Breaking Yum and Lee Generic Constructions of Certificate-less and Certificate-based Encryption Schemes

David Galindo[1], Paz Morillo[2] and Carla Ràfols[2]

[1] Institute for Computing and Information Sciences, Radboud University Nijmegen,
P.O.Box 9010, 6500 GL, Nijmegen, The Netherlands d.galindo@cs.ru.nl

[2] Universitat Politècnica de Catalunya, C/Jordi Girona, 1-3 08034 Barcelona
{paz,crafols}@ma4.upc.edu

Abstract. Identity-based public key cryptography is aimed at simplifying the management of certificates in traditional public key infrastructures by means of using the identity of a user as its public key. The user must identify itself to a trusted authority in order to obtain the secret key corresponding to its identity. The main drawback of this special form of public key cryptography is that it is key escrowed. Certificate-based and certificate-less cryptography have been recently proposed as intermediate paradigms between traditional and identity-based cryptography, seeking to simplify the management of certificates while avoiding the key escrow property of identity-based cryptography. In this work we cryptanalyse the certificate-based and certificate-less encryption schemes presented by Yum and Lee at EuroPKI 2004 and ICCSA 2004 conferences.

Keywords: public-key infrastructure, identity-based encryption, certificate-based and certificate-less encryption, cryptanalysis.

1 Introduction

In traditional public key cryptography (PKC) the authenticity of the public keys must be certified by a trusted third party, which is called Certification Authority (CA). The infrastructure required to support traditional PKC is the main difficulty in its deployment. Many of the problems of any public key infrastructure arise from the management of certificates, which includes storage, revocation and distribution.

In 1984, Shamir proposed the concept of identity-based PKC, which sought to reduce the requirements on the public key infrastructure by using a well-known aspect of the client's identity as its public key. With this approach, certification becomes implicit. For instance, in the case of identity-based encryption (IBE), the sender of a message does not need to check whether the receiver is certified or not. Instead, prior to decryption, the receiver must identify himself to a trusted authority who is in possession of a master key. If the identification is successful, the authority sends the user his private key. The first practical provably secure

IBE scheme was proposed by Boneh and Franklin in 2001, using bilinear maps on elliptic curves and it was proven secure in the random oracle model [7]. The main drawback of IBE is that it is inherently key escrowed, which limits the applicability of IBE.

Motivated by the above problem, the concept of certificate-based PKC was introduced by Gentry in [11]. In this model, certificates are needed to generate the user's secret key, so certification becomes implicit. In addition there is no key escrow, since the user's secret key is generated by joining both the certificate and a private information only known to the user. In a certificate-based encryption (CBE) scheme, senders are not required to obtain fresh information of receivers' certificate status; the receiver will be able to decrypt only if its public key is certified.

Independently from the previous work, the concept of certificate-less PKC was introduced by Al Riyami and Paterson in [1]. In contrast to traditional public key cryptographic systems, CL-PKC does not require the use of certificates to guarantee the authenticity of public keys. It does rely on the use of a trusted authority who is in possession of a master key. On the other hand, CL-PKC does not suffer from key escrow, since the authority does not have access to the user's private key. Several cryptographic primitives for certificate-less PKC were proposed in [1], including a certificate-less public key encryption (CL-PKE) scheme.

In contrast to IBE, the confidentiality of CBE and CL-PKE schemes must be protected against dishonest users as well as against the trusted authorities. Security notions taking into account these new scenarios were proposed in the seminal works [11, 1].

Thus, certificate-less PKC and certificate-based PKC can be conceptually seen as intermediates between traditional PKC and identity-based PKC. This idea motivated the work by Yum and Lee [15, 16], in which they tried to show a formal equivalence among IBE, CBE and CL-PKE. In particular, their intention was to show that IBE implies both CBE and CL-PKE by giving a generic construction from IBE to those primitives. To do so, they defined a weaker security model for CL-PKE than the original model introduced in [1]. Their generic constructions have been cited as sound constructions in the works [2, 3, 9, 12, 13][3].

Our contribution. In this paper we show that a dishonest authority can break the security of the three generic constructions of CBE and CL-PKE schemes given in [15, 16]. These constructions are inherently flawed due to a naive use of double encryption as highlighted in [10]. We stress that our attacks are within the restricted security model proposed by Yum and Lee, that is, *our results contradict* three of their theorems.

Related work. In a recent work [13], Libert and Quisquater [13] show that the transformation from IBE to CL-PKE in [15] due to Yum and Lee is insecure in the full original security model [1]. Their attack does not in work in the restricted

[3] In the work [13] only the transformations in [16] are regarded as valid constructions in the restricted security model.

security model of [15], so the attack in [13] *does not contradict* Yum and Lee claim.

A generic construction from IBE to CBE was outlined by Dodis and Katz in [10]. They study the security of *multiple encryption*, i.e. the encryption of data using multiple, independent encryption schemes. They provide a generic construction of multiple encryption for public key encryption schemes and suggest how to use their ideas to obtain CBE secure constructions. In [2] a transformation from CL-PKE to CBE was proposed, but the security proof was only given for one of the two attacks that a CBE scheme has to withstand. Recent work [12] pointed out the impossibility of using the same techniques to prove security against the other type of attacks, calling into question the meaningfulness of that transformation.

Regarding CL-PKE, the generic constructions from IBE to CL-PKE we are aware of are to be found in [5, 13]. The drawback of these constructions is that they use the random oracle model heuristic, and therefore it does not actually guarantees soundness of the security reductions in the standard complexity model [8]. In [13] it is also pointed out that the generic construction IBE-to-CBE suggested in [10] also applies to the CL-PKE case, as long as the restricted security model of Yum and Lee is considered.

Therefore, designing a generic transformation from IBE to CL-PKE without random oracles in the full security model proposed in [1] remains an open problem to the best of our knowledge.

2 Definitions for identity-based encryption

We begin by fixing some notation. If A is a non-empty set, then $x \leftarrow A$ denotes that x has been uniformly chosen in A. If A is a finite set, then $|A|$ denotes its cardinality.

An identity-based encryption scheme is specified by four probabilistic polynomial time (PPT) algorithms (see for instance [6]):

- $\mathsf{ID.Gen}$ takes a security parameter k and returns the system parameters $\texttt{ID.pms}$ and master-key $\texttt{ID.msk}$. The system parameters include the description of sets $\mathcal{M}, \mathcal{C}$, which denote the set of messages and ciphertexts respectively. $\texttt{ID.pms}$ is publicly available, while $\texttt{ID.msk}$ is kept secret by the trusted authority.
- $\mathsf{ID.Ext}$ takes as inputs $\texttt{ID.pms}, \texttt{ID.msk}$ and an arbitrary string $ID \in \{0,1\}^*$ and returns a private key d_{ID} to the user with identity ID. This must be done over a secure channel, since d_{ID} enables to decrypt ciphertexts under the identity ID.
- $\mathsf{ID.Enc}$ takes as inputs $\texttt{ID.pms}$, $ID \in \{0,1\}^*$ and $M \in \mathcal{M}$. It returns a ciphertext $C \in \mathcal{C}$.
- $\mathsf{ID.Dec}$ takes as inputs $\texttt{ID.pms}$, $C \in \mathcal{C}$ and a private key d_{ID}, and it returns $M \in \mathcal{M}$ or rejects.

Chosen ciphertext security. An IBE scheme is said to have indistinguishability against an adaptive chosen ciphertext attack ($\mathsf{IND\text{-}ID\text{-}CCA}$) if any PPT algorithm $\mathcal{A}$ has a negligible advantage in the following game:

Setup The challenger takes a security parameter k and runs the $\mathsf{ID.Gen}$ algorithm. It gives $\mathtt{ID.pms}$ to the adversary. It keeps $\mathtt{ID.msk}$ to itself.

Phase 1 The adversary issues queries of the form

- Extraction query $\langle ID_i \rangle$. The challenger runs algorithm $\mathsf{ID.Ext}$ to generate the private key d_i corresponding to ID_i. It sends d_i to the adversary.
- Decryption query $\langle ID_i, C_i \rangle$. The challenger generates the private key d_i. It then runs $\mathsf{ID.Dec}$ to decrypt C_i under ID_i.

These queries may be asked adaptively, that is, each query may depend on the answers obtained to the previous queries.

Challenge The adversary outputs equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and an identity ID_{ch}. The only constraint is that the private key for ID_{ch} was not requested in Phase 1. The challenger picks $b \leftarrow \{0,1\}$ and sets $C = \mathsf{ID.Enc}(\mathtt{ID.pms}, ID_{\mathsf{ch}}, M_b)$. It sends C to the adversary.

Phase 2 The adversary issues extraction and decryption queries as in Phase 1, with the restriction $\langle ID_i \rangle \neq \langle ID_{\mathsf{ch}} \rangle$ and $\langle ID_i, C_i \rangle \neq \langle ID_{\mathsf{ch}}, C \rangle$.

Guess The adversary outputs a guess $b' \in \{0,1\}$.

Such an adversary is called an $\mathsf{IND\text{-}ID\text{-}CCA}$ adversary $\mathcal{A}$, and its advantage is defined as $\mathsf{Adv}_{\mathcal{E},\mathcal{A}}^{\mathsf{ID-CCA}}(1^k) = |\Pr[b = b'] - 1/2|$.

Definition 1. *An* IBE *system $\mathcal{E}$ is* secure under chosen ciphertext attacks *if for any probabilistic polynomial time* $\mathsf{IND\text{-}ID\text{-}CCA}$ *adversary $\mathcal{A}$ the function* $\mathsf{Adv}_{\mathcal{E},\mathcal{A}}^{\mathsf{CCA}}(1^k)$ *is negligible.*

3 Definitions for certificate-based encryption

A certificate-based encryption scheme is a tuple of five PPT algorithms:

- $\mathsf{CB.Gen}$ is a probabilistic algorithm taking as input a security parameter k. It returns $\mathtt{CB.msk}$ (the certifier's master-key) and public parameters $\mathtt{CB.pms}$ that include the description of a string space Λ. Usually this algorithm is run by the CA. The system parameters include the description of sets $\mathcal{M}, \mathcal{C}$, which denote the set of messages and ciphertexts respectively.
- $\mathsf{CB.SetKeyPair}$ is a probabilistic algorithm that takes $\mathtt{CB.pms}$ as input[4]. It returns a pair public key - private key (PK, SK).
- $\mathsf{CB.Certify}$ is an algorithm that takes as input $\langle \mathtt{CB.msk}, \mathtt{CB.pms}, i, \mathtt{user}, PK \rangle$. It returns $Cert_i$, which is sent to the client. Here i identifies i-th time period, while $\mathtt{user} \in \Lambda$ contains other information needed to certify the client such as the client's identifying information, and PK is a public key.

[4] Actually, in the CBE generic construction by Yum and Lee [15], it is additionally assumed that $\mathtt{user}$ is also part of the input.

- $\mathsf{CB.Enc}$ is a probabilistic algorithm taking as inputs $\langle \texttt{CB.pms}, M, i, \texttt{user}, PK \rangle$ where $M \in \mathcal{M}$ is a message. It returns a ciphertext $C \in \mathcal{C}$ for message M or $\perp$ if PK is not a valid public key.
- $\mathsf{CB.Dec}$ is a deterministic algorithm taking as inputs $\langle \texttt{CB.pms}, Cert_i, SK, C \rangle$ as input in time period i. It returns either a message $M \in \mathcal{M}$ or the special symbol $\perp$ indicating a decryption failure.

Naturally, we require that if C is the result of applying algorithm $\mathsf{CB.Enc}$ with input $\langle \texttt{CB.pms}, M, i, \texttt{user}, PK \rangle$ and (PK, SK) is a valid key-pair, then M is the result of applying algorithm $\mathsf{CB.Dec}$ on input $\langle \texttt{CB.pms}, Cert_i, SK, C \rangle$, where $Cert_i$ is the output of the $\mathsf{CB.Certify}$. We write

$$\mathsf{CB.Dec}\big(\texttt{CB.pms}, Cert_i, SK, \mathsf{CB.Enc}(\texttt{CB.pms}, M, i, \texttt{user}, PK)\big) = M.$$

3.1 Security

The security of a certificate-based encryption scheme is defined against two different types of adversaries. The Type I adversary $\mathcal{A}_I$ has no access to the master key, but may make certification queries and decryption queries. This adversary models the security against non-certified users and general eavesdroppers. Secondly, the Type II adversary $\mathcal{A}_{II}$ is equipped with the master key and models an eavesdropping CA. In the following we give the definitions corresponding to the second type of adversary, since this is the adversary for which the attack presented in this paper is successful. For the full security definition of a CBE scheme we refer the reader to [2], which slightly weakened the attack of the certifier on the original definition of [11], which was inconsistent with the concrete scheme that [11] itself presented.

CBE Game 2. Attack of the certifier

Setup The challenger runs $\mathsf{CB.Gen}$, gives $\texttt{CB.pms}$ and $\texttt{CB.msk}$ to the adversary $\mathcal{A}_{II}$. The challenger then runs $\mathsf{CB.SetKeyPair}$ to obtain a key-pair $\langle PK, SK \rangle$ and gives PK to the adversary $\mathcal{A}_{II}$

Phase 1 The adversary issues decryption queries $q_1, \ldots, q_m$ where each q_j is a decryption query $\langle i, \texttt{user}, PK, C \rangle$. On this query, the challenger generates $Cert_i$ by using algorithms $\mathsf{CB.Certify}$ with inputs $\langle \texttt{CB.msk}, \texttt{CB.pms}, i, \texttt{user}, PK \rangle$ and outputs $\mathsf{CB.Dec}_{Cert_i, SK}(C)$, else it returns $\perp$. These queries may be asked adaptively, that is, they may depend on the answers to previous queries.

Challenge On challenge query $\langle i^*, \texttt{user}^*, M_0, M_1 \rangle$, where $M_0, M_1 \in \mathcal{M}$ are of equal length, the challenger checks that $\texttt{user}^* \in \varLambda$. If so, it chooses a random bit b and returns $C^* = \mathsf{CB.Enc}_{i^*, \texttt{user}^*, PK^*}(M_b)$; else it returns $\perp$.

Phase 2 As in phase 1, with the restriction

$$\langle i, \texttt{user}, PK, C \rangle \neq \langle i^*, \texttt{user}^*, PK, C^* \rangle.$$

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

We define the advantage of an adversary $\mathcal{A}_{II}$ as

$$\mathsf{Adv}^{\mathsf{CBE-CCA}}_{\mathcal{E},\mathcal{A}_{II}}(1^k) = |\Pr[b = b'] - 1/2| \,.$$

Definition 2. *A* CBE *scheme is said to be* secure against adaptive chosen ciphertext attacks from the certification authority *if no probabilistic polynomially bounded adversary has non-negligible advantage in CBE Game 2.*

4 Certificate-less public key encryption definitions

A certificate-less public key encryption scheme is a tuple of seven PPT algorithms:

–CL.Gen is a probabilistic algorithm taking as input a security parameter k. It returns the system parameters CL.pms and CL.msk. The system parameters include the message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$.

–CL.PartialKey is a probabilistic algorithm that takes CL.pms, CL.msk and an identifier $ID_A \in \{0,1\}^*$ for entity A as inputs. It returns a partial private key D_A.

–CL.SecretVal is a probabilistic algorithm that takes as inputs[5] CL.pms and returns a secret value x_A.

–CL.SetPrivKey is a deterministic algorithm that takes as inputs CL.pms, D_A and x_A and returns S_A, a (full) private key.

–CL.SetPubKey is a deterministic algorithm taking as input CL.pms, x_A. It returns a public key P_A.

–CL.Enc is a probabilistic algorithm taking as inputs CL.pms, M, P_A, ID_A where $M \in \mathcal{M}$ is a message. It returns a ciphertext $C \in \mathcal{C}$ for message M or $\perp$ indicating a encryption failure.

–CL.Dec is a deterministic algorithm taking as inputs CL.pms, S_A, C. It returns either a message $M \in \mathcal{M}$ or the special symbol $\perp$ indicating a decryption failure.

Naturally, we require that if C is the result of applying algorithm CB.Enc with input CL.pms, P_A, ID_A, M, then M is the result of applying algorithm CB.Dec on input CL.pms, S_A, C. That is,

$$\mathsf{CB.Dec}\big(\mathtt{CL.pms}, S_A, \mathsf{CB.Enc}(\mathtt{CL.pms}, M, P_A, ID_A)\big) = M.$$

Algorithms CL.SetPrivKey and CL.SetPubKey are normally run by an entity A for itself, after running CL.SecretVal. Usually A is the only entity in possession of S_A and x_A. Algorithms CL.Gen and CL.PartialKey are usually run by a trusted authority, called key generation center (KGC).

[5] Actually, in the CL-PKE generic constructions by Yum and Lee [15, 16], it is additionally assumed that ID_A is also part of the input.

4.1 Security

The security of a certificate-less encryption scheme is defined against two different types of adversaries. The Type I adversary $\mathcal{A}_I$ has no access to the master-key $\mathtt{CL.msk}$, but may replace public keys, extract partial private and private keys, and make decryption queries. This adversary models a non-registered user and a general eavesdropper. The Type II adversary $\mathcal{A}_{II}$ is equipped with the master-key and models an eavesdropping KGC. $\mathcal{A}_{II}$ is not allowed to replace public keys. In the following we give the definitions corresponding to the second type of adversary, since the attack we describe in this paper is carried out by the KGC. We stress that Yum and Lee security model for this adversary is unchanged from [1].

CL Game 2. Attack of a type II adversary

Setup The challenger runs $\mathsf{CL.Gen}$, and gives $\mathtt{CL.pms}$ and $\mathtt{CL.msk}$ to the adversary $\mathcal{A}_{II}$.

Phase 1 The adversary issues queries $q_1, \ldots, q_m$ where each q_j is one of public key, private key and decryption query.

Challenge On challenge query $\langle ID_{\mathsf{ch}}, M_0, M_1 \rangle$, where $M_0, M_1 \in \mathcal{M}$ are of equal length and the private key of ID_{ch} was not queried in phase 1, the challenger chooses a random bit b and returns $C^* = \mathsf{CL.Enc}(M_b)$ the encryption of M_b under the current public key P_{ch} for ID_{ch}. Then C^* is delivered to the adversary.

Phase 2 As in phase 1, except that $\mathcal{A}_{II}$ can not make a decryption query on the challenge ciphertext C^* for $(ID_{\mathsf{ch}}, P_{\mathsf{ch}})$ nor a private key query on the challenge identity ID_{ch}.

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

We define the advantage of an adversary $\mathcal{A}_{II}$ in CL Game 2 as $\mathsf{Adv}^{\mathsf{CL-CCA}}_{\mathcal{E},\mathcal{A}_{II}}(1^k) = |\Pr[b = b'] - 1/2|$.

Definition 3. *A* CL-PKE *scheme is said to be* secure against adaptive chosen ciphertext attacks from the key generation center *if no probabilistic polynomially bounded adversary has non-negligible advantage in CL Game 2.*

5 An attack against the generic construction for CBE from EuroPKI 2004

At EuroPKI 2004, Yum and Lee [16] proposed a generic construction for IND-CBE-CCA certificate-based encryption schemes from IND-ID-CCA identity-based encryption schemes. Their construction is depicted in Figure 1. The main idea of their construction is to use double encryption with respect to IBE. One of the decryption keys is known by the certifier, while the other decryption key is only known to the user. Unfortunately, the double encryption design used in [16] is insecure in the light of [10].

```
$\mathsf{CB.Gen}(1^k)$
  $(\mathtt{ID.pms}_{CA}, \mathtt{ID.msk}_{CA}) \leftarrow \mathsf{ID.Gen}(1^k)$
  $\mathtt{CB.msk} \leftarrow \mathtt{ID.msk}_{CA}$
  $\mathtt{CB.pms} \leftarrow \mathtt{ID.pms}_{CA}$
  Return $(\mathtt{CB.pms}, \mathtt{CB.msk})$

$\mathsf{CB.SetKeyPair}(\mathtt{CB.pms}, \mathtt{user})$
  $(\mathtt{ID.pms}_U, \mathtt{ID.msk}_U) \leftarrow \mathsf{ID.Gen}(1^k)$
  $d_U \leftarrow \mathsf{ID.Ext}(\mathtt{ID.pms}_U, \mathtt{ID.msk}_U, \mathtt{user})$
  $SK_U \leftarrow (d_U, \mathtt{ID.pms}_U)$
  $PK_U \leftarrow \mathtt{ID.pms}_U$
  Return $(PK_U, SK_U)$

$\mathsf{CB.Certify}(\mathtt{CB.msk}, \mathtt{CB.pms}, i, \mathtt{user}, PK_U)$
  $Cert_i^U$
  $\leftarrow \mathsf{ID.Ext}\Big(\mathtt{CB.pms}, \mathtt{CB.msk}, (\mathtt{user}, i, PK_U)\Big)$
  Return $Cert_i^U$

$\mathsf{CB.Enc}(\mathtt{CB.pms}, M, i, \mathtt{user}, PK_U)$
  $C' \leftarrow \mathsf{ID.Enc}(PK_U, \mathtt{user}, M)$
  $C \leftarrow \mathsf{ID.Enc}(\mathtt{CB.pms}, (\mathtt{user}, i, PK_U), C')$
  Return $C$
$\mathsf{CB.Dec}(\mathtt{CB.pms}, Cert_i^U, SK_U, C)$
  Parse $SK_U$ as $(d_U, \mathtt{ID.pms}_U)$
  $C' \leftarrow \mathsf{ID.Dec}(\mathtt{CB.pms}, Cert_i^U, C)$
  $M \leftarrow \mathsf{ID.Dec}(PK_U, d_U, C')$
  Return $M$
```

Fig. 1. Yum-Lee transformation from IBE to CBE.

We note that this construction does not achieve the required security for certificate-based schemes, at least in the case of an attack of the certifier, as defined in Section 3.1. Remember that the certifier is equipped with his own secret key CB.msk and that it is allowed to make decryption queries, with the natural limitation that he cannot ask for the challenge ciphertext. The attack begins once the certifier (called adversary $\mathcal{A}_{II}$ in the CBE game 2) obtains the challenge ciphertext $C^* = \mathsf{CB.Enc}(\mathtt{CB.pms}, M_b, i^*, \mathtt{user}^*, PK_U^*)$ for M_0, M_1 and unknown $b \in \{0,1\}$ chosen by the challenger. The attack works as follows:

1. $\mathcal{A}_{II}$ generates the certificate $Cert_{i^*}^U$ for $\mathtt{user}^*, i^*, PK_U^*$ by running
$$\mathsf{CB.Certify}(\mathtt{CB.msk}, \mathtt{CB.pms}, i^*, \mathtt{user}^*, PK_U^*).$$
2. This certificate is used to decrypt and obtain $C' \leftarrow \mathsf{ID.Dec}(\mathtt{CB.pms}, Cert_{i^*}^U, C^*)$.
3. Since ID.Enc is a probabilistic algorithm, $\mathcal{A}_{II}$ reencrypts C' until obtains $C'' = \mathsf{ID.Enc}(\mathtt{CB.pms}, (\mathtt{user}^*, i^*, PK_U^*), C')$ such that $C'' \neq C^*$.
4. $\mathcal{A}_{II}$ asks the decryption oracle for the decryption of C''. Since $C'' \neq C^*$, this is a valid decryption query and $\mathcal{A}_{II}$ gets back M_b.

The advantage of this adversary is $1/2$, so the scheme in Figure 1 is not secure in the sense of against adaptive chosen ciphertext attacks from the certification authority.

This attack can be easily avoided following [10]. In fact, the proposal of [10] for a generic construction of CBE is very similar to [15]. The main difference is that it uses parallel encryption instead of sequential encryption, but the idea to obtain full security are the same. Informally, this idea is to use the verifier's key of a one-time signature scheme as a label when encrypting and then sign the whole ciphertext. The non-malleability of the ciphertext and the security of the signature scheme prevent the attack from being successful.

6 An attack against Yum and Lee generic constructions for CL-PKE schemes

In the same paper [16], Yum and Lee also gave a generic transformation from IBE to CL-PKE. The security model they considered for CL-PKC is much more restricted than the original one of [2]. The transformation [16] presented is depicted in Figure 2. In the same vein as in the previous construction, a double identity-based encryption mechanism is used. One of the decryption keys is known by the key generation center, while the other decryption key is only known to the user. Unfortunately, the double encryption is done with the naive technique described in [10], which is insecure even in the weaker security model considered by [16].

```
$\mathsf{CL.Gen}(1^k)$
  $(\texttt{ID.pms}_{KGC}, \texttt{ID.msk}_{KGC}) \leftarrow \mathsf{ID.Gen}(1^k)$
  $\texttt{CL.msk} \leftarrow \texttt{ID.msk}_{KGC}$
  $\texttt{CL.pms} \leftarrow \texttt{ID.pms}_{KGC}$
  Return $(\texttt{CL.pms}, \texttt{CL.msk})$

$\mathsf{CL.PartialKey}(\texttt{CL.pms}, \texttt{CL.msk}, ID_A)$
  $d_A \leftarrow \mathsf{ID.Ext}(\texttt{CL.pms}, \texttt{CL.msk}, ID_A)$
  $D_A \leftarrow (d_A, ID_A)$
  Return $D_A$

$\mathsf{CL.SecretVal}(\texttt{CL.pms}, ID_A)$
  $(\texttt{ID.pms}_A, \texttt{ID.msk}_A) \leftarrow \mathsf{ID.Gen}(1^k)$
  $x_A \leftarrow (\texttt{ID.pms}_A, \texttt{ID.msk}_A, ID_A)$
  Return $x_A$

$\mathsf{CL.SetPrivKey}(\texttt{CL.pms}, D_A, x_A)$
  Parse $x_A$ as $(\texttt{ID.pms}_A, \texttt{ID.msk}_A, ID_A)$
  Parse $D_A$ as $(d_A, ID_A)$
  $d'_A \leftarrow \mathsf{ID.Ext}(\texttt{ID.pms}_A, \texttt{ID.msk}_A, ID_A)$
  $S_A \leftarrow (d_A, d'_A, \texttt{ID.pms}_A)$
  Return $S_A$

$\mathsf{CL.SetPubKey}(\texttt{CL.pms}, x_A)$
  Parse $x_A$ as $(\texttt{ID.pms}_A, \texttt{ID.msk}_A, ID_A)$
  $P_A \leftarrow \texttt{ID.pms}_A$
  Return $P_A$

$\mathsf{CL.Enc}(\texttt{CL.pms}, M, P_A, ID_A)$
  $C' \leftarrow \mathsf{ID.Enc}(P_A, ID_A, M)$
  $C \leftarrow \mathsf{ID.Enc}(\texttt{CL.pms}, ID_A, C')$
  Return $C$

$\mathsf{CB.Dec}(\texttt{CL.pms}, S_A, C)$
  Parse $S_A$ as $(d_A, d'_A, \texttt{ID.pms}_A)$
  $C' \leftarrow \mathsf{ID.Dec}(\texttt{CL.pms}, d_A, C)$
  $M \leftarrow \mathsf{ID.Dec}(\texttt{ID.pms}_A, d'_A, C')$
  Return $M$
```

Fig. 2. Yum-Lee transformation from IBE to CL-PKE.

Indeed, it is not hard to see that their construction suffers from exactly the same problem as the one for certificate-based encryption and that the attack of the Type II adversary succeeds for exactly the same reason. The attack begins once the adversary $\mathcal{A}_{II}$ in the CL Game 2 described in Section 4.1 obtains the challenge ciphertext $\mathsf{CL.Enc}(\texttt{CL.pms}, M_b, P^*_A, ID^*_A)$ for M_0, M_1 and unknown $b \in \{0,1\}$ chosen by the challenger. The attack works as follows:

1. Since the challenger has given $\mathcal{A}_{II}$ the KGC's master-key $\texttt{CB.msk}$, the adversary can generate the partial private key $D_A^* = (d_A^*, ID_A^*)$ for user ID_A^* by running $D_A^* \leftarrow \mathsf{CL.PartialKey}(\texttt{CL.pms}, \texttt{CL.msk}, ID_A^*)$.
2. This partial private key is used to decrypt and obtain

$$C' \leftarrow \mathsf{ID.Dec}(\texttt{CL.pms}, d_A^*, C^*).$$

3. Since $\mathsf{ID.Enc}$ is a probabilistic algorithm, $\mathcal{A}_{II}$ reencrypts C' until obtains $C'' \leftarrow \mathsf{ID.Enc}(P_A^*, ID_A^*, C')$ such that $C'' \neq C^*$.
4. $\mathcal{A}_{II}$ asks the decryption oracle for the decryption of C''. Since $C'' \neq C^*$, this is a valid decryption query and $\mathcal{A}_{II}$ gets back M_b.

The advantage of this adversary is 1/2, so the scheme in Figure 2 is not secure in the sense of against adaptive chosen ciphertext attacks from the key generation center.

In the work [15], the authors give another transformation from identity-based encryption to certificate-less encryption. In this case, the user employs a traditional public key encryption scheme [4] instead of an identity-based encryption scheme. The rest of the construction exactly resembles the one described in the previous figure and therefore the attack just presented also applies to [15].

The construction in [15] is also criticized in [13], where a similar attack is proposed. However, the attack in [13] is for a type I adversary and only works in the full model, since this adversary is significantly weakened in the work of [15].

References

1. S. Al-Riyami and K.G. Paterson. Certificateless public key cryptography. In *Advances in Cryptology - ASIACRYPT 2003*, LNCS vol. 2894, pp. 452-473, Springer-Verlag, 2003.
2. S. Al-Riyami and K.G. Paterson. CBE from CL-PKE: A generic construction and efficient scheme. *Public Key Cryptography - PKC 2005*, LNCS vol. 3386, pp. 398-415, Springer-Verlag, 2005.
3. J. Baek, R. Safavi-Naini and W. Susilo. Certificateless Public Key Encryption Without Pairing. *Information Security Conference - ISC 2005*, LNCS vol. 3650, pp. 134-148, Springer-Verlag, 2005.
4. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. *Advances in Cryptology - CRYPTO 1998*, LNCS vol. 1462, pp. 26-45, Springer-Verlag, 1998.
5. K. Bentahar and P. Farshim and J. Malone-Lee and N.P. Smart. Generic Constructions of Identity-Based and Certificateless KEMs. Cryptology ePrint Archive, Report 2005/058.
6. D. Boneh and M. Franklin. Identity-Based Encryption ¿From The Weil Pairing, *Advances in Cryptology - Crypto 2001*, LNCS vol. 2139, pp.213-229, Springer-Verlag, 2001.
7. M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. *ACM CCS 93*, pp. 62-73, ACM Press, 1993.
8. R. Canetti, O. Goldreich and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM* vol. 51 num. 4, pp. 557-594, ACM press, 2004.

9. A. Dent and C. Kudla. On Proofs of Security for Certificateless Cryptosystems. Cryptology ePrint Archive, Report 2005/348.
10. Y. Dodis and J. Katz. Chosen-Ciphertext Security of Multiple Encryption, *Theory of Cryptography Conference - TCC 2005*, LNCS vol. 3378, pp. 188-209, Springer-Verlag, 2005.
11. C. Gentry. Certificate-Based Encryption and the Certificate-Revocations Problem, *Advances in Cryptology - Eurocrypt 2003*, LNCS vol. 2656, pp. 272-291, Springer-Verlag, 2003.
12. B.G. Kang and J.H. Park. It is possible to have CBE from CL-PKE? Cryptology ePrint Archive, Report 2005/431, 2005. http://eprint.iacr.org/.
13. B. Libert and J.J. Quisquater. On Constructing Certificateless Cryptosystems from Identity Based Encryption. *Public Key Cryptography 2006 - PKC 2006*, LNCS, Springer-Verlag. To appear
14. A. Shamir. Identity-based cryptosystems and signature schemes, *Advances in Cryptology - Crypto 1984*, LNCS vol. 196, pp. 47-53, Springer-Verlag, 1985.
15. D.H. Yum and P.J. Lee. Generic Construction of Certificateless Encryption. In *Computational Science and Its Applications - ICCSA 2004*, LNCS vol. 3043, pp. 802 - 811, Springer-Verlag, 2004.
16. D.H. Yum and P.J. Lee. Identity-based cryptography in public key management. In *EuroPKI 2004*, LNCS vol. 3093, pp. 71-84, Springer-Verlag, 2004.