Further Observations on Certificate-Base Encryption and its Generic Construction from Certificateless Public Key Encryption*

Yang Lu

College of Computer and Information Engineering, Hohai University
210098 Nanjing, Jiangsu Province, China
luyangnsd@163.com

Abstract. Certificate-based encryption (CBE) is a new asymmetric encryption paradigm which was introduced to solve the certificate management problem in traditional public key encryption (PKI). It combines PKE and identity-based encryption (IBE) while preserving some of their most attractive features. CBE provides an efficient implicit certificate mechanism which eliminates the third-party queries and simplifies the certificate revocation problem in the traditional PKI. It also solves the key escrow problem and key distribution problem inherent in IBE. In this paper, we introduce the key replacement attack and the malicious-but-passive certifier attack into CBE, and define a class of new security models for CBE under different security levels according to the power of the adversaries against CBE. Our new security models are more elaborated and stronger compared with other existing ones. Then, we propose a generic construction of CBE from certificateless public key encryption and prove its security under the proposed security models in the standard model. We also show a concrete conversion using the proposed generic construction.

Keywords: certificate-based encryption, security model, generic construction, certificateless public key encryption, standard model.

1. Introduction

In traditional public key cryptography (PKC), cryptographic keys are generated randomly with no connection to users' identities. Therefore, it is infeasible to prove that a party is indeed the holder of a given public key. This problem can be solved by introducing public key certificates generated by a trusted third party called the Certification Authority (CA) that can provide an unforgeable and trusted link between a public key and the identity of its holder. This kind of certificate systems is referred to as the Public key Infrastructure (PKI). However, the need for PKI-supporting certificates is

* This paper is supported by the National Natural Science Foundation of China (No. 60903018) and the Fundamental Research Funds for the Central Universities (No. 2010B06414).

considered as the main difficulty in the deployment and management of traditional PKC. To simplify the management of the public key certificates, Shamir [1] introduced the concept of identity-based cryptography (IBC) in which the public key of each user is derived directly from its identity, such as an IP address or an e-mail address, and the corresponding private key is generated by a trusted third party called Private Key Generator (PKG). Rather than obtaining the disparate public keys and the certificates of its intended recipients separately as is done in traditional PKC, a message sender who knows the identities of its recipients needs only to obtain the public parameters of the PKG. Therefore, the main practical benefit of IBC lies in great reduction of need for public key certificates. However, the PKG can generate the private keys of all its users, so private key escrow becomes an inherent problem in IBC. Moreover, private keys must be sent to the users over secure channels. It makes private key distribution a daunting task.

To fill the gap between traditional PKC and IBC, Al-Riyami and Paterson [2] proposed a new paradigm called certificateless public key cryptography (CL-PKC) in 2003. In CL-PKC, a trusted third party called Key Generation Center (KGC) is involved in the process of issuing a partial secret key for each user. The user independently generates its public/private key pair and combines the partial secret key from KGC with its private key to generate the final decryption key. This way, KGC does not know the decryption key of any user. Therefore, CL-PKC solves the key escrow problem inherent in IBC. However, due to the lack of public key certificate to ensure the authenticity of the user's public key, it is important to assume that an adversary in the certificateless system can replace the user's public key with a false key of its choice, which is also known as key replacement attack. Cryptographic protocols in certificateless system are easily suffered from this kind of attack. Moreover, partial secret keys must be sent to the users over secure channels. It makes CL-PKC suffer the same key distribution problem as IBC.

In Eurocrypt 2003, Gentry [3] introduced the notion of certificate-based encryption (CBE), which combines identity-based encryption (IBE) and traditional PKI-supporting public key encryption (PKE) while preserving some of their most attractive features. CBE provides an implicit certificate mechanism and allows a periodical update of certificate status. As in the traditional PKE, each user generates his own public/private key pair and requests a certificate from a trusted third party, which is called as the certifier. The certifier generates a certificate as in a traditional PKI and is responsible for pushing a fresh certificate only to the holder of the public key at beginning of each time period. A certificate in CBE has all the functionalities of a traditional PKI certificate, and also acts as a partial decryption key. This additional functionality provides an implicit certificate mechanism so that the sender is not required to obtain fresh information on certificate status and the recipient can only decrypt the ciphertext using his private key along with an up-to-date certificate from its certifier. The feature of implicit certificate allows us to eliminate third-party queries for the certificate status and to simplify the public key revocation problem so that CBE does not need infrastructures like CRL and OCSP. Therefore, CBE can be used to construct a more efficient

PKI requiring fewer infrastructures. Furthermore, there is no key escrow problem (since the certifier does not know the private keys of users) and key distribution problem (since the certificates need not be kept secret) in CBE.

1.1. Related Work

In the original work [3], Gentry constructed a CBE scheme in the random oracle [4] from the BF-IBE scheme [5]. A subsequent paper by Yum and Lee [6] provided a formal equivalence theorem among IBE, certificateless public key encryption (CL-PKE) [2] and CBE, and showed that IBE implies both CBE and CL-PKE by giving a generic construction from IBE to those primitives. However, Galindo et al. [7] pointed out that a dishonest authority could break the security of their generic constructions. Actually, these generic constructions were inherently flawed due to a naive use of double encryption without further treatments. In [8], Lu et al. solved this problem by using the Fujisaki-Okamoto conversions [9, 10] and gave a method to achieve generic CCA-secure CBE constructions in the random oracle model. Lu et al. also proposed two generic constructions of CBE without random oracles in [11]. In 2005, Al-Riyami and Paterson [12] gave an analysis of Gentry's CBE concept and repaired a number of problems in the original definitions for CBE. They also presented a generic conversion from CL-PKE to CBE and claimed that a secure CBE scheme could be constructed from any secure CL-PKE scheme using this conversion. Kang and Park [13] pointed out that their conversion was incorrect due to the flaw in their security proof. In [14], Yum and Lee proposed a separable implicit certificate revocation system called status CBE to relieve the certifier's burden of certificate revocation, in which the authenticity of a public key is guaranteed by a long-lived certificate and the certificate revocation problem is resolved by a short-lived certificate. However, their status CBE scheme is pointed out by Park and Lee [15] to be insecure under the key replacement attack. In 2006, Morillo and Ràfols [16] proposed the first CBE scheme in the standard model from the Waters-IBE scheme [17] and the BB-IBE scheme [18]. In 2008, Galindo et al. [19] revised the CBE scheme in [16] and proposed an improved scheme. Liu and Zhou [20] also proposed another CBE scheme in the standard model from the Gentry-IBE scheme [21]. In 2009, Lu et al. [22] proposed a quite efficient CBE scheme in the random oracle model from the SK-IBE scheme [23, 24], which requires computing only one pairing in the decryption algorithm.

1.2. Our Contributions

The contributions of this paper are twofold. The first contribution is that we provide more reasonable and elaborated security models for CBE. Although Al-Riyami and Paterson [12] have repaired a number of problems in the original definition of the security model for CBE and proposed a revised one,

the new definition is still not satisfactory. Inspired by the definitions of security models for CL-PKE [25] and CBS [26], we introduce the key replacement attack and the malicious-but-passive certifier attack into CBE, and define a class of new security models for CBE. We also divide these security models into different security levels according to the power of the adversaries against CBE so that our definitions will provide a systematic approach for analyzing the exiting CBE schemes and constructing new CBE schemes. The second contribution is that we make a further investigation on the relationship between CBE and CL-PKE. As discussed in [12], CBE and CL-PKE are two similar concepts, and also share some common features. In [6], Yum and Lee showed that CBE and CL-PKE can be constructed from two IBE schemes. So CBE and CL-PKE can be constructed from each other via two intermediate IBE schemes. However, the direct conversion from CL-PKE to CBE still remains open. In this paper, we resolve this open problem by proposing a new generic construction of CBE from CL-PKE in the standard model.

2. Definition of Certificate-Based Encryption

In a CBE scheme, a certificate generator, which is called as the certifier, will first generate the system parameter including a master key and a list of public system parameters. The certifier will use the system parameter to generate certificates for users in the system. Users then will generate their own public/private key pairs and contact the certifier to obtain the corresponding certificates. A user should use its private key and the certificate from the certifier as the decryption key to decrypt the ciphertext received. The following definition of CBE is modified from [12], where the original definition given by Gentry in [3] was reconsidered.

Definition 1. A CBE scheme is a 5-tuple of polynomial time algorithms (CB-Setup, CB-SetKeyPair, CB-Certify, CB-Encrypt, CB-Decrypt) such that:

- CB-Setup is a probabilistic algorithm run by a certifier that takes a security parameter k and a total number of time periods N as input, and outputs a master key *CB-msk* and a list of public parameters *CB-params* that include the descriptions of a finite identity information space $IDSPC^{CB}$, a finite plaintext space $MSPC^{CB}$ and a finite ciphertext space $CSPC^{CB}$.
- CB-SetKeyPair is a probabilistic algorithm run by a user that takes the public parameters *CB-params* as input, and outputs a public/private key pair (*CB-PK*, *CB-SK*).
- CB-Certify is a deterministic or probabilistic algorithm run by a certifier that takes the public parameters *CB-params*, the master key *CB-msk*, an index $\tau \in [0, N-1)$ of the current time period, an identity $id \in IDSPC^{CB}$ and a public key *CB-PK* as input, and outputs a certificate $CB\text{-}Cert_\tau$ which is sent to the user with identity *id* through an open channel.

- CB-Encrypt is a probabilistic algorithm that takes the public parameters *CB-params*, an index $\tau \in [0, N-1)$ of the current time period, an identity $id \in IDSPC^{CB}$, a public key *CB-PK* and a plaintext $M \in MSPC^{CB}$ as input, and outputs a ciphertext $C \in CSPC^{CB}$.
- CB-Decrypt is a deterministic algorithm that takes the public parameters *CB-params*, a private key *CB-SK*, a certificate $CB\text{-}Cert_\tau$ and a ciphertext *C* as input, and outputs either a message $M \in MSPC^{CB}$ or a special symbol $\perp$ indicating a decryption failure.

Correctness. It is required that CB-Decrypt(*CB-params*, *CB-SK*, $CB\text{-}Cert_\tau$, CB-Encrypt(*CB-params*, τ, *id*, *CB-PK*, *M*)) = *M* for any $M \in MSPC^{CB}$, where (*CB-PK*, *CB-SK*) is a valid public/private key pair generated by CB-SetKeyPair on input <*CB-params*> and $CB\text{-}Cert_\tau$ is a valid certificate generated by CB-Certify on input <*CB-params*, *CB-msk*, τ, *id*, *CB-PK*>.

Remark 1. In [12], the definition of CBE includes a certificate consolidation algorithm *CB-Consolidate* which is run by each user to take <*CB-params*, τ, *id*, $CB\text{-}Cert_\tau$> and optionally $CB\text{-}Cert'_{\tau-1}$ as input and to generate the final certificate $CB\text{-}Cert'_\tau$ used by the user *id* in the time period τ. However, we note that a concrete CBE scheme need not involve certificate consolidation. In this situation, the algorithm *CB-Consolidate* will simply output $CB\text{-}Cert'_\tau = CB\text{-}Cert_\tau$. Since this algorithm is not used in almost all the existing CBE schemes, we also omit this algorithm in this paper.

3. Security Models for Certificate-Based Encryption

Roughly speaking, the security of a CBE scheme requires that a user with the identity *id* can decrypt a valid ciphertext generated in the time period τ under the public key *CB-PK* if and only he has the correct *CB-SK* and $CB\text{-}Cert_\tau$. In other words, he cannot recover the plaintext from a valid ciphertext correctly with only *CB-SK* or $CB\text{-}Cert_\tau$.

In [3] and [12], the security models for CBE are both defined by two types of adversaries: Type-I adversary and Type-II adversary, where Type-I adversary models an uncertified client who has not the legitimate certificate and Type-II adversary models a malicious certifier in possession of the master secret key. Different from the original security model in [3] where the challenger against Type-II adversary is allowed to work with multiple values of the public system parameters, the security model in [12] requires that the public parameters and master key are fixed and supplied to Type-II adversary at the beginning of the simulation. Kang and Park [13] pointed out that this restriction is sufficiently reasonable because a certifier does not change its public parameters frequently in practice. However, both these two security models may be not elaborated and strong enough for the practical applications. For example, these two security models both require that Type-I

adversary should provide a private key along with the corresponding public key in all of decryption oracle queries. This restriction enables the challenger to handle these decryption queries, but is unnecessary and also restricts the ability of Type-I adversary. Actually, the challenger can handle decryption queries using some special purpose knowledge extractors without requiring the adversary to provide the private key. Besides this, both these two security models do not consider the key replacement attack. It seems that the key replacement attack does not exist in CBE due to the use of certificates. However, in CBE only the owner needs to check the validity of its certificate and other users do not need. Therefore, such attack actually does exist. A concrete example is the status CBE scheme proposed by Yum and Lee [14]. In [15], this scheme is pointed out to be insecure under the key replacement attack. Since a reasonable and elaborated security model is indispensable to the construction of provably secure cryptographic schemes, we should define a more reasonable and elaborated security model for CBE. Inspired by the improvements in the definitions of security notions for CL-PKE [25] and CBS [26], we define a class of new security models for CBE under different security levels according to the power of the adversaries against CBE. Our definitions abolish the unnecessary restrictions in the existing security models, and also introduce the key replacement attack and the malicious-but-passive certifier attack. In the following, we give the concrete definitions of these security models and also investigate the relationships among them.

3.1. Oracles

We first define the oracles that an adversary against CBE may access and how each oracle query should be responded by a challenger $\mathcal{C}$. We assume that $\mathcal{C}$ keeps a history of "query-answer" while interacting with the adversary.

- CB-RequestPublicKey: On input an identity *id*, the challenger $\mathcal{C}$ responds with the public key *CB-PK* for *id*. If the identity *id* has no associated public key, then $\mathcal{C}$ generates a public key *CB-PK* for *id* by running CB-SetKeyPair.
- CB-ReplacePublicKey: The adversary can repeatedly replace the public key of any entity with any value of its choice. On input an identity *id* and a value *CB-PK'*, the challenger $\mathcal{C}$ replaces the current public key *CB-PK* with *CB-PK'*. Note that the current value of a user's public key is used by $\mathcal{C}$ in any computations or responses to the adversary's requests. This oracle models the adversary's ability to convince a legitimate user to use an invalid public key and enables our security models to capture the public key replacement attack.
- CB-ExtractPrivateKey: On input an identity *id*, the challenger $\mathcal{C}$ responds with the private key *CB-SK* for *id*. If the identity *id* has no associated private key, then $\mathcal{C}$ generates a private key *CB-SK* for *id* by running the algorithm

CB-SetKeyPair. However, it is unreasonable to expect $\mathcal{C}$ to be able to respond to such a query if the public key *CB-PK* for *id* has already been replaced.

- CB-RequestCertificate: On input an index τ of a time period and an identity *id*, the challenger $\mathcal{C}$ responds with a certificate $CB\text{-}Cert_\tau$ for *id* in the time period τ. If the identity *id* has no associated certificate in the time period τ, then $\mathcal{C}$ generates $CB\text{-}Cert_\tau$ by running CB-Certify.
- CB-Decrypt: Considering the different levels of the decrypting power the challenger $\mathcal{C}$ may have, the decryption oracle can be divided into following three types:
  - CB-StrongDecrypt: On input an index τ of a time period, an identity *id*, and a ciphertext *C*, the challenger responds with the correct decryption of *C*, even if the public key for *id* has been replaced. This is a rather strong property for the security model of CBE. After all, the challenger may no longer know the correct corresponding private key. However, this capability may give the adversary more power in breaking the scheme. For further discussion of this feature (but in CL-PKE setting), see [2].
  - CB-NormalDecrypt: On input an index τ of a time period, an identity *id*, and a ciphertext *C*, the challenger $\mathcal{C}$ responds with the decryption of the ciphertext *C* using the original private key for *id* and the certificate for *id* in the time period τ. Note that the functionality of this oracle can be achieved by a strong decryption oracle.
  - CB-WeakDecrypt: On input an index τ of a time period, an identity *id*, a private key *CB-SK* and a ciphertext *C*, the challenger $\mathcal{C}$ responds with the decryption of the ciphertext *C* using *CB-SK* and the certificate for *id* in the time period τ. Note that the functionality of such an oracle also can be achieved by a strong decryption oracle.

3.2. Type-I Security

The Type-I security model of CBE is designed to protect against an uncertified user who dose not obtain a legitimate certificate from its certifier and is trying to gain some information about a message from its encryption. According to the attack power of such an adversary against CBE, we classify Type-I security into three levels: weak Type-I (wType-I) security, normal Type-I (nType-I) security and strong Type-I (sType-I) security.

Weak Type-I Security. We first define the wType-I security model for CBE. This security notion is defined by a following weak IND-CB-CCA2 Game-I in which Type-I adversary $\mathcal{A}_I$ can not replace public keys of any users and make the strong decryption queries, but may request public keys and certificates, extract private keys and make normal or weak decryption queries:

- Setup: The challenger $\mathcal{C}$ runs the algorithm CB-Setup(1^k, N) to generate a master key *CB-msk* and a list of public system parameters *CB-params*. It outputs *CB-params* to $\mathcal{A}_I$.
- Phase 1: Upon receiving *CB-params*, $\mathcal{A}_I$ queries the oracles CB-RequestPublicKey, CB-RequestCertificate, CB-ExtractPrivateKey, and CB-NormalDecrypt or CB-WeakDecrypt in an adaptive manner.
- Challenge: Once $\mathcal{A}_I$ decides that Phase 1 is over, it outputs an index τ^* of a time period, an identity id^* and two equal length messages M_0, M_1, on which it wants to be challenged. The challenger $\mathcal{C}$ randomly chooses a bit $b \in \{0, 1\}$, computes C^* = CB-Encrypt(*CB-params*, τ^*, id^*, *CB-PK**, M_b), and then outputs C^* as the challenge ciphertext to $\mathcal{A}_I$.
- Phase 2: $\mathcal{A}_I$ issues a second sequence of queries as in Phase 1.
- Guess: Finally, $\mathcal{A}_I$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$. The restrictions are that: (1) $\mathcal{A}_I$ cannot query RequestCertificate(τ^*, id^*); (2) $\mathcal{A}_I$ cannot query CB-NormalDecrypt(τ^*, id^*, C^*) or CB-WeakDecrypt(τ^*, id^*, *CB-SK**, C^*); (3) $\mathcal{A}_I$ cannot query the oracle CB-StrongDecrypt. The advantage of $\mathcal{A}_I$ is defined to be $\mathrm{Adv}(\mathcal{A}_I) = |\Pr[b = b'] - \frac{1}{2}|$.

Definition 2. A CBE scheme is said to be wType-I secure if no probabilistic and polynomial-time adversary can have non-negligible advantage in winning the weak IND-CB-CCA2 Game-I.

Normal Type-I Security. Different from the wType-I security model, the nType-I security model gives Type-I adversary to the ability to replace the public keys of any users with values of its choice. However, it also prevents the adversary from querying the strong decryption oracle. This kind of security is defined by a normal IND-CB-CCA2 Game-I which is very similar to the weak IND-CB-CCA2 Game-I, but with the following two differences:

- $\mathcal{A}_I$ can query CB-ReplacePublicKey on any identity;
- $\mathcal{A}_I$ cannot query CB-ExtractPrivateKey on any identity if the corresponding public key has been replaced;

Definition 3. A CBE scheme is said to be nType-I secure if no probabilistic and polynomial-time adversary can have non-negligible advantage in winning the normal IND-CB-CCA2 Game-I.

Strong Type-I Security. Finally, we define the strongest Type-I security notion for CBE, namely the sType-I security. In this kind of security model, the

adversary is allowed to query the strong decryption oracle. That is, the adversary is able to obtain the correct decryption of any ciphertext under the public key chosen by itself without providing the corresponding private key. The sType-I security is defined by a strong IND-CB-CCA2 Game-I which is very similar to the normal IND-CB-CCA2 Game-I, but with the following two differences:

- $\mathcal{A}_I$ can query the oracle CB-StrongDecrypt rather than CB-NormalDecrypt and CB-WeakDecrypt;
- $\mathcal{A}_I$ cannot query CB-StrongDecrypt(id^*, τ^*, C^*).

Definition 4. A CBE scheme is said to be sType-I secure if no probabilistic and polynomial-time adversary can have non-negligible advantage in winning the strong IND-CB-CCA2 Game-I.

3.3. Type-II Security

The Type-II security model for CBE is designed to protect against an honest-but-curious certifier who always generates its master key and the public system parameters honestly according to the scheme specification. Hence, a Type-II adversary in this security model is equipped with the master key and needs not to access the oracle RequestCertificate, as it is able to compute these values by itself. As the Type-I security, the Type-II security also can be classified into three levels: weak Type-II (wType-II) security, normal Type-II (nType-II) security and strong Type-II (sType-II) security.

Weak Type-II Security. The wType-II security model for CBE is defined by a following weak IND-CB-CCA2 Game-II in which Type-II adversary $\mathcal{A}_{II}$ can not replace any user's public key, but may request public keys, extract private keys and make normal decryption queries.

- Setup: The challenger $\mathcal{C}$ runs the algorithm CB-Setup(1^k, N) to generate a master key *CB-msk* and a list of public system parameters *CB-params*. It outputs *CB-msk* and *CB-params* to $\mathcal{A}_{II}$.
- Phase 1: Upon receiving *CB-msk* and *CB-params*, $\mathcal{A}_{II}$ starts issuing queries to the oracles CB-RequestPublicKey, CB-ExtractPrivateKey, and CB-NormalDecrypt.
- Challenge: Once $\mathcal{A}_{II}$ decides the Phase 1 is over, it outputs an index τ^* of a time period, an identity id^* and two equal length messages M_0, M_1, on which it wants to be challenged. The challenger $\mathcal{C}$ randomly chooses a bit $b \in \{0, 1\}$, computes C^* = CB-Encrypt(*CB-params*, τ^*, id^*, *CB-PK**, M_b), and then outputs C^* as the challenge ciphertext to $\mathcal{A}_{II}$.

- Phase 2: $\mathcal{A}_{II}$ issues a second sequence of queries as in Phase 1.
- Guess: Finally, $\mathcal{A}_{II}$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$. The restrictions are that: (1) $\mathcal{A}_{II}$ cannot query CB-ExtractPrivateKey(id^*); (2) $\mathcal{A}_{II}$ cannot query CB-NormalDecrypt(id^*, τ^*, C^*). The advantage of $\mathcal{A}_{II}$ in this game is defined to be $Adv(\mathcal{A}_{II}) = |Pr[b = b'] - \frac{1}{2}|$.

Definition 5. A CBE scheme is wType-II secure if no probabilistic and polynomial-time adversary can have non-negligible advantage in winning the above weak IND-CB-CCA2 Game-II.

Remark 2. Our definition of the wType-II security for CBE is very similar to the definition of Type-II security for CBE in [12]. The only difference is that the Type-II adversary in our definition is allowed to work with multiple public keys and to select any one of them for the challenge, while such type of adversary in [12] is given only a specific public key by the challenger at the beginning of the game.

Normal Type-II Security. Different from the wType-II security model, the nType-II security model gives Type-II adversary to the ability to replace the public keys of any users with values of its choice. But it also prevents the adversary from querying the strong decryption oracle. This kind of security model is defined by a normal IND-CB-CCA2 Game-II which is very similar to the weak IND-CB-CCA2 Game-II, but with the following two differences:

- $\mathcal{A}_{II}$ cannot query CB-ExtractPrivateKey on any identity if the corresponding public key has been replaced;
- $\mathcal{A}_{II}$ cannot be challenged on an identity for which it has replaced the public key.

Definition 6. A CBE scheme is said to be nType-II secure if no probabilistic and polynomial-time adversary can have non-negligible advantage in winning the normal IND-CB-CCA2 Game-II.

Strong Type-II Security. In the nType-I security model, if Type-II adversary is allowed to query the strong decryption oracle, then we will obtain the sType-II security notion for CBE. The sType-II security is defined by a strong IND-CB-CCA2 Game-II which is very similar to the normal IND-CB-CCA2 Game-II, but with the following two differences:

- $\mathcal{A}_{II}$ can query CB-StrongDecrypt rather than CB-NormalDecrypt and CB-WeakDecrypt;

– $\mathcal{A}_{II}$ cannot query CB-StrongDecrypt(id^*, τ^*, C^*).

Definition 7. A CBE scheme is said to be sType-II secure if no probabilistic and polynomial-time adversary can have non-negligible advantage in winning the strong IND-CB-CCA2 Game-II.

3.4. Malicious-but-passive Type-II Security

We now define a much stronger Type-II security model for CBE, namely the malicious-but-passive Type-II (mType-II) security model. This kind of model is designed to protect against a malicious-but-passive certifier who may generate its master key and the public system parameters maliciously at the setup stage of the system, instead of generating its master key and the public system parameters honestly according to the scheme specification and suddenly becoming malicious as the honest-but-curious certifier in the Type-II security model. So an adversary in this security model controls the generation of the master key and the public system parameters, and that of any user's certificate. The malicious-but-passive attack by the trusted third party was first introduced to the security of CL-PKC by Au et al. [27], in which they showed that the malicious-but-passive KGC in some certificateless schemes like [2] can generate its master key and the public system parameters maliciously so that it can decrypt all the ciphertext in the system without knowing the users' private key.

The general mType-II security model for CBE is expressed by the following malicious-but-passive IND-CB-CCA2 Game-II:

– Setup: The challenger $\mathcal{C}$ invokes a malicious-but-passive Type-II adversary $\mathcal{A}_{II}$ on input 1^k and N. $\mathcal{A}_{II}$ returns a list of public system parameters *CB-params* to $\mathcal{C}$. It is required that *CB-params* is computationally indistinguishable from the output of CB-Setup(1^k, N). At this stage, $\mathcal{A}_{II}$ is not allowed to query any oracle.[1]
– Phase 1: In this phase, $\mathcal{A}_{II}$ may have access to some certain oracles according to its attack power.
– Challenge: Once $\mathcal{A}_{II}$ decides the Phase 1 is over, it outputs an index τ^* of a time period, an identity id^* and two equal length messages M_0, M_1, on which it wants to be challenged. The challenger $\mathcal{C}$ randomly chooses a bit $b \in \{0, 1\}$, computes C^* = CB-Encrypt(*CB-params*, τ^*, id^*, *CB-PK**, M_b), and then outputs C^* as the challenge ciphertext to $\mathcal{A}_{II}$.
– Phase 2: $\mathcal{A}_{II}$ issues a second sequence of queries as in Phase 1.

[1] One exception is that if the security analysis is done under the random oracle model, then such an adversary can query the specified random oracles.

– Guess: Finally, $\mathcal{A}_{II}$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

The advantage of $\mathcal{A}_{II}$ is defined to be $Adv(\mathcal{A}_{II}) = |Pr[b = b'] - \frac{1}{2}|$.

As the Type-II security for CBE, we also can define three different levels of mType-II security: weak mType-II (wmType-II) security, normal mType-II (nmType-II) security and strong mType-II (smType-II) security. Since these security notions can be defined in the same way as the Type-II security, we omit the concrete definitions here.

3.5. Relation among Security Models for CBE

We now study the relation among the above different security models for CBE. Firstly, according the attack power of the adversaries in each security model, it is not difficult to deduce the following relations:

– sType-I $\Rightarrow$ nType-I $\Rightarrow$ wType-I
– sType-II $\Rightarrow$ nType-II $\Rightarrow$ wType-II
– smType-II $\Rightarrow$ nmType-II $\wedge$ sType-II
– nmType-II $\Rightarrow$ wmType-II $\wedge$ nType-II
– wmType-II $\Rightarrow$ wType-II.

In the above, $A \Rightarrow B$ denotes that a CBE scheme which is A secure must be B secure, and $A \Rightarrow B \wedge C$ denotes that a CBE scheme which is A secure must both be B secure and C secure. It is clear that the sType-I security and the smType-II security are the strongest security levels that a CBE scheme could achieve.

We note that all the existing CBE schemes are proved secure using the common observational or black-box proof technique which requires that an algorithm (also called a solver) should use an attacker as a subroutine in solving a mathematical problem. However, the following two theorems state that the black-box security proof technique may not be used to prove a CBE scheme to both be sType-I secure and smType-II (or sType-II) secure in the standard model.

Theorem 1. In the standard model, if there exists a black-box proof for the sType-I security of a CBE scheme, then that CBE scheme must not be nmType-II secure.

Proof. Assume that there exists a CBE scheme which is sType-I secure. Then, there exists a PPT challenger $\mathcal{C}_I$ for the strong IND-CB-CCA2 Game-I such that $\mathcal{C}_I$ successfully simulates the strong IND-CB-CCA2 Game-I with overwhelming probability and no sType-I adversaies win the game with non-negligible advantage. According to Definition 4 in Section 3.2, $\mathcal{C}_I$ provides the

following oracles: CB-RequestPublicKey, CB-RequestCertificate, CB-ReplacePublicKey, CB-ExtractPrivateKey, and CB-StrongDecrypt. We show how to construct a PPT nmType-II adversary $\mathcal{A}_{II}$ to win the normal and malicious-but-passive IND-CB-CCA2 Game-II with non-negligible advantage by interacting with $\mathcal{C}_I$ as follows:

- At the beginning of the normal and malicious-but-passive IND-CB-CCA2 Game-II, challenger $\mathcal{C}_{II}$ invokes $\mathcal{A}_{II}$ on input 1^k and N. $\mathcal{A}_{II}$ invokes $\mathcal{C}_I$ on input 1^k and N to get *CB-params*, and then returns *CB-params* to $\mathcal{C}_{II}$.
- $\mathcal{A}_{II}$ randomly chooses an identity id^* and queries CB-RequestPublicKey(id^*) to $\mathcal{C}_{II}$. Let the public key returned by $\mathcal{C}_{II}$ be $CB\text{-}PK^*$.
- $\mathcal{A}_{II}$ randomly chooses an index τ^* of a time period, then queries CB-RequestPublicKey(id^*), CB-RequestCertificate(τ^*, id^*) and CB-ReplacePublicKey(id^*, $CB\text{-}PK^*$) to $\mathcal{C}_I$ respectively.
- $\mathcal{A}_{II}$ randomly chooses two equal length messages M_0, M_1, and submits (τ^*, id^*, M_0, M_1) to $\mathcal{C}_{II}$ as its challenge output. Suppose that the returned challenge ciphertext is C^* which is computed as CB-Encrypt(*params*, τ^*, id^*, $CB\text{-}PK^*$, M_b) for a random bit $b \in \{0, 1\}$.
- $\mathcal{A}_{II}$ queries CB-StrongDecrypt(τ^*, *id*, C^*) to $\mathcal{C}_I$ and is responded with a message M^*.
- If $M^* = M_0$, $\mathcal{A}_{II}$ outputs 0; otherwise, $\mathcal{A}_{II}$ returns 1.

Since $\mathcal{C}_I$ successfully simulates the strong IND-CB-CCA2 Game-I with overwhelming probability, it implies that $\mathcal{C}_I$ will simulate the strong decryption oracle successfully and output the correct message $M^* = M_b$ with overwhelming probability to respond the strong decryption oracle query by $\mathcal{A}_{II}$. Hence, $\mathcal{A}_{II}$ will output the right answer at a non-negligible probability. This proves that the advantage of $\mathcal{A}_{II}$ in the normal and malicious-but-passive IND-CB-CCA2 Game-II is non-negligible. □

The above theorem shows that the sType-I security and the nmType-II security cannot co-exist on any CBE schemes without random oracles in the black-box proof. Since the smType-II security implies the nmType-II security, so the sType-I security and the smType-II security also cannot co-exist on any CBE schemes without random oracles in the black-box proof.

Similarly, a sType-I challenger must be an nType-II attacker in the standard model. That is, the sType-I security and the sType-II security also cannot co-exist on any CBE scheme in the standard model.

Theorem 2. In the standard model, if there exists a black-box proof for the sType-I security of a CBE scheme, then it must not be nType-II secure.

Proof. The proof of this theorem is similar to that of Theorem 1 only with some minor modifications and hence is omitted.

Remark 3. It should be noted that we may prove a CBE scheme to both be sType-I secure and smType-II secure (or sType-II secure) in the random oracle using the black-box security proving technique. For example, the CBE scheme in [28] is proved to be sType-I secure and sType-II secure in the random oracle. This result does not contradict our conclusions above. After all, the game challenger in the random oracle is always assumed to have the full control of some specified random oracles while the one in the standard model has no such power.

Remark 4. The game hopping proof technique [29, 30] may used to prove a CBE scheme to both be sType-I secure and smType-II secure (or sType-II secure) in the standard model. Recently, Dent et al. [31] successfully used this new proof technique to prove their CL-PKE scheme to both be strong Type-I and Type-II secure in the standard model. It makes us believe that the sType-I security and the smType-II (or sType-II) security can co-exist on a CBE scheme without random oracles in a game hopping proof.

4. Generic Construction of CBE from CL-PKE

In this section, we propose a new generic construction of CBE from CL-PKE, and prove the security of the certificate scheme CBE from the construction under different security levels.

4.1. Syntax of CL-PKE

We first briefly review the definition of CL-PKE. In the original work [2], a CL-PKE scheme is defined by seven algorithms (CL-Setup, CL-PartialKeyExtract, CL-SetSecretValue, CL-SetPrivateKey, CL-SetPublicKey, CL-Encrypt, CL-Decrypt) such that:

- CL-Setup: On input a security parameter k, it returns a master key $CL\text{-}msk$ and a list of public system parameters $CL\text{-}params$ that include the descriptions of a finite identity information space $IDSPC^{CL}$, a finite plaintext space $MSPC^{CL}$ and a finite ciphertext space $CSPC^{CL}$. This algorithm is run by the trusted third-party KGC.

- CL-PartialKeyGen: On input *CL-msk*, *CL-params* and an identity *ID* $\in$ $IDSPC^{CL}$ for a user, it returns a partial private key *CL-PPK* for the user with identity *ID*. This algorithm is also run by KGC.
- CL-SetSecretValue: On input *CL-params*, it returns a secret value *CL-SV* for a user.
- CL-SetPrivateKey: On input *CL-params*, *CL-PPK* and *CL-SV*, it returns a full private key *CL-SK* for a user.
- CL-SetPublicKey: On input *CL-params* and *CL-SV*, it returns a public key *CL-PK* for a user.
- CL-Encrypt: On input *CL-params*, *ID*, *CL-PK* and a message *M*, it returns a ciphertext *C*.
- CL-Decrypt: On input *CL-params*, *CL-SK* and a ciphertext *C*, it returns the plaintext *M* or $\perp$ indicating a decryption failure.

In [27, 32], Au et al. also introduced a five-algorithm definition of CL-PKE, which omits the algorithm CL-SetPrivateKey, and replaces the algorithms CL-SetSecretValue and CL-SetPublicKey with a single algorithm CL-UserKeyGen. Concretely, in their definition, a CL-PKE scheme is specified by five algorithms (CL-MasterKeyGen, CL-PartialKeyGen, CL-UserKeyGen, CL-Encrypt, CL-Decrypt) such that:

- CL-MasterKeyGen: On input a security parameter *k*, it returns a master key *CL-msk* and a list of public system parameters *CL-params*.
- CL-PartialKeyGen: On input *CL-msk*, *CL-params* and an identity *ID* $\in$ $IDSPC^{CL}$ for a user, it returns a partial secret key *CL-PSK* for the user with identity *ID*.
- CL-UserKeyGen: On input *CL-params*, it returns a public/secret key pair (*CL-PK*, *CL-SK*) for a user.
- CL-Encrypt: On input *CL-params*, *ID*, *CL-PK* and a message *M*, it returns a ciphertext *C*.
- CL-Decrypt: On input *CL-params*, *CL-SK*, *CL-PSK* and a ciphertext *C*, it returns the plaintext *M* or $\perp$ indicating a decryption failure.

As discussed in [27], this new approach of defining CL-PKE schemes is more versatile than the original seven-algorithm definition in [2], and still maintains the unique feature of CL-PKE schemes. Actually, in [25], Dent also suggested the similar method to construct a CL-PKE scheme by replacing the algorithms CL-SetSecretValue and CL-SetPublicKey with a single algorithm CL-SetUserKey and showed that a CL-PKE scheme presented in the old formulation can also be presented in the new formulation. Our generic construction will adopt the five-algorithm definition of CL-PKE.

4.2. CBE from CL-PKE

Let Π^{CL} = (CL-MasterKeyGen, CL-PartialKeyExtract, CL-UserKeyGen, CL-Encrypt, CL-Decrypt) be a five-algorithm CL-PKE scheme as described

above. Then, a CBE scheme Π^{CB} = (CB-Setup, CB-SetKeyPair, CB-Certify, CB-Encrypt, CB-Decrypt) can be generically constructed from the scheme CL-PKE as follows:

CB-Setup(1^k, *N*):
(*CL-params*, *CL-msk*) ← CL-MasterKeyGen(1^k)
CB-params ← (*CL-params*, *N*)
CB-msk ← *CL-msk*
Return (*CB-params*, *CB-msk*)

CB-SetKeyPair(*CB-params*):
Parse CB-params as (*CL-params*, *N*)
(*CL-PK*, *CL-SK*) ← CL-UserKeyGen(*CL-params*)
(*CB-PK*, *CB-SK*) ← (*CL-PK*, *CL-SK*)
Return (*CB-PK*, *CB-SK*)

CB-Certify(*CB-params*, *CB-msk*, τ, *id*, *CB-PK*):
Parse CB-params as (*CL-params*, *N*)
CL-msk ← *CB-msk*
ID ← *id*||τ||*CB-PK*
CL-PSK ← CL-PartialKeyGen(*CL-params*, *CL-msk*, *ID*)
$CB\text{-}Cert_{\tau}$ ← *CL-PSK*
Return $CB\text{-}Cert_{id,\tau}$

CB-Encrypt(*CB-params*, τ, *id*, *CB-PK*, *M*):
Parse CB-params as (*CL-params*, *N*)
CL-PK ← *CB-PK*
ID ← *id*||τ||*CB-PK*
C ← CL-Encrypt(*CL-params*, *ID*, *CL-PK*, *M*)
Return *C*

CB-Decrypt(*CB-params*, *CB-SK*, $CB\text{-}Cert_{\tau}$, *C*):
Parse CB-params as (*CL-params*, *N*)
CL-SK ← *CB-SK*
CL-PSK ← $CB\text{-}Cert_{\tau}$
M ← CL-Decrypt(*CL-params*, *CL-SK*, *CL-PSK*, *C*)
Return *M*

In the above generic construction, we use the algorithms CL-UserKeyGen and CL-PartialKeyGen to generate the public/private key pair and the certificate in the CBE scheme Π^{CB} respectively. The message and ciphertext spaces of the scheme Π^{CB} are same as those of the scheme Π^{CL}. Furthermore, the identities in the scheme Π^{CL} are of the form *id*||τ||*CB-PK*, that is, the identity information space $IDSPC^{CL}$ in Π^{CL} is equal to $IDSPC^{CB} \times \{0,1\}^{l} \times PKSPC^{CB}$, where *l* is the smallest integer such that $N \leq 2^{l}$ and $PKSPC^{CB}$ is the public key space in Π^{CB}. We should claim that, in the practical conversion, we may use a collision resistant hash function to map $IDSPC^{CB} \times \{0,1\}^{l} \times PKSPC^{CB}$ to a binary string space in which the string has a reasonable length as the identity information space of the CL-PKE scheme to reduce the complexity of the resulting CBE scheme. Here, we put $IDSPC^{CB} \times \{0,1\}^{l} \times PKSPC^{CB}$ as the identity information space of Π^{CL} directly only to simplify the security proof of the resulting CBE scheme Π^{CB}.

Next are our conclusions about the relationships between the resulting CBE scheme Π^{CB} and the underlying CL-PKE scheme Π^{CL}. We refer the readers to [25, 27] for the security definitions of CL-PKE.

Theorem 3. Suppose that the CL-PKE scheme ΠP^{CL} is strong Type-I† secure (resp., weak Type-Ia† secure), then the CBE scheme Π^{CB} from the above generic construction is sType-I secure (resp., nType-I secure).

Proof. Let $\mathcal{A}_I$ be a sType-I adversary against the CBE scheme Π^{CB} with advantage ε. We show how to make use of the adversary $\mathcal{A}_I$ to construct a strong Type-I† adversary $\mathcal{B}_I$ against the CL-PKE scheme Π^{CL} with the same advantage ε. Let $\mathcal{C}$ be the challenger against $\mathcal{B}_I$ in the strong IND-CL-CCA2 Game-I, who provides $\mathcal{B}_I$ with following oracles:

- CL-RequestPublicKey(*ID*): return the public key for the identity *ID*.
- CL-ReplacePublicKey(*ID*, *CL-PK*$'$): replace the current public key of the identity *ID* with the value *CL-PK*$'$.
- CL-ExtractSecretKey(*ID*): return the secret key (value) for the identity *ID*.
- CL-ExtractPartialKey(*ID*): return the partial private key for the identity *ID*.
- CL-StrongDecrypt(*ID*, *C*): return the correct decryption of *C*.

After given the public system parameters *CL-params* by $\mathcal{C}$, $\mathcal{B}_I$ simulates the challenger in the strong IND-CB-CCA2 Game-I and interacts with $\mathcal{A}_I$ as follows:

- Setup: $\mathcal{B}_I$ forwards *CL-params* as *CB-params* to $\mathcal{A}_I$.
- Phase 1: Upon receiving *CB-params*, $\mathcal{A}_I$ quires onto the oracles CB-RequestPublicKey, CB-RequestCertificate, CB-ExtractPrivateKey, CB-ReplacePublicKey and CB-StrongDecrypt in an adaptive manner. $\mathcal{B}_I$ responds as follows:
  - CB-RequestPublicKey(*id*): On receiving such a query, $\mathcal{B}_I$ first extends the identity *id* to a valid identity *ID* = *id*0^m in CL-PKE by inserting a suffix consisting of *m* zeros to the identity *id*, where $m = l + |PKSPC^{CB}|$. We assume that $\mathcal{B}_I$ always uses the same method to extend an identity in Π^{CB} to a valid identity in Π^{CL} in the following simulation. Then $\mathcal{B}_I$ queries CL-RequestPublicKey(*ID*) to $\mathcal{C}$ and returns $\mathcal{C}$'s respond to $\mathcal{A}_I$.
  - CB-ReplacePublicKey(*id*, *CB-PK*$'$): On receiving such a query, $\mathcal{B}_I$ makes a public key replace query CL-ReplacePublicKey(*ID*, *CB-PK*$'$) to $\mathcal{C}$ to replace the public key of the identity *ID* with the value *CB-PK*$'$.
  - CB-RequestCertificate(*id*, τ): On receiving such a query, $\mathcal{B}_I$ first queries

CL-RequestPublicKey(*ID*) to obtain a public key *CL-PK* for the identity *ID*, sets $ID' = id||\tau||CB\text{-}PK$ and queries CL-ReplacePublicKey(ID', *CB-PK*) to replace the public key of the identity *ID* with *CB-PK*. Then, it queries CL-ExtractPartialKey(ID') to $\mathcal{C}$ and returns $\mathcal{C}$'s response to $\mathcal{A}_I$.

- CB-ExtractPrivateKey(*id*): On receiving such a query, $\mathcal{B}_I$ queries CL-ExtractSecretKey(*ID*) to $\mathcal{C}$. If $\mathcal{C}$ responds with a secret key, then $\mathcal{B}_I$ returns $\mathcal{C}$'s response to $\mathcal{A}_I$. Otherwise, if $\mathcal{C}$ rejects its query, namely that the public key for *ID* has been replaced, then $\mathcal{B}_I$ rejects $\mathcal{A}_I$'s query too.
- CB-StrongDecrypt(τ, *id*, *C*): On receiving such a query, $\mathcal{B}_I$ first queries CL-RequestPublicKey(*ID*) to obtain a public key *CL-PK* for the identity *ID*, sets $ID' = id||\tau||CB\text{-}PK$ and queries CL-ReplacePublicKey(ID', *CB-PK*) to replace the public key of the identity ID' with *CL-PK*. Then, it queries CL-StrongDecrypt(ID', *C*) to $\mathcal{C}$ and returns $\mathcal{C}$'s response to $\mathcal{A}_I$.

– Challenge: Once $\mathcal{A}_I$ decides that Phase 1 is over, it outputs an index τ^* of a time period, an identity id^* and two equal length messages M_0, M_1, on which it wants to be challenged. $\mathcal{B}_I$ first queries CL-RequestPublicKey(ID^*) to obtain a public key $CL\text{-}PK^*$ for the identity ID^*, then queries CL-ReplacePublicKey($id^*||\tau^*||CL\text{-}PK^*$, $CL\text{-}PK^*$) to replace the public key of the identity $id^*||\tau^*||CL\text{-}PK^*$ with $CL\text{-}PK^*$. After that, it terminates Phase 1 of the strong IND-CL-CCA2 Game-I and submits ($id^*||\tau^*||CL\text{-}PK^*$, M_0, M_1) to $\mathcal{C}$ to enter its challenge phase. The latter responds with a challenge ciphertext C^* = CL-Encrypt(*CL-params*, $id||\tau||CL\text{-}PK^*$, $CL\text{-}PK^*$, M_b) for a random bit $b \in \{0,1\}$. $\mathcal{B}_I$ forwards C^* to $\mathcal{A}_I$ as the challenge ciphertext in the strong IND-CB-CCA2 Game-I.
– Phase 2: $\mathcal{A}_I$ issues a second sequence of queries as in Phase 1, with the restrictions specified in Definition 4.
– Guess: Finally, $\mathcal{A}_I$ outputs a guess $b' \in \{0,1\}$ for *b*, and $\mathcal{B}_I$ outputs the same bit to $\mathcal{C}$.

Now, we calculated $\mathcal{B}_I$'s advantage of outputting the right bit in the above game. Firstly, it is obvious that if $\mathcal{B}_I$ does not abort during the simulation, then $\mathcal{A}_I$'s view is adventitial to its view in the real attack. So, if $\mathcal{B}_I$ does not abort, we have that $|\Pr[b = b'] - \frac{1}{2}| = \varepsilon$. Next, we analyze the probability that $\mathcal{B}_I$ does not abort during the simulation. According to the definition of the strong Type-I† security of CL-PKE [25], $\mathcal{B}_I$ may abort when one of the following four events happens:

- Event 1: $\mathcal{B}_I$ is forced to query both the oracles CL-ExtractSecretKey and CL-ExtractPartialKey on the challenge identity $id^*||\tau^*||CL\text{-}PK^*$.
- Event 2: $\mathcal{B}_I$ is forced to query CL-ExtractSecretKey on any identity if the corresponding public key has been replaced.
- Event 3: $\mathcal{B}_I$ is forced to both query CL-ReplacePublicKey on the challenge identity $id^*||\tau^*||CL\text{-}PK^*$ before the challenge phase and CL-ExtractPartialKey on the challenge identity $id^*||\tau^*||CL\text{-}PK^*$.
- Event 4: $\mathcal{B}_I$ is forced to query CL-StrongDecrypt on the challenge ciphertext C^* for the challenge identity $id^*||\tau^*||CL\text{-}PK^*$ in Phase 2.

We show that all of the above events never occur in $\mathcal{B}_I$'s simulation.

- Firstly, Event 1 can happen when $\mathcal{A}_I$ query both CB-ExtractPrivateKey($id^*||\tau^*||CL\text{-}PK^*$) and CB-RequestCertificate(id^*, τ^*). This event never occurs in $\mathcal{B}_I$'s simulation since $\mathcal{A}_I$ is forbidden from querying CB-RequestCertificate(id^*, τ^*) and never queries CB-ExtractPrivateKey($id^*||\tau^*||CL\text{-}PK^*$). Note that $id^*||\tau^*||CL\text{-}PK^*$ is an identity which never appears in the identity information space of the scheme Π^{CB}.
- Secondly, Event 2 can happen only if $\mathcal{A}_I$ query CB-ExtractPrivateKey on an identity which has been replaced the public key. However, $\mathcal{A}_I$ is forbidden from querying such query in the strong IND-CB-CCA2 Game-I. So this event never occurs in $\mathcal{B}_I$'s simulation.
- Thirdly, Event 3 can happen only if $\mathcal{A}_I$ query CB-RequestCertificate(id^*, τ^*). But this is exactly the certificate query which $\mathcal{A}_I$ is forbidden from making in the strong IND-CB-CCA2 Game-I. So this event never occurs in $\mathcal{B}_I$'s simulation.
- Finally, Event 4 can happen only if $\mathcal{A}_I$ query CB-StrongDecrypt(τ^*, id^*, C^*). However, $\mathcal{A}_I$ is forbidden from making such decryption query in the strong IND-CB-CCA2 Game-I. So this event never occurs in $\mathcal{B}_I$'s simulation.

To summarize, $\mathcal{B}_I$ never aborts during the simulation and provides a perfect simulation of challenger against $\mathcal{A}_I$ in the strong IND-CB-CCA2 Game-I. Thus, it has an advantage ε in guessing b. Since Π^{CL} is a strong Type-I† secure CL-PKE scheme, then Π^{CB} is a sType-I secure CBE scheme. Similarly, we can prove that an nType-I adversary against Π^{CB} can be used to construct a weak Type-Ia† adversary against the scheme Π^{CL}. This completes the proof of this theorem. □

Theorem 4. Suppose that the CL-PKE scheme Π^{CL} is strong and malicious-but-passive Type-II† secure (resp., weak and malicious-but-passive Type-II† secure), then the CBE scheme Π^{CB} from the above generic construction is smType-II secure (resp., nmType-II secure).

Proof. Let $\mathcal{A}_{II}$ be a smType-II adversary against the scheme Π^{CB} with advantage ε. We show how to make use of the adversary $\mathcal{A}_{II}$ to construct a strong and malicious-but-passive Type-II† adversary $\mathcal{B}_{II}$ against the scheme Π^{CL} with the same advantage ε. Let $\mathcal{C}$ be the challenger against $\mathcal{B}_{II}$ in the strong and malicious-but-passive IND-CL-CCA2 Game-II. $\mathcal{C}$ invokes $\mathcal{B}_{II}$ on input 1^k to begin the strong and malicious-but-passive IND-CL-CCA2 Game-II. $\mathcal{B}_{II}$ simulates the challenger in the strong IND-CB-CCA2 Game-II and interacts with $\mathcal{A}_{II}$ as follows:

- Setup: $\mathcal{B}_{II}$ invokes $\mathcal{A}_{II}$ on input (1^k, *N)* and obtains a list of public parameters *CB-params*. $\mathcal{B}_{II}$ forwards *CB-params* as *CL-params* to $\mathcal{C}$. Note that $\mathcal{C}$ provides $\mathcal{B}_{II}$ with oracles CL-RequestPublicKey, CL-ReplacePublicKey, CL-ExtractSecretKey, CL-StrongDecrypt, which are defined as same as in the proof of Theorem 3.
- Phase 1: In this phase, $\mathcal{A}_{II}$ quires onto the oracles CB-RequestPublicKey, CB-ExtractPrivateKey, CB-ReplacePublicKey and CB-StrongDecrypt in an adaptive manner. $\mathcal{B}_{II}$ responds as in the proof of Theorem 3.
- Challenge: Once $\mathcal{A}_{II}$ decides that Phase 1 is over, it outputs an index τ^* of a time period, an identity id^* and two equal length messages M_0, M_1, on which it wants to be challenged. $\mathcal{B}_{II}$ first queries CL-RequestPublicKey(ID^*) to obtain a public key $CL\text{-}PK^*$ for the identity ID^*, then queries CL-ReplacePublicKey($id^*||\tau^*||CL\text{-}PK^*$, $CL\text{-}PK^*$) to replace the public key of the identity $id^*||\tau^*||CL\text{-}PK^*$ with $CL\text{-}PK^*$. After that, it terminates Phase 1 of the strong and malicious-but-passive IND-CL-CCA2 Game-II and submits ($id^*||\tau^*||CL\text{-}PK^*$, M_0, M_1) to $\mathcal{C}$ to enter its challenge phase. The latter responds with a challenge ciphertext C^* = CL-Encrypt(*CL-params*, $id^*||\tau^*||CL\text{-}PK^*$, $CL\text{-}PK^*$, M_b) for a random bit $b \in \{0,1\}$. $\mathcal{B}_{II}$ forwards C^* to $\mathcal{A}_{II}$ as the challenge ciphertext in the strong and malicious-but-passive IND-CB-CCA2 Game-II.
- Phase 2: $\mathcal{A}_{II}$ issues a second sequence of queries as in Phase 1.
- Guess: Finally, $\mathcal{A}_{II}$ outputs a guess $b' \in \{0,1\}$ for b, and $\mathcal{B}_{II}$ outputs the same bit to $\mathcal{C}$.

Now, we calculated $\mathcal{B}_{II}$'s advantage of outputting the right bit in the above game. Firstly, it is obvious that if $\mathcal{B}_{II}$ does not abort during the simulation then $\mathcal{A}_{II}$'s view is identical to its view in the real attack. So, if $\mathcal{B}_{II}$ does not abort, we have that $|\Pr[b = b'] - \frac{1}{2}| = \varepsilon$. Next, we analyze the probability that $\mathcal{B}_{II}$ does not abort during the simulation. According to the definition of the strong and malicious-but-passive Type-II† security of CL-PKE [27], $\mathcal{B}_{II}$ can abort when one of the following four events happens:

- Event 1: $\mathcal{B}_I$ is forced to query CL-ReplacePublicKey on the challenge identity $id^*||\tau^*||CL\text{-}PK^*$ before the challenge phase.
- Event 2: $\mathcal{B}_I$ is forced to query CL-ExtractSecretKey on the challenge identity $id^*||\tau^*||CL\text{-}PK^*$.
- Event 3: $\mathcal{B}_I$ is forced to query CL-ExtractSecretKey on any identity if the corresponding public key has been replaced.
- Event 4: $\mathcal{B}_I$ is forced to query CL-StrongDecrypt on the challenge ciphertext C^* for the challenge identity $id^*||\tau^*||CL\text{-}PK^*$ in Phase 2.

We show that all of the above events never occur in $\mathcal{B}_{II}$'s simulation.

- Firstly, it is clear that Event 1 and Event 2 never occur in $\mathcal{B}_{II}$'s simulation since the identity $id^*||\tau^*||CL\text{-}PK^*$ is never queried upon by $\mathcal{A}_{II}$.
- Secondly, Event 3 can happen only if $\mathcal{A}_{II}$ query CB-ExtractPrivateKey on an identity which has been replaced the public key. However, $\mathcal{A}_{II}$ is forbidden from making such queries in the strong and malicious-but-passive IND-CB-CCA2 Game-II. So this event never occurs in $\mathcal{B}_{II}$'s simulation.
- Finally, Event 4 can happen only if $\mathcal{A}_{II}$ query CB-StrongDecrypt(τ^*, id^*, C^*). However, $\mathcal{A}_I$ is forbidden from making such decryption query in the strong and malicious-but-passive IND-CB-CCA2 Game-II. So this event never occurs in $\mathcal{B}_{II}$'s simulation.

To summarize, $\mathcal{B}_{II}$ never aborts during the simulation and provides a perfect simulation of challenger against $\mathcal{A}_{II}$. Thus, it has an advantage ε in guessing b. Since Π^{CL} is a strong and malicious-but-passive Type-II† secure CL-PKE scheme, then Π^{CB} is a smType-II secure CBE scheme. Similarly, we can prove that an nmType-II adversary against the scheme Π^{CB} can be used to construct a weak and malicious-but-passive Type-II† adversary against the scheme Π^{CL}. This completes the proof of this theorem. □

Similar to Theorem 4, it is not difficult to deduce that:

Theorem 5. Suppose that the scheme the CL-PKE scheme Π^{CL} is strong Type-II† secure (resp., weak Type-II† secure), then the CBE scheme Π^{CB} from the above generic construction is sType-II secure (resp., nType-II secure).

5. A Concrete Conversion

In this section, we show a concrete conversion from the CL-PKE scheme in [33] to a CBE scheme using our generic construction. We first briefly review the concept of bilinear map and the related complexity assumption, and then describe the concrete CBE scheme.

5.1. Bilinear Map and Complexity Assumption

Let p be a large prime number, G and G_T denote two multiplicative cyclic groups of the same order p. A mapping e: $G \times G \rightarrow G_T$ is called a bilinear map if it satisfies the following properties:

- Bilinearity: $e(u^a, v^b) = e(u, v)^{ab}$ for all $u, v \in G$ and $a, b \in Z_p^*$.
- Non-degeneracy: $e(g, g) \neq 1$ for a random generator $g \in G$.
- Computablity: $e(u, v)$ can be efficiently computed for all $u, v \in G$.

Definition 8. The Decisional Bilinear Diffie-Hellman (DBDH) problem in (G, G_T) is defined as follows: Given a tuple $(g, g^a, g^b, g^c) \in G^4$ and an element $T \in G_T$ where $a, b, c \in Z_p^*$, decide whether $T = e(g, g)^{abc}$ or T is a random element of G_T. Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) algorithm that takes as input a random instance of the DBDH problem and outputs a bit $b \in \{0,1\}$. We say that the DBDH assumption holds in (G, G_T) if no PPT algorithm has non-negligible advantage in solving the DBDH problem in (G, G_T). The advantage of $\mathcal{A}$ is defined as

$$Adv_{\mathcal{A}}^{DBDH}(k) = \left|\Pr[1 \leftarrow \mathcal{A}(g, g^a, g^b, g^c, e(g,g)^{abc})] - \Pr[1 \leftarrow \mathcal{A}(g, g^a, g^b, g^c, T)]\right|$$

where the probability is over the randomly chosen $a, b, c \in Z_p^*$ and the random bits consumed by $\mathcal{A}$.

Definition 9. A hash function $H \leftarrow \mathcal{H}(k)$ is collision resistant if for all PPT algorithms $\mathcal{A}$ the advantage

$$Adv_{\mathcal{A}}^{CR}(k) = \Pr[H(x) = H(y) \wedge x \neq y \mid (x, y) \leftarrow \mathcal{A}(1^k, H) \wedge H \leftarrow \mathcal{H}(k)]$$

is negligible.

5.2. A New CBE Scheme

Now, we describe the CBE scheme converted from the CL-PKE scheme in [33], which is constructed based on the Water's IBE scheme [17] and is proved to be secure against malicious-but-passive KGC (Key Generation Center) attack in the standard model. We note that this CL-PKE scheme is the first one in the literature to achieve the strongest Type-II security without random oracles. So the resulting CBE scheme in this subsection is also the first one to achieve the malicious-but-passive Type-II security without random oracles in the case of CBE. The scheme is described as follows:

- CB-Setup(1^k): Given a security parameter $k \in Z^+$ and a total number of time periods $N \in Z^+$, this algorithm generates the public parameters and the master key as follows: Generate two multiplicative cyclic groups G and G_T of big prime order p, and a bilinear pairing map e: $G \times G \rightarrow G_T$. Choose a random generator $g \in G$. Randomly choose $\alpha,\beta,\mu,\upsilon \in Z_p^*$ and $\mu_i,\upsilon_i \in Z_p^*$ for $i = 1,2,\ldots,n$, and compute $g_1 = g^\alpha$, $h = e(g^\alpha, g^\beta)$, $u = g^\mu$, $v = g^\upsilon$, $u_1 = g^{\mu_1},\ldots,u_n = g^{\mu_n}$, $v_1 = g^{\upsilon_1},\ldots,v_n = g^{\upsilon_n}$, where n is the bit length of the identity information of the underlying CL-PKE scheme. Let H: $\{0,1\}^* \rightarrow \{0,1\}^n$ be a collision resistant hash function. The public parameters are *CB-params* = {p, e, G_1, G_T, g, g_1, h, u, u_1,…, u_n, v, v_1,…, v_n, H} and master key is *CB-msk* = β.
- CB-SetKeyPair(*CB-params*): This algorithm chooses a random element $x \in Z_q^*$ as the private key *CB-SK* for a user and generates the corresponding public key as *CB-PK* = (X, σ) where $X = h^x$ and σ is the Schnorr one-time signature of the message $id||X$ using x as the signing key and (h, X) as the verification key. We refer the readers to [33] for the details about the generation of the public key.
- CB-Certify**(***CB-params, CB-msk*, τ, *id*, *CB-PK***):** This algorithm first sets *ID* = $id||\tau||X||\sigma$. Let ID_i be the i-th bit of *ID*. Then, it randomly selects $s \in Z_p^*$ and computes

$$CB\text{-}Cert_\tau = (Cert_\tau^1, Cert_\tau^2) = (g_1^\beta \cdot F_u(ID)^s, g^s) \text{ where } F_u(ID) = u\prod_{i=1}^n u_i^{ID_i}.$$

- CB-Enc(*CB-params,* τ, *id*, *CB-PK*, *M*)**:** This algorithm first checks whether the public key *CB-PK* = (X, σ) is correctly formed by verifying whether σ is a valid signature of the message $id||X$, using (h, X) as the verification key. If not, it outputs $\perp$ and aborts the algorithm. Otherwise, it randomly chooses $r \in Z_p^*$ and computes the ciphertext as

$$C = (C_0, C_1, C_2, C_3) = (M \cdot X^r, g^r, F_u(ID)^r, F_v(w)^r)$$

where $ID = id||\tau||X||\sigma$, $w = H(C_0, C_1, C_2, ID) \in \{0,1\}^n$ and $F_v(w) = u\prod_{i=1}^n v_i^{w_i}$.

- CB-Dec(*CB-params, CB-SK, CB-Cert*$_{id,\tau}$, *C*): This algorithm first parses C into (C_0, C_1, C_2, C_3) and checks that $e(C_1, F_u(ID) \cdot F_v(w)) = e(g, C_2 \cdot C_3)$. If not, it

outputs $\perp$ and aborts the algorithm. Otherwise, it randomly chooses $t \in Z_p^*$ and computes

$$(d_1, d_2) = ((Cert_\tau^1)^{CB\text{-}SK} \cdot F(ID)_u^t, (Cert_\tau^2)^{CB\text{-}SK} \cdot g^t) = (g_1^{\beta x} \cdot F(ID)_u^{sx+t}, g^{sx+t}).$$

It then computes the plaintext as

$$M = C_0 \cdot e(C_2, d_2)/e(C_1, d_1).$$

Next is our conclusion about the security of the above CBE scheme.

Theorem 6. The above CBE scheme is nType-I and nmType-II secure if the DBDH assumption holds in (G, G_T) and the hash function H is collision resistant.

Proof. The correctness of this theorem can be proved by combining Theorem 3, Theorem 4 in this paper, and Theorem 5, Theorem 6 in [32].

6. Conclusion

In this paper, we made further observations on CBE and its generic construction from CL-PKE. We first analyzed the existing security models of CBE and gave new definitions of the security models for CBE under different security levels according to the attacking power of the adversaries against CBE. Our definitions are more reasonable and elaborated compared with other existing ones. We then proposed a generic construction of CBE from CL-PKE which is secure in the standard model if the underlying CL-PKE scheme satisfies certain security. Finally, we gave a concrete conversion from an existing CL-PKE scheme to a CBE scheme using our generic construction.

References

1. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In Proceedings of CRYPTO 1984, Lecture Notes in Computer Science, Vol. 196. Springer-Verlag, Berlin Heidelberg, 47–53. (1984)
2. Al-Riyami, S. S., Paterson, K. G.: Certificateless Public Key Cryptography. In Proceedings of ASIACRYPT 2003, Lecture Notes in Computer Science, Vol. 2894. Springer-Verlag, Berlin Heidelbergk, 452–473. (2003)
3. Gentry, C.: Certificate-Bbased Encryption and the Certificate Revocation Problem. In Proceedings of EUROCRYPT 2003, Lecture Notes in Computer Science, Vol. 2656. Springer-Verlag, Berlin Heidelberg, 272–293. (2003)
4. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of ACM CCS 1993. ACM Press, 62–73. (1993)
5. Boneh, D., Franklin, M.: Identity Based Encryption from the Weil Pairing. In Proceedings of CRYPTO 2001, Lecture Notes in Computer Science, Vol. 2139. Springer-Verlag, Berlin Heidelberg, 213–229. (2001)

6. Yum, D. H., Lee, P. J.: Identity-Based Cryptography in Public Key Management. In Proceedings of EuroPKI 2004, Lecture Notes in Computer Science, Vol. 3093. Springer-Verlag, Berlin Heidelberg, 71–84. (2004)
7. Galindo, D., Morillo, P., Ràfols, C.: Breaking Yum and Lee Generic Constructions of Certificateless and Certificate-Based Encryption Schemes. In Proceedings of EuroPKI 2006, Lecture Notes in Computer Science, Vol. 4043. Springer-Verlag, Berlin Heidelberg New York, 81–91. (2006)
8. Lu, Y., Li, J. G., Xiao, J. M.: Generic Construction of Certificate-Based Encryption. In Proceedings of the 9th International Conference for Young Computer Scientists, IEEE CS, 1518–1594. (2008)
9. Fujisaki, E., Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost. In Proceedings of PKC 1999, Lecture Notes in Computer Science, Vol. 1560. Springer-Verlag, Berlin Heidelberg New York, 53–68. (1999)
10. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In Proceedings of CRYPTO 1999, Lecture Notes in Computer Science, Vol. 1666. Springer-Verlag, Berlin Heidelberg, 537–554. (1999)
11. Lu, Y., Li, J. G.: Generic Construction of Certificate-Based Encryption in the Standard Model. In Proceedings of the 2nd International Symposium on Electronic Commerce and Security, IEEE CS, 25–29. (2009)
12. Al-Riyami, S. S., Paterson, K. G.: CBE from CL-PKE: A Generic Construction and Efficient Schemes. In Proceedings of PKC 2005, Lecture Notes in Computer Science, Vol. 3386. Springer-Verlag, Berlin Heidelberg, 398–415. (2005)
13. Kang, B. G., Park, J. H.: Is It Possible to Have CBE from CL-PKE? Cryptology ePrint Archive, Report 2005/431 (2005). [Online]. Available: http://eprint.iacr.org/2005/431.pdf (current December 2010)
14. Yum, D. H., Lee, P. J.: Separable Implicit Certificate Revocation. In Proceedings of ICISC 2004, Lecture Notes in Computer Science, Vol. 3506. Springer-Verlag, Berlin Heidelberg New York, 121–136. (2005)
15. Park, J. H., Lee, D. H.: On the Security of Status Certificate-Based Encryption Scheme. IEICE Trans. Fundamentals, Vol. E90-A, No.1, 303–304. (2007)
16. Morillo, P., Ràfols, C.: Certificate-Based Encryption without Random Oracles. Cryptology ePrint Archive, Report 2006/12 (2006). [Online]. Available: http://eprint.iacr.org/2006/12.pdf (current December 2010)
17. Waters, B.: Efficient Identity-Based Encryption without Random Oracles. In Proceedings of EUROCRYPT'05, Lecture Notes in Computer Science, Vol. 3494. Springer-Verlag, Berlin Heidelberg New York, 114–127. (2005)
18. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity Based Encryption without Random Oracles. In Proceedings of EUROCRYPT 2004, Lecture Notes in Computer Science, Vol. 3027. Springer-Verlag, Berlin Heidelberg New York, 223–238. (2004)
19. Galindo, D., Morillo, P., Ràfols, C.: Improved Certificate-Based Encryption in the Standard Model. Journal of Systems and Software, Vol. 81, No. 7, 1218–1226. (2008)
20. Liu, J. K., Zhou, J., 2008. Efficient Certificate-Based Encryption in the Standard Model. In Proceedings of SCN 2008, Lecture Notes in Computer Science, Vol. 5229. Springer-Verlag, Berlin Heidelberg New York, 144–155. (2008)
21. Gentry, C.: Practical Identity-Based Encryption without Random Oracles. In Proceedings of EUROCRYPT 2006, Lecture Notes in Computer Science, Vol. 4004. Springer-Verlag, Berlin Heidelberg New York, 445–464. (2006)
22. Lu, Y., Li, J. G., Xiao, J. M.: Constructing Efficient Certificate-Based Encryption with Paring. Journal of Computers, Vol. 4, No. 1, 19–26. (2009)

23. Sakai, R., Kasahara, M.: ID Based Cryptosystems with Pairing on Elliptic Curve. Cryptology ePrint Archive, Report 2003/054 (2003). [Online]. Available: http://eprint.iacr.org/2003/054.pdf (current December 2010)
24. Chen, L. Q., Cheng, Z. H.: Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. In Proceedings of Cryptography and Coding 2005, Lecture Notes in Computer Science, Vol. 3706. Springer-Verlag, Berlin Heidelberg, 442–459. (2005)
25. Dent, A. W.: A Survey of Certificateless Encryption Schemes and Security Models. International Journal of Information Security, Vol. 7, No. 5, 349–377. (2008)
26. Wu, W., Mu, Y., Susilo, W., Huang, X. Y.: Certificate-Based Signatures Revisited. Journal of Universal Computer Science, Vol. 15, No. 8, 1659–1684. (2009)
27. Au, M. H., Chen, J., Liu, J. K., Mu, Y., Wong, D., Yang, G.: Malicious KGC Attacks in Certificateless Cryptography. In Proceedings of ASIACCS 2007. ACM, 302–311. (2007)
28. Lu, Y., Li, J. G.: Strongly Secure Certificate-Based Encryption Scheme with Low Communication Bandwidth. Cryptology ePrint Archive, Report 2010/553 (2010). [Online]. Available: http://eprint.iacr.org/2010/553.pdf (current December 2010)
29. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Proceedings of Eurocrypt 2006, Lecture Notes in Computer Science, Vol. 4004. Springer-Verlag, Berlin Heidelberg, 409–426. (2006)
30. Shoup, V.: Sequences of Games: A Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive, Report 2004/332 (2004). [Online]. Available: http://eprint.iacr.org/2004/332.pdf (current December 2010)
31. Dent, A. W., Libert, B., Paterson, K. G.: Certificateless Encryption Schemes Strongly Secure in the Standard Model. In Proceedings of PKC 2008, Lecture Notes in Computer Science, Vol. 4943. Springer-Verlag, Berlin Heidelberg, 344–359. (2008)
32. 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 ASIACCS 2007, ACM, 273-283. (2007)
33. Hwang, Y. H., Liu, J. K., Chow, S. S. M.: Certificateless Public Key Encryption Secure against Malicious KGC Attacks in the Standard Model. Journal of Universal Computer Science, Vol. 14, No. 3, 463–480. (2008)