attacker to replace the public key of a target user, while the existing security models completely forbid this power to Type-II attackers. Specifically, our new model allows a Type-II attacker to replace the public key, under the condition that the attacker does not know the corresponding private key. For example, in our model the PKG (Type II adversary) may replace the public key of target user Alice with that of user Bob and then deceives the encrypter Cindy into encrypting one message for Alice with this fake public key. In this way, the PKG does not know the corresponding private key, although he already replaced public key. Hence, it is reasonable to believe that the PKG should remain unable to decrypt ciphertexts under public key replacement. In other words, if he can decrypt such a ciphertext, this behavior is interesting and should be taken into account as an attack.

Interestingly, our new model can be supported by most existing CLE schemes that are secure in the models not allowing Type-II attackers to replace public key in any form at all. In other words, the new attack model may be just literally stronger than previous attack models. The intuitive reason is that our new security model still meets the basic rule that the attacker does not know the private key or the certificate. It is this observation, which are neglected in existing study, that leads us to complete the security proof for Al-Riyami-Paterson conversion.

Second, we prove that any CLE scheme secure in this new security model can be directly transformed into a secure CBE scheme using Al-Riyami-Paterson conversion method. Compared to Wu et al.'s result, our security proof does not need any additional assumptions such as random oracles or collision resistant hash functions. This means that a CLE scheme secure in the stand model will lead to a CBE scheme secure in the standard model

without introducing any additional assumptions. To show the application of our result, a concrete provably secure CBE scheme is presented. Our work again confirms the intuitive opinion that CLE and CBE are closely related to each other.

1.4. Orgnization

Organization. Section 2 reviews the syntax definition of CLE and proposes the new security model for CLE. Section 3 briefly reviews the definition and security model for CBE. Section 4 presents the security proof for the Al-Riyami and Paterson's generic construction of CBE from CLE proposed in PKC 2005. Section 5 shows the CLE scheme proposed in [11] is secure in the new security model and hence can be used to generate a provably secure CBE scheme. At last, Section 6 draws the conclusion.

2. Certificateless Public Key Encryption

In this section, we first review the syntax definition of CLE [5] which specifies the algorithms. Then the new security model of CLE is proposed. As informally mentioned in introduction, compared with other existing secuity models, its basic feature is to allow Type II attackers to conditionally replace public keys. At last, we compare the new attack models with other existing ones. This comparison will show that the new security model is more comprehensive and helps to prove the security of Al-Riyami-Paterson conversion.

2.1. Syntax of CLE

Definition 1. [Syntax of CLE]. A *Certificateless Public Key Encryption Scheme* consists of the following algorithms, where the prefix "CL." is used to specify that this is in the certificateless system.

- $\mathsf{CL.Setup}(1^k) \rightarrow (msk, params)$.

  It takes 1^k as input where k is the security parameter, and returns a master secret key msk and the system parameter $params$.

- $\mathsf{CL.SetSecretValue}(params) \rightarrow x_{ID}$.

  It takes as inputs $params$, and returns a secret value x_{ID} for the identity ID.

- CL.SetPublicKey$(params, x_{ID}) \rightarrow pk_{ID}$.

  It takes $params$ and x_{ID} as input and returns the user public key pk_{ID}.

- CL.ExtractPartialPrivateKey$(msk, ID) \rightarrow d_{ID}$.

  It takes msk and ID as input and returns the partial private key d_{ID}.

- CL.SetPrivateKey$(params, x_{ID}, d_{ID}) \rightarrow sk_{ID}$.

  It takes $params, x_{ID}$ and d_{ID} as input and returns the full private key sk_{ID}.

- CL.Encrypt$(params, ID, pk_{ID}, m) \rightarrow c$.

  It takes as input the master public key $params$, a user's identity ID, a user's public key pk_{ID} and the message m, and outputs the ciphertext c.

- CL.Decrypt$(params, sk_{ID}, c) \rightarrow m$.

  It takes the master public key $params$, the full private key sk_{ID}, and a ciphertext c as input and returns a plaintext m.

CORRECTNESS requires that ciphertexts generated by the algorithm CL.Encrypt can be correctly decrypted using CL.Decrypt: For any $c =$ CL.Encrypt($params$, ID, pk_{ID}, m), Pr[CL.Decrypt$(params, sk_{ID}, c) = m$]=1.

2.2. Security Definition of CLE

Generally speaking, the attack model should be made as strong as possible to involve as many kinds of attacks as possible. In contrast, here we should make the new attack model as weak as possible, under the condition that it is enough for the CLE scheme to be transformed into the secure CBE scheme through Al-Riaymi-Paterson generic method. In other words, given Al-Riyami-Paterson conversion method to construct CBE from CLE, we should try to find the lower bound of CLE security level which is enough for provably secure conversion. This brings forth some application merits, such as making as many existing CLE schemes as possible suitable for Al-Riyami-Paterson method. Compared with the representative security model [11, 13], we allow Type II attacker to replace public keys under the condition that the secret value remains unknown to the Type II attacker. The more detailed comments are provided in the next subsection.

Definition 2. [New Security of CLE]. A CLE scheme is CL-IND-CCA secure if no polynomially bounded adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, being Type-I adversary $\mathcal{A}^I$ or Type-II advresary $\mathcal{A}^{II}$ (their detailed description will be given below after this definition), has a non-negligible advantage in the following CLE game. First, the four phases of the game are presented. Then, the description of oracles and two types of attack models mentioned in the four phases is provided afterwards.

- Setup Phase: $\mathsf{Setup}(1^k) \rightarrow (params, msk)$.

  Challenger $\mathcal{C}$ takes a security parameter 1^k as input and runs the **CL.Setup** algorithm. It gives $\mathcal{A}$ the system parameter $params$. If $\mathcal{A}$ is of Type I, then $\mathcal{C}$ keeps msk to itself. Otherwise, it gives msk to $\mathcal{A}$.

- Phase 1: $\mathcal{A}_1^O(params, [msk]) \rightarrow (ID^*, m_0, m_1)$.

  After making a sequence of oracle queries with some restriction rules (see below after this game description), $\mathcal{A}$ terminates by outputting the challenge identity ID^* with the corresponding challenge public key $pk^*_{ID^*}$, two messages of equal length (m_0, m_1).

- Challenge Phase:
  $\mathsf{CL.Encrypt}(params, ID^*, pk^*_{ID^*}, m_b) \rightarrow c^*$.

  The challenger randomly chooses a bit $b \in \{0, 1\}$ and computes the challenge ciphertext

  $$C^* = \mathsf{CL.Encrypt}(params, ID^*, pk^*_{ID^*}, m_b)$$

  using the value of $pk^*_{ID^*}$ currently associated with the identity ID^*.

- Phase 2: $\mathcal{A}_2^O(c^*) \rightarrow b'$.

  After a sequence of oracle queries with some restrictions (see bellow after this game framework), $\mathcal{A}$ terminates by outputting a guess b' for b. The advantage of $\mathcal{A}$ in this game is defined to be:

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

Oracles for $\mathcal{A}$. We first define the oracles that the attacker may have access to. Then we define the two types of adversaries based on the oracles allowed.

- RequestPublicKey(ID). The attacker supplies an identity ID and the challenger responds with the public key pk_{ID} for ID. If the identity ID has no associated public key, then the challenger generates a public key for ID by running CL.SetSecretValue and CL.SetPublicKey.

- ReplacePublicKey(ID, pk'_{ID}). The attacker supplies an identity ID and a public key value pk'_{ID} in the public key space, and the challenger replaces the current public key with pk'_{ID}.

- RequestPartialPrivateKey(ID). The attacker supplies an identity ID and the challenger responds with the partial private key d_{ID}.

- Decrypt-I(ID, x_{ID}, c). The attacker supplies an identity ID and its current secret value x_{ID} and a ciphertext c, and the challenger responds with the decryption of c under the full private key sk_{ID} corresponding to the current public key. This oracle is provided for the adversary of Type I.

- Decrypt-II(ID, c). The attacker supplies the ciphertext c and the identity ID whose current public key is required to be the original public key of itself or some other identity obtained from the **RequestPublicKey** oracle. This oracle is prepared for the adversary of Type II.

Type I CL-IND-CCA Adversary. Adversary $\mathcal{A}^I$, without the master key, has accesses to **RequestPubliceKey, RepalcePubliceKey, RequestPartialPrivateKey, Decrypt-I**, with the following restrictions to avoid trivially successful attacks:

- I-1. It cannot make a decryption query with respect to the challenge combination $ID^*, pk^*_{ID^*}, c^*$.

- I-2. It cannot replace the public key for ID^* with the challenge public key $pk^*_{ID^*}$ before c^* has been issued, and later make a partial private key query on the combination $ID^*, pk^*_{ID^*}$ at any point.

Type II CL-IND-CCA Adversary. Adversary $\mathcal{A}^{II}$, with the master key provided in the **Setup** Phase, has accesses to **RequestPubliceKey, ReplacePubliceKey, Decrypt-II**, has the following restrictions in accessing the above oracles:

- II-1 (same to I-1). It cannot make a decryption query with respect to the challenge combination $ID^*, pk^*_{ID^*}, c^*$.

- II-2. It cannot make the oracle $\mathsf{ReplacePublicKey}(ID, pk)$, **unless** pk is the original public key returned by the **RequestPublicKey** oracle for some identity.

2.3. Discussion on Adversary Model

Now, we compare the two kinds of attack models with those used by Al-Riyami and Paterson [11, 13], and explain the merits of our new security formalization.

For Type I adversary, our security model is clearly weaker than that in [11] as follows.

- Unlike the security model in [11], $\mathcal{A}^I$ in our model is not allowed to issue any private key query. At this point, our attacker model is weaker than that in [11].

- Unlike our security model, $\mathcal{A}^I$ in [11] is not forced to offers the current secret value when it make the decryption query. At this point, our attack model is also weaker than that in [11]. Additionally, the Type I attacker in our model makes a decryption query, only when the corresponding public key has already been replaced by himself.

For Type II adversary, the differences between our model and that in [11] are as follows.

- Unlike the security model in [11], $\mathcal{A}^{II}$ in our model is not allowed to issue any private key query. At this point, our attacker model is weaker than that in [11].

- $\mathcal{A}^{II}$ in [11] is not allowed to replace public key at all, while $\mathcal{A}^{II}$ in our model is conditionally allowed to replace the public key. In particular, the condition is that this replaced key value is not the one generated by himself, but the original public key of some other identity generated by the challenger. Of course, this condition guarantees that the attacker does not know the current secret value, after replacing the public key. In the following remark, we show that this difference does not make our Type II adversary model essentially stronger than others.

Remark 1. Here we point out that there still remains the essential common point that (1) $\mathcal{A}^{II}$ in both security models does not know the secret value corresponding to the involved public key and that (2) the attacker still can

not trivially succeed. Hence, this point does not make our attack model essentially more stronger than that in [11]. Concretely speaking, for the security reduction of many existing provably secure CLE schemes such as that in [11], like what will be seen in Section 5, the key point is not whether the challenge identity's public key is replaced or not, but whether the adversary knows the challenge secret value (or the full private key).

At last, we claim that our new attack model is weaker or not essentially stronger than other existing ones such as that in [11]. As a result, on one hand, with the above new security model, we can answer the open problem of formal security proof for Al-Riyami-Paterson conversion. On the other hand, more existing CLE schemes are proved to be suitable for the Al-Riyami-Paterson conversion method, since the lower security of CLE requirement is enough for the generic conversion.

3. Certificate Based Encryption

In this section, we briefly review the definition and security model for CBE from [11].

3.1. Syntax of CBE

Definition 3 [Syntax of CBE]. A Certificate Based Encryption Scheme (CBE) consists of six algorithms as follows.

- $\mathsf{CB.Setup}(1^k) \rightarrow (sk_{CA}, params)$.

  It takes as input the security parameter 1^k and returns the certifier's master key SK_{CA} and the system parameter $params$ that include the description of a string space Λ.

- $\mathsf{CB.SetKeyPair}(params) \rightarrow (pk, sk)$.

  It takes input $params$, and outputs the public key pk and the secret key sk for some entity.

- $\mathsf{CB.Certify}(sk_{CA}, params, \tau, \lambda \in \Lambda, pk) \rightarrow cert'$.

  It takes as input $sk_{CA}, params, \tau, \lambda \in \Lambda, pk$ and outputs the certificate $cert'$, where τ is a string identifying a time period, λ contains other information needed to certify the client such as the client's identifying information and pk is the public key of some identity ID.

- $\mathsf{CB.Consolidate}(params, \tau, \lambda, cert'_\tau, < cert_{\tau-1} >) \rightarrow cert_\tau$.

  It takes input $params, \tau, \lambda, cert'_\tau$ and optionally $cert_{\tau-1}$, and returns $cert_\tau$, the certificate used by the identity ID in time period τ.

- $\mathsf{CB.Encncrypt}(params, \tau, \lambda, pk, m) \rightarrow c$.

  It takes as input $params, \tau, \lambda, pk, m$ where m is the message, and returns a ciphertext c for the message m.

- $\mathsf{CB.Decrypt}(params, cert_\tau, sk, c) \rightarrow m$.

  It takes as input $params, cert_\tau, sk$ and ciphertext c, and outputs the plaintext m.

3.2. Security Definition of CBE

Definition 4.[Security Definition of CBE]. A CBE scheme is CB-IND-CCA secure if no polynomially bounded adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ of Type I or Type II has a non-negligible advantage in the following CBE game.

- Setup Phase: $\mathsf{CB.Setup}(1^k) \rightarrow (params, msk)$.

  Challenger $\mathcal{C}$ takes a security parameter 1^k as input and runs the algorithm $\mathsf{CB.Setup}(1^k)$. It gives $\mathcal{A}$ the system parameter $params$. If $\mathcal{A}$ is of Type I, then $\mathcal{C}$ keeps master secret key msk to itself. Otherwise, it gives $\mathcal{A}$ the master secret key msk and pk^*, where (pk^*, sk^*) are generated with the challenger running the algorithm $\mathsf{CB.SetKeyPair}$.

- Phase 1: $\mathcal{A}_1^{\mathcal{O}}(params) \rightarrow (ID^*, m_0, m_1)$.

  After making a sequence of oracle queries with some restrictions (see below after this game framework), $\mathcal{A}$ terminates by outputting the challenge time period τ^*, certifying information λ^*, two messages of equal length (m_0, m_1). If $\mathcal{A}$ is of Type I, it additionally gives (pk^*, sk^*) to $\mathcal{C}$.

- Challenge Phase:
  $\mathsf{CB.Encrypt}(params, \tau^*, \lambda^*, pk^*, m_b) \rightarrow c^*$.

  The challenger randomly chooses a bit $b \in \{0, 1\}$ and computes the challenge ciphertext

$$c^* = \mathsf{CB.Encrypt}(params, \tau^*, \lambda^*, pk^*, m_b).$$

- Phase 2 : $\mathcal{A}_2^{\mathcal{O}}(c^*) \rightarrow b'$.

  After a sequence of oracle queries with some restrictions (see bellow after this game framework), $\mathcal{A}$ terminates by outputting a guess b' for b. The advantage of $\mathcal{A}$ in this game is defined to be:

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

Oracles for $\mathcal{A}$. In this game, the CL-IND-CCA adversary $\mathcal{A}$ against a CBE scheme may have accesses to the following oracles.

- RequestCertificate(τ, λ, pk, sk). On certification query (τ, λ, pk, sk), $\mathcal{C}$ checks that $\lambda \in \Lambda$ and that (pk, sk) is a valid key-pair. If so, $\mathcal{C}$ responds by running CB.Certify $(sk_{CA}, params, \tau, \lambda, pk)$ to generate $cert'_{\tau}$ for the identity.

- CB-Decrypt $(\tau, \lambda, pk, sk, c)$. On decryption query $(\tau, \lambda, pk, sk, c)$, the challenger $\mathcal{C}$ checks that $\lambda \in \Lambda$ and that (pk, sk) is a valid key-pair. If so, $\mathcal{C}$ runs CB.Certify and CB.Consolidate on input (sk_{CA}, $params$, τ, λ, pk) to obtain $cert_{\tau}$, runs CB.Decrypt on ciphertext c, private key sk, and $cert_{\tau}$, and returns the output to the adversary.

CBE IND-CCA Adversary of Type I. Adversary $\mathcal{A}^I$, without sk_{CA}, is given the access to the certification oracle **RequestCertificate** and the decryption oracle **CB-Decrypt** with restriction that it can not make the decryption oracle $(\tau^*, \lambda^*, pk^*, sk^*, c)$ or the certification query $(\tau^*, \lambda^*, pk^*, sk^*)$.

CBE IND-CCA Adversary of Type II. Adversary $\mathcal{A}^{II}$ has access to sk_{CA}, but does not get to choose a challenge public key to attack. Instead, it is given a specific public key from $\mathcal{C}$ at the beginning of the game. So $\mathcal{A}^{II}$ can compute $cert'_{\tau}$ for any public key pk, given sk_{CA}. The only restriction on $\mathcal{A}^{II}$ is that τ^*, λ^*, c^* can not be the subject of a decryption query after issuing the challenge.

4. Security Proof of Al-Riyami-Paterson Conversion

In this section, we first review Al-Riyami-Paterson generic conversion method from CLE to CBE [11, 13]. Next, we provide its security proof by Theorem 1. At last, we discuss some issues on this security proof.

4.1. Description of Al-Riyami-Paterson Conversion

Let Π^{CL} be a CLE scheme with algorithms CL.Setup, CL.SetSecretValue, CL.SetPublicKey, CL.ExtractPartialPrivateKey, CL.SetPrivateKey, CL.Encrypt, CL.Decrypt as specified in Definition 1. By applying the Al-Riyami-Paterson conversion to Π^{CLE}, we can generically construct a CBE scheme Π^{CB} involving six algorithms CB.Setup, CB.SetKeyPair, CB.Certify, CB.Consolidate, CB.Encncrypt, CB.Decrypt as follows.

- CB.Setup. On input a security parameter k, first run CL.Setup(1^k) to obtain the master key msk and $CL.params$. Then set $sk_{CA} = msk$ and Λ be any subset of $\{0,1\}^*$. Define $CB.params$ by extending $CL.params$ to include a description of Λ.

- CB.SetKeyPair. On input $CB.params$, first extract $CL.params$ from $CB.params$. Then run CL.SetSecretValue$(CL.params) = x$ and CL.Set PublicKey $(CL.params, x) = CL.pk$. The output is $(CB.pk, CB.sk) = (CL.pk, x)$.

- CB.Certify. On input (sk_{CA}, $CB.params$, τ, λ, $CB.pk$), first extract $CL.\ params$ from $CB.params$. Then set the $ID = CB.params||\ \tau||\ \lambda||CB.pk$ and $msk = sk_{CA}$. Next, run CL.ExtractPartialPrivateKey $(CL.params, msk, ID)$=d_{ID}. At last, output is $Cert'_\tau = d_{ID}$.

- CB.Consolidate. On input ($CB.params$, τ, λ, $CB.pk$, $Cert'_\tau$), output $Cert_\tau = Cert'_\tau$.

- CB.Encncrypt. On input $CB.params, \tau, \lambda, CB.pk, m$, extract $CL.params$ from $CB.params$. Then set $ID = CB.params||\tau||\lambda||CB.pk$, and $CL.pk_{ID} = CB.pk$. The output is $c = CL.Encrypt(CL.params,\ ID,\ CL.pk_{ID}, m)$.

- CB.Decrypt. On input ($CB.params$, $cert_\tau$, $CB.sk$, c) in time period τ, first extract $CL.params$ from $CB.params$. Then set $d_{ID} = cert_\tau$ and $x_{ID} = CB.sk$. Next, run CL.SetPrivateKey$(CL.params,\ d_{ID},\ x_{ID})$=$CL.sk_{ID}$. The output is CL.Decrypt$(CL.params,\ CL.sk_{ID},\ c)$.

4.2. Security Proof of Al-Riyami-Paterson Conversion

In [11], Al-Riyami and Paterson proved that Al-Riyami-Paterson conversion from CLE to CBE is secure against the Type-I attackers. And we have

mentioned in Section 2 that Type-I CL-IND-CCA adversary formalized by us is weaker than that in [11]. Hence, we can get that Al-Riyami-Paterson generic conversion remains secure against the Type I adversary model of ours. For paper integrity, we provide the complete security proof.

Lemma 1. Suppose that $\mathcal{A}^I$ is a Type I CB-IND-CCA adversary against Π^{CB} with advantage ϵ in time t. Then there is a Type I CL-IND-CCA adversary $\mathcal{B}^I$ against Π^{CL} with advantage ϵ in time $O(t)$.

Proof. Let $\mathcal{C}$ denote the Π^{CL} challenger against $\mathcal{B}^I$. $\mathcal{B}^I$ mounts a CL-IND-CCA attack on Π^{CL} using help from $\mathcal{A}^I$ as follows.

- **Setup Phase** for CBE game. $\mathcal{B}^I$ obtains from $\mathcal{C}$ the system parameter of Π^{CL} and extends it into the system parameter $CB.params$ of Π^{CB} as done in the description of Π^{CB}. $\mathcal{B}^I$ supplies it to $\mathcal{A}^I$.

- **Phase 1** for CBE game. When $\mathcal{A}^I$ enters the phase 1 for the CBE game, $\mathcal{B}^I$ accordingly enters the phase 1 of the CLE game. For the oracle queries from $\mathcal{A}^I$, $\mathcal{B}^I$ handles these queries as follows.

  - On the certification query (τ, λ, pk, sk) from $\mathcal{A}^I$, the adversary $\mathcal{B}^I$ makes a **ReplacePublicKey** query for the identity $ID = CB.params\ ||\tau||\lambda||pk$, replacing the public key with the value pk. Then $\mathcal{B}^I$ makes a **RequestPartialPrivateKey** query to $\mathcal{C}$ for the identity ID and returns the resulting partial private key to $\mathcal{A}^I$ as the certificate.
  - On decryption query $(\tau, \lambda, pk, sk, c)$, adversary $\mathcal{B}^I$ makes a **ReplacePublicKey** query for the identity $ID = CB.params||\tau||\lambda||pk$, replacing the public key with the value pk. $\mathcal{B}^I$ makes the decryption query $\mathsf{Decrypt\text{-}I}(ID, sk, c)$, and then relays $\mathcal{C}$'s response to $\mathcal{A}^I$.

  When $\mathcal{A}^I$ returns a challenge query $(\tau^*, \lambda^*, pk^*, sk^*, m_0, m_1)$, adversary $\mathcal{B}^I$ makes a **ReplacePublicKey** query for the identity $ID^* = CB.params||\tau^*||\lambda^*||pk^*$, replacing the public key with the value pk^*. Then $\mathcal{B}^I$ sends ID^* and m_0, m_1 to $\mathcal{C}$.

- **Challenge Phase** for CBE game. When the challenger $\mathcal{C}$ responds with a challenge ciphertext c^*, which is the encryption of message m_b (for some bit b) for identity ID^* and public key pk^* for Π^{CL}, $\mathcal{B}^I$ forwards c^* to $\mathcal{A}$ as the response to $\mathcal{A}$'s challenge query.

- **Phase 2** for CBE game. When $\mathcal{A}^I$ enters the phase 2 of the CBE game, $\mathcal{B}^I$ handles the oracle queries as above in the phase 1 of CBE game. Eventually, $\mathcal{A}^I$ should make a guess b' for b. Then $\mathcal{B}^I$ outputs b' as its guess for b to $\mathcal{C}$.

Analysis. We now analyze the behavior of $\mathcal{B}^I$ and $\mathcal{A}^I$ in this simulation. We claim that if algorithm $\mathcal{B}^I$ does not abort during the simulation then algorithm $\mathcal{A}^I$'s view is identical to its view in the real attack. We justify this claim as follows. According to the description of Al-Riyami-Paterson conversion, adversary $\mathcal{B}^I$'s responses to decryption and certification queries are as those seen by $\mathcal{A}^I$ in a real attack, provided that $\mathcal{B}^I$ does not abort. Furthermore, the challenge ciphertext c^* is a valid Π^{CB} encryption of m_b where $b \in \{0,1\}$ is random. Thus, by definition of algorithm $\mathcal{A}^I$, we have that $2|\Pr[b = b'] - \frac{1}{2}| = \epsilon$, if $\mathcal{B}^I$ does not abort.

The probability that $\mathcal{B}^I$ does not abort during the simulation remains to be calculated. In the CBE security definition, $\mathcal{A}^I$ is not allowed to make the decryption query and the certification query relative to the challenge target $ID^*, \tau^*, \lambda^*, pk^*, c^*$. Thus, during the simulation, $\mathcal{B}^I$ is not forced to face the restriction rule I-1, I-2, defined in CL-IND-CCA Adversary of Type I. Now, we can see that $\mathcal{B}^I$ perfectly simulates the environment for $\mathcal{A}^I$.

At last, since what $\mathcal{B}^I$ does in reduction is just issuing some relative queries to $\mathcal{C}$, it is obvious that the time of $\mathcal{B}^I$ is almost equal to the time t of $\mathcal{A}^I$. Hence we say that the running time of $\mathcal{B}$ is $O(t)$. □

Lemma 2. Suppose that $\mathcal{A}^{II}$ is a Type II CB-IND-CCA adversary against Π^{CB} with advantage ϵ and running time t. Then there is a Type II CL-IND-CCA adversary $\mathcal{B}^{II}$ against Π^{CL} with advantage ϵ and $O(t)$.

Proof. Let $\mathcal{C}$ denote a Π^{CL} challenger against $\mathcal{B}^{II}$. $\mathcal{B}^{II}$ mounts a CL-IND-CCA attack on Π^{CL} using help from $\mathcal{A}_{II}$ as follows.

- **Setup Phase** for CBE game. $\mathcal{B}^{II}$ obtains from $\mathcal{C}$ the master key sk_{CA} and the system parameter of Π^{CL} and then extends it into the system parameter Π^{CB} as done in the description of Π^{CB}. $\mathcal{B}^{II}$ randomly selects a valid time period value τ', a valid certifying information λ', and a valid public key pk'. It sets $ID' = CB.params||\tau'||\lambda'||pk'$ and obtains the public key $pk_{ID'}$ by querying the oracle $\mathsf{RequestPublicKey}(ID')$. $\mathcal{B}^{II}$ sets the challenge public key $pk^* = pk_{ID'}$ for $\mathcal{A}^{II}$. $\mathcal{B}^{II}$ passes $CB.params$, sk_{CA} and pk^* to $\mathcal{A}^{II}$.

- **Phase 1** for CBE game. In this phase, when $\mathcal{A}^{II}$ make a decryption query (τ, λ, c), $\mathcal{B}^{II}$ does as follows. It makes sure the validity of λ. It sets $ID = CB.params||\tau||\lambda||pk^*$, and then sequentially makes the two oracle queries $\mathsf{ReplacePublicKey}(ID, pk^*)$ and $\mathsf{Decrypt\text{-}II}(ID, c)$ to obtain the plaintext m. At last, $\mathcal{B}^{II}$ returns m to $\mathcal{A}^{II}$.

- **Challenge Phase** for CBE game. When $\mathcal{A}^{II}$ presents the challenge query $(\tau^*, \lambda^*, m_0, m_1)$, $\mathcal{B}^{II}$ does as follows. $\mathcal{B}^{II}$ sets $ID^* = CB.params||\tau^*||\lambda^*||pk^*$ and then makes the oracle query $\mathsf{ReplacePublicKey}(ID^*, pk^*)$. $\mathcal{B}^{II}$ sends the challenge identity ID^*, m_0, m_1 to its own challenger and obtains the challenge ciphertext c^*. $\mathcal{B}^{II}$ sends c^* to $\mathcal{A}^{II}$.

- **Phase 2** for CBE game. In this phase, $\mathcal{B}^{II}$ handles the decryption queries from $\mathcal{A}^{II}$ as in **Phase 1**, except that the trivial decryption query τ^*, λ^*, c^* will be refused. At last, when $\mathcal{A}^{II}$ returns its guess b', $\mathcal{B}^{II}$ will passes b' to its own challenger.

Analysis. By the description of the above generic construction of Π^{CB}, and especially the decryption algorithm, we can see that $\mathcal{B}^{II}$ correctly simulates the decryption oracle and challenge ciphertext c^*. Additionally, since $\mathcal{A}^{II}$ is prohibited from making the decryption query on (τ^*, λ^*, c^*), $\mathcal{B}^{II}$ is never forced to make the decryption query on (ID^*, pk^*, c^*), which is the restriction rule II-1 in the CLE security model. Furthermore, it is obvious that $\mathcal{B}^{II}$ does not violates the rule II-2 to replace public keys, as defined in the CLE security model. Hence we can see that $\mathcal{B}^{II}$ can perfectly handle the simulation for $\mathcal{A}^{II}$, and then that the advantage of $\mathcal{B}^{II}$ is equal to that of $\mathcal{A}^{II}$.

At last, since what $\mathcal{B}^{II}$ does in reduction is just issuing some relative queries to $\mathcal{C}$, it is obvious that the time of $\mathcal{B}^{II}$ is almost equal to the running time t of $\mathcal{A}^{II}$. Hence we say that the running time of $\mathcal{B}^{II}$ is $O(t)$. □

By Lemma 1 and Lemma 2, we immediately get the following theorem.

Theorem 1. Suppose that Π^{CL} is a CL-IND-CCA secure CLE scheme, and that Π^{CL} is used to build the CBE scheme Π^{CB} as above. Then Π^{CB} is a CB-IND-CCA secure CBE scheme.

4.3. Discussion on Security Proof

Now we point out some issues on the security proof. We will show the reason why the new attack model of Type II makes the security proof succeed, the advantages for applications of Al-Riyami-Paterson conversion, the provable security parameters (standard model and perfect tightness).

- As analyzed by Kang and Park [13], what makes the security proof fail is that (1) $\mathcal{B}^{II}$ has no way to ensure that the original public key of $ID^* = CB.params||\tau^* \ ||\lambda^*||pk^*$ should be pk^*, and (2) by the Type II attack model definition, $\mathcal{B}^{II}$ is not allowed change ID^*'s public key into pk^* through replacing public keys. For the argument in more details, refers to [13]. In contrast, in Definition 2 of ours, we provide $\mathcal{B}^{II}$ the oracle access to replace public keys with a reasonable restriction. In more details, by the Type II attack model formalized in Definition 2, $\mathcal{B}^{II}$ can change ID^*'s public key into pk^* through replacing public keys. It is this point that helps us to solve the open security proof problem.

- Our security proof makes the Al-Riyami-Paterson conversion more conveniently and more generally applicable. In fact, as discussed in Section 2, the new security model of ours is very weak, but strong enough for the security proof. As a result, the weaker security requirement makes the conversion paradigm more generally suitable, and makes it easier to check whether a CLE scheme is suitable for constructing a CBE scheme.

- The security proof is handled in the standard model, without involving any additional random oracles. Hence, given a CLE scheme secure in the standard model for our security definition, then the resulted CBE scheme by Al-Riyami-Paterson conversion is also secure in the standard model.

- The security proof is tight, since the the advantage and running time of the constructed CBE game is almost equal to those of the underlying CLE scheme.

5. Application Example - One Concrete CBE scheme

To demonstrate the application of the Al-Riyami-Paterson conversion, this section describes a concrete CBE scheme from a CLE scheme which was first proposed in [11] and then slightly improved for solving a trivial security flaw in [16]. We start by reviewing the bilinear groups and the related complexity assumption.

5.1. Bilinear Pairing and Complexity Assumption

This section briefly reviews the definition of bilinear pairings and the related complexity assumptions.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q and let P be a generator of $\mathbb{G}_1$, where $\mathbb{G}_1$ is additively represented and $\mathbb{G}_2$ is multiplicatively. A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is said to be a bilinear pairing, if the following three conditions hold: (1)e is bilinear, i.e. $e(aP, bP) = e(P, P)^{ab}$ for all $a, b \in \mathbb{Z}_q^*$; (2) e is non-degenerate, i.e. $e(P, P) \neq 1$, where 1 is the identity of $\mathbb{G}_2$; (3) e is efficiently computable.

Definition 5 [Bilinear Diffie-Hellman Problem,BDH] Given P, aP, bP, cP with uniformly random choices of $a, b, c \in \mathbb{Z}_q$, output $e(P, P)^{abc}$.

An algorithm $\mathcal{A}$ has advantage ϵ in solving the BDH problem, if

$$Pr[\mathcal{A}(P, aP, bP, cP) = e(P, P)^{abc}] = \epsilon].$$

The BDH problem is said to be (t, ϵ)-intractable if there is no algorithm to solve this problem with time less than t and advantage greater than ϵ.

5.2. Concrete CBE Scheme

The scheme described in this section is based on the CLE scheme in [11]. It consists of following algorithms.

- $\mathsf{CB.Setup}(1^k) \to (sk_{CA}, params)$.

  Run by the CA (key generating center). It specifies the hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \to \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \to Z_q^*$, $H_4 : \{0,1\}^n \to \{0,1\}^n$, $H_5 : \mathbb{G}_1 \to \{0,1\}^n$, where n denotes the length of a plaintext. It chooses its master key $sk_{CA} = s$ uniformly at random from $\mathbb{Z}_q$ and computes the its public key $P_{CA} = sP$. The system parameter

  $$params = (q, n, e, \mathbb{G}_1, \mathbb{G}_2, P, P_{CA}, H_1, H_2, H_3, H_4, H_5).$$

- $\mathsf{CB.SetKeyPair}(params) \to (pk_A, sk_A)$.

  It selects a random $sk_A \in Z_q$, computes $pk_A = sk_A P$ and outputs sk_A, pk_A as As secret/public key pair.

- $\mathsf{CB.Certify}(sk_{CA}, params, \tau, \lambda \in \Lambda, pk_A) \to cert'$.

  It sets $ID_A = CB.params|| \tau|| \lambda||pk_A$, computes $Q_A = H_1(ID_A)$, and outputs $cert' = sQ_A$.

- CB.Consolidate$(params, \tau, \lambda, cert'_\tau) \rightarrow cert_\tau$.

  It returns $cert_\tau = cert'_\tau$.

- CB.Encncrypt$(params, \tau, \lambda, pk_A, m) \rightarrow c$.

  It sets $ID_A = CB.params|| \ \tau|| \ \lambda||pk_A$, computes $Q_A = H_1(ID_A)$. It chooses a random $\sigma \in \{0,1\}^n$, set $r = H_3(ID_A, pk_A, \sigma, m)$, and outputs the ciphertext

$$c = (rP, \sigma \oplus H_2(e(Q_A, P_0)^r) \oplus H_5(rP_A), m \oplus H_4(\sigma)).$$

- CB.Decrypt$(params, cert_\tau, sk_A, c) \rightarrow m$.

  Suppose $c = (U, V, W)$, $x_A = sk_A$ and $D_A = cert_\tau$. It first computes $\sigma' = V \oplus H_2(e(D_A, U)) \oplus H_5(x_A U)$ and then $m' = W \oplus H_4(\sigma')$. It sets $r' = H_3(ID_A, pk_A, \sigma', m')$, and tests if $U = r'P$. If not, it rejects the ciphertext. Otherwise, it outputs m' as the decryption.

Theorem 2. Suppose that there is no polynomially bounded algorithm that can solve the BDH problem with non-negligible advantage. Then the above CBE scheme is CB-IND-CCA secure.

Proof. The proof follows Theorem 1 and the fact that the underlying CLE scheme is CL-IND-CCA secure if the BDH problem is hard in G [11, 16]. Here note that the underlying CLE scheme is the improved version in [16] based on that in [11]. Here note that the underlying CLE scheme can be proved secure in their security model [11, 16]. However, as discussed in section 2.3, our new attack model is weaker or not essentially stronger than that in [11, 16]. As a result, the underlying scheme [11, 16] can be easily proved secure in our new security model, by following the security provided in [11, 12]. The underlying CLE scheme will be proved secure in Appendix. □

6. Conclusion

In this paper, we proposed a new security model for CLE (Certificateless Encryption) schemes, featured by conditionally allowing Type II attackers to replace public keys. With help from this improved security definition, we proved that that the Al-Riyami-Paterson generic conversion from CLE to CBE (Certificate Based Encryption) is secure. Namely, if the CLE scheme is secure in our new security model, then the resulting CBE scheme obtained

by using this conversion is also secure. As an example, a concrete provably secure CBE scheme is presented to demonstrate the applicability of our result. The further applications of the new security model for CLE will be explored in the future research.

Acknowledgement. The authors thank Prof. Kenneth G. Paterson for his comments on a preliminary version of this paper.

Reference

[1] Shamir A.(1984) Identity-based cryptosystems and signature schemes. Crypto 1984, LNCS Vol. 196, pp. 47-53, 1984.

[2] Boneh D., Franklin M.(2003) Identity-based encryption from the Weil pairing. SIAM J. Comput., vol. 32(3): 586-615 (2003).

[3] Gentry C.(2003) Certificate-based encryption and the certicate revocation problem. Advances in Cryptology - EUROCRYPT 2003, Lecture Notes in Comput. Sci., vol. 2656, pp. 272-293, 2003.

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

[5] Al-Riyami S. S., Paterson K. G.(2003) Certificateless public key cryptography. in:Advancesin Cryptology-ASIACRYPT 2003, pp.452-473.

[6] Dent A. W.(2008) A survey of certificateless encryption schemes and security models. International journal of information security. 2008, 7(9):349-377

[7] Girault M.(1991) Self-certified public keys. In:AdvancesinCryptology-EUROCRYPT,1991,LNCS 547, pp.490C497.

[8] Yum D.H., Lee P.J.(2004) Identity-based cryptography in public key management. Public Key Infrastructure - EuroPKI 2004, Lecture Notes in Comput. Sci., vol. 3093, pp. 71-84, 2004.

[9] Yum D.H., Lee P.J.(2004) Generic Construction of Certificateless Encryption. In Computational Science and Its Applications - ICCSA 2004 , LNCS vol. 3043, pp. 802-811, Springer-Verlag, 2004

[10] Galindo D., Morillo P., Rfols C.(2006) Breaking Yum and Lee generic constructions of certificateless and certificate-Based encryption schemes. In Proceedings of EuroPKI 2006, LNCS 4043, Springer-Verlag, Berlin Heidelberg New York, 81-91.

[11] Al-Riyami S.S., Paterson K.G.(2005). CBE from CLE: A generic construction and effiient schemes. Public Key Cryptography - PKC 2005, LNCS 3386, pp. 398-415, 2005.

[12] Al-Riyami S.S.(2004) Cryptographic schemes based on elliptic curve pairings, Ph.D. thesis, University of London, 2004, Section 6.5.

[13] Kang B.G., Park J.H.(2005) Is it possible to have CBE from CLE?. eprint.iacr.org/2005/431.ps

[14] Wu W., Mu Y., Susilo W., Huang X., Xu L.(2012) Provably secure construction of certificate-based encryption from certificateless encryption. The computer journal Advance Access published January 5, 2012, doi:10.1093/comjnl/bxr130.

[15] Chatterjee S., Menezes A., Sarkar P. Another Look at Tightness. http://eprint.iacr.org/2011/442

[16] Libert S., Quisquater J.J. On Constructing Certificateless Cryptosystems from Identity Based Encryption. In: PKC 2006, LNCS 3958, pp. 474-490.

Appendix

In [11], Al-Riyami and Paterson proposed a new CLE scheme whose complete security proof is provided in Al-Riyami's dissertation [12]. Here note that this new CLE scheme is denoted by different symbols in [11] and [12], i.e. $\mathsf{FullCLE}^*$ in [11] and $\mathsf{FullCLE}$ in [12] respectively. In [16], Libert and Quisquater pointed out a pretty trivial attack and solved it with a slight modification. In our CBE scheme, we use this improved CLE scheme as the underlying CLE scheme. In this paper, we will use the symbol $\mathsf{FullCLE}$ to denote this improved CLE scheme [11, 16].

First, we review that, trivially, the CLE scheme $\mathsf{FullCLE}$ where its ciphertext

$$C = (rP, \sigma \oplus H_2(e(Q_A, P_0)^r) \oplus H_5(rP_A), M \oplus H_4(\sigma)),$$

where $r = H_3(ID_A, P_A, \sigma, m)$, can been seen combination of the IBE scheme BF-HybridPub where its ciphertext

$$C = (rP, \sigma \oplus H_2(e(Q_A, P_0)^r), M \oplus H_4(\sigma)),$$

and the conventional PKE scheme ELG-Hyrbridpub where the ciphertext

$$C = (rP, \sigma \oplus H_5(rP_A), M \oplus H_4(\sigma)).$$

For the complete description of FullCLE, BF-HybridPub, ELG-Hyrbridpub, and the IND-CCA security definition of public key encryption schemes, refer to [11]. Here we omitted the complicated details.

We will show that the underlying CLE scheme FullCLE is secure in the new security model of ours. In other words, it satisfies the security requirements of Al-Riyami-Paterson conversion and hence naturally induce a secure CBE scheme. Since Type I attack model of ours is weaker than that by Al-Riyami and Paterson in [12, 11] and FullCLE has been proved secure in [12] and improved in [16], Lemma 3 follows.

Lemma 3. If the CLE scheme FullCLE is secure against the Type I CL-IND-CCA adversary as defined in [11, 16], then it also secure against the Type I CL-IND-CCA adversary as defined by Definition 2.

Next, we prove the following lemma.

Lemma 4. Suppose that there exists an efficient Type II IND-CCA adversary $\mathcal{A}_{II}$ against FullCLE with the advantage ϵ and the running time t. Then there is an efficient IND-CCA adversary $\mathcal{B}$ against ELG-HybridPub with the advantage at least $\frac{1}{q}\epsilon$ and the running time $O(t)$.

Proof. Let $\mathcal{A}^{II}$ be a Type II CL-IND-CCA adversary against FullCLE. Suppose $\mathcal{A}^{II}$ has the advantage ϵ. We show how to construct from $\mathcal{A}^{II}$ an IND-CCA adversary $\mathcal{B}$ against the PKE scheme ELG-HybridPub.

Let $\mathcal{C}$ denote the challenger against the IND-CCA adversary $\mathcal{B}$ for the PKE scheme ELG-HybridPub. The challenger $\mathcal{C}$ begins by supplying $\mathcal{B}$ with a public key R and the public parameter $params = (G_1, G_2, \hat{e}, n, P, H_3, H_4, H_5)$. Adversary $\mathcal{B}$ mounts an IND-CCA attack on the key K_{pub} using help from $\mathcal{A}^{II}$ as follows.

Setup Phase of CLE game. $\mathcal{B}$ simulates the Setup phase of the CLE game for $\mathcal{A}^{II}$ by choosing a random $s \in Z_q^*$ as the master key, setting $P_0 = sP$ as the system public key and supplying $\mathcal{A}^{II}$ with $params = (G_1, G_2, \hat{e}, n, P, P_0, H_1, H_2, H_3, H_4, H_5)$, and the master key s. Here, H_1 and

H_2 are additional random oracles. Additionally, $\mathcal{B}$ chooses an index I with $1 \leq I \leq q_1$, where it is assumed that $\mathcal{A}^{II}$ makes q_1 **RequestPublicKey** queries. These two random oracles are handled as follows.

H_1 **queries:** Adversary $\mathcal{B}$ simulates and answers H_1 queries by maintaining a list of queries and replies.

H_2 **queries:** Adversary $\mathcal{B}$ simulates and answers H_2 queries by maintaining a list of queries and replies.

Phase 1 of of CLE game. In this phase, $\mathcal{B}$ handles the oracle requests from $\mathcal{A}^{II}$ as follows.

- On the i-th **RequestPublicKey** query, if $i = I$, $\mathcal{B}$ just returns R; otherwise, it randomly generates the secret value $x_i \in Z_q$ and the public key $pk_i = x_i P$ and adds the entry (x_i, pk_i) to the initially empty list L_{pk}.

- On the query **ReplacePublicKey**(ID, pk), if pk is some other identity's original public key returned from the **RequestPublicKey** oracle, then $\mathcal{B}$ sets pk as the public key value of ID; otherwise, it refuse this request.

- On the the decryption query **Decrypt-II** (ID, c) with $c = (U, V, W)$, $\mathcal{B}$ computes $\xi = e(U, sH_1(ID))$. If the current public key $pk = pk^*$, $\mathcal{B}$ relays the decryption query $(U, V \oplus H_2(\xi), W)$ to $\mathcal{C}$. If pk is the list L_{pk} and the corresponding secret value is x, $\mathcal{B}$ can perform this decryption himself using the partial private key $sH_1(ID)$ and the secret value x. The FullCLE decryption of (U, V, W) for the identity ID is equal to the ELG-HybridPub decryption of $(U, V \oplus H_2(\xi), W)$ under the private key corresponding to K_{pub}. Hence, $\mathcal{B}$ rightly simulates the decryption oracle.

At the end of this phase, $\mathcal{A}^{II}$ picks ID^* and two messages m_0, m_1 on which it wants to be challenged, and sends them to $\mathcal{B}$.

Challenge Phase of of CLE game. If the current public key of ID^* is not equal to R, $\mathcal{B}$ will abort. Otherwise, Algorithm $\mathcal{B}$ makes the challenge ciphertext as follows. $\mathcal{B}$ gives $\mathcal{C}$ the pair (m_0, m_1) as the messages on which it wishes to be challenged. $\mathcal{C}$ responds with the challenge ciphertext $c' = (U', V', W')$, such that c' is the ELG-HybridPub encryption of m_b under K_{pub} for a random $b \in \{0, 1\}$. Then $\mathcal{B}$ computes $\xi' = \hat{e}(U', sH_1(ID^*))$ and sets $c^* = (U', V' \oplus H_2(\xi'), W')$,and delivers c^* to $\mathcal{A}^{II}$. It is not hard to see that C^* is the FullCLE encryption of m_b for identifier ID^* (with public key R).

Phase 2 of of CLE game. Adversary $\mathcal{B}$ continues to respond to requests in the same way as it did in Phase 1, except that the trivial decryption query with respect to (ID^*, pk^*, c^*) will be refused. Eventually, $\mathcal{A}_{II}$ will make a guess b' for b. B outputs b' as its guess for b.

Analysis. In the above simulation, the only event making $\mathcal{B}$ to fail is that the challenge public key pk^* is not equal to R. This event happens with probability $\frac{1}{q_1}$. If pk^* is equal R, it is easy to see that $\mathcal{B}$ will succeed with probability at least ϵ. Thus, by the above simulation, $\mathcal{B}$ succeeds with probability at least $\frac{1}{q_1}\epsilon$. since what $\mathcal{B}$ does in the above reduction is just issuing some relative queries to $\mathcal{C}$, it is obvious that the time of $\mathcal{B}^I$ is almost equal to the running time t of $\mathcal{A}^I$. Hence we say that the running time of $\mathcal{B}$ is $O(t)$. □

By Lemma 3 and Lemma 4, we have:

Theorem 3. FullCLE is secure against the Type-II CL-IND-CCA adversary as defined Definition 2 under the BDH assumption. Thus it can be used to construct a secure CBE scheme through Al-Riyami-Paterson conversion.