Repairing a Security-Mediated Certificateless Encryption Scheme from PKC 2006

Joonsang Baek and Guilin Wang

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{jsbaek,glwang}@i2r.a-star.edu.sg

Abstract. At PKC 2006, Chow, Boyd, and Nieto introduced the concept of security-mediated certificateless (SMC) cryptography. This notion can be considered as a variant of certificateless cryptography with the property of instantaneous key revocation, or a variant of mediated cryptography without full key escrow. They presented a definition of security for SMC encryption, which covers (fully-adaptive) chosen ciphertext attack with public key replacement considered as a strong but essential attack on certificateless cryptographic schemes. They proposed two SMC encryption schemes, one is a generic construction based on any public key encryption, identity-based encryption and one-time signature schemes and the other is a concrete construction based on bilinear pairings, which were shown to be secure under their security definition. In this note, we, however, present two types of attacks demonstrating that their generic construction for SMC encryption fails to meet their security requirement. We then discuss how to repair the scheme and provide a provably-secure solution.

1 Introduction

Security-mediated certificateless (SMC) cryptography, introduced at PKC 2006 by Chow, Boyd, and Nieto [3], aims to achieve the following properties simultaneously:

- No need of certification in contrast to the traditional PKI-based cryptography.
- No full private key escrow in contrast to the identity-based cryptography [2].
- Instant key revocation in contrast to the certificateless cryptography [1].

Chow et al. [3] presented a definition of security for SMC encryption, which covers the security against (adaptive) chosen ciphertext attack in which an adversary has an access to decryption oracles separately run by the Security Mediator (SEM) and the user; and the security against public key replacement attack [1] in which an adversary can replace the user's public key with a minimal restriction. They proposed two SMC encryption schemes claimed to be secure in their definition of security. In this note, we show that one of their schemes, which is

a generic construction based on secure public key encryption, identity-based encryption and one-time signature, in fact fails to meet their definition of security. Additionally, we discuss how to repair this scheme and provide a provably-secure solution.

2 Review of the CBN Scheme

Throughout this paper, we call the generic construction of SMC encryption given in [3] "CBN scheme" for simplicity.

In general, a SMC encryption scheme consists of seven algorithms, (Setup, Set-Private-Key, Set-Public-Key, Register-Public-Key, Encrypt, SEM-Decrypt, User-Decrypt), and involves three entities, the key generation center (KGC), the security mediator (SEM) and a set of users. Each of the algorithms and entities of the SMC encryption scheme works as follows:

On input a security parameter k, the KGC runs setup to generate system parameters params and a master-key. By running Set-Private-Key and Set-Public-Key, each user can generate his/her private and public key pair. Then, users are required to register their identities and the public keys with the KGC by running Register-Public-Key. For each registered identity, the KGC uses the master-key to generate and send the (private) SEM-key to the SEM in a secure way. Note that there may exist multiple SEMs, but each SEM is not given access to the master-key at any time. Encrypt is the encryption algorithm that outputs a ciphertext, when a message, an identity and a registered public key are given. SEM-Decrypt is executed by the SEM using the SEM-key to fulfill the partial decryption for a user. Finally, User-Decrypt takes this partial decryption result and the user's private key as inputs to recover the encrypted message.

In essence, the CBN scheme is constructed by adopting the multiple encryption technique proposed by Dodis and Katz [6]. More specifically, the CBN scheme uses an identity-based encryption scheme for the SEM side and a public key encryption scheme for the user side. Let PKE = (PKE.Gen, PKE.Enc, PKE.Dec) be a generic public key encryption scheme, IBE=(IBE.Set, IBE.Gen, IBE.Enc, IBE.Dec) an identity-based encryption scheme, and S= (SGen, Sig, Vfy) a one-time signature scheme. Then, the CBN scheme can be described as follows.

- Setup: On input a security parameter k, execute IBE.Set to generate system parameters params and the master-key. In addition, a function H is selected from a family of collision-resistant hash functions.
- Set-Private-Key and Set-Public-Key: On input a security parameter k, execute PKE.Gen to generate the user's public/private key pair (EK,DK).
- Register-Public-Key: On input the public key EK and an identity ID_A and the master secret master-key, the SEM-key for a user A is set as $D_A = \mathsf{IBE.Gen}_{\mathsf{params}}(\mathsf{ID_A})$. As part of the registration procedure, A is assumed to prove his/her knowledge of the private key DK corresponding to the registered public key EK.
- Encrypt: Inputs are a message $M \in \{0,1\}^n$, an identity ID_A and public key EK.

1. Generate one-time signature key pair (SK,VK) using SGen.
2. Choose a random label ℓ.
3. Choose $s_1 \in \{0,1\}^n$ at random and set $s_2 = M \oplus s_1$.
4. Compute $C_1 = \mathsf{IBE.Enc}_{\mathsf{params}}(\mathsf{ID}_A, s_1)$.
5. Compute $C_2 = \mathsf{PKE.Enc}^{\ell}_{\mathsf{EK}}(s_2)$. (Note that the superscript l in $\mathsf{PKE.Enc}^{\ell}_{\mathsf{EK}}(s_2)$ means that the label l is provided as input to $\mathsf{PKE.Enc}_{\mathsf{EK}}$).
6. Compute $\alpha = H(C_1, C_2, \ell)$.
7. Compute an one-time signature $\sigma = \mathsf{Sig}_{\mathsf{SK}}(\alpha)$.
8. Output a ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma, \ell)$.

– SEM-Decrypt: Inputs are a ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma, \ell)$, an identity ID_A and SEM-key D_A.
1. Check that ID_A is a legitimate user whose key is not revoked.
2. Compute $\alpha = H(C_1, C_2, \ell)$.
3. Check that σ is a valid one-time signature on α by $\mathsf{Vfy}_{\mathsf{VK}}(\alpha, \sigma)$.
4. Output $\perp$ if verification fails.
5. Otherwise, output $V_1' = \mathsf{IBE.Dec}_{D_A}(C_1)$.

– User-Decrypt: Inputs are a ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma, \ell)$, the token V_1' from the SEM, and a secret DK.
1. Compute α and check the validity of σ as in SEM-Decrypt.
2. Output $\perp$ if verification fails.
3. Otherwise, compute $V_2' = \mathsf{PKE.Dec}^{\ell}_{\mathsf{DK}}(C_2)$.
4. Output the message $M' = V_1' \oplus V_2'$.

3 Two Attacks on the CBN Scheme

Following the definition of security for CL-PKE [1], Chow et al. presented a definition of security for SMC encryption. As mentioned earlier, this definition includes the security against fully-adaptive chosen ciphertext attack (CCA) and the security against public key replacement attack. More precisely, the following two types of adversary for a public key key replacement attack are considered.

– Type-I adversary does not have access to the master-key, but is allowed to choose any public key to be used for the challenge ciphertext (and enquire related oracles).
– Type-II adversary has access to the master-key (and related oracles), but only a registered public key can be used for the challenge ciphertext.

Similar to the CL-PKE model, the adversary can extract SEM-key by sending an identity but is forbidden from getting SEM-key for the identity and replacing the public key for the same identity (Readers are referred to Section 4 of [3] for details).

In [3], it is claimed that if both the underlying building blocks IBE and PKE are secure (Note, however, that it was not clearly stated that in what sense they are secure.), the CBN scheme is also secure against both Type-I and Type-II

adversaries. However, they do not provide a full proof for this. – They simply provide a sketch for some parts of simulations.

In what follows, we demonstrate two simple chosen ciphertext attacks which show the claim on the security of the CBN scheme is in fact incorrect. The validity of those two attacks can be verified easily.

Chosen Ciphertext Attack with Public Key Replacement. A Type-I adversary B performs the following.

1) Generate a valid public and private key pair $(\mathsf{EK}, \mathsf{DK})$ and replace the user A's public key (associated with her identity ID_A) with EK. At the same time, output two random strings M_0 and M_1 as challenge messages.
2) Upon receiving a challenge ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma, \ell)$ which encrypts M_b where b is uniformly chosen at random from $\{0, 1\}$, decrypt C_2 using DK to obtain s_2.
3) Create $C' = (C_1, C_2', \mathsf{VK}', \sigma', \ell')$ by randomly selecting a new one-time signature key pair $(\mathsf{SK}', \mathsf{VK}')$, a new label $\ell' (\neq \ell)$ and a new string $s_2' (\neq s_2)$; and computing $C_2' = \mathsf{PKE.Enc}^{\ell'}_{\mathsf{EK}'}(s_2')$, $\sigma' = \mathsf{Sig}_{\mathsf{SK}'}(H(C_1, C_2', \ell'))$.
4) Send C' to the SEM-decryption oracle as a **SEM decryption** query to obtain s_1.
5) Compute $M_b \equiv s_1 \oplus s_2$.

Notice that the success probability of the above adversary is 1. Actually, one can break the chosen ciphertext security of the CBN scheme without replacing the public key of the target user.

Chosen Ciphertext Attack without Public Key Replacement. An adversary B performs the following.

1) Create a new one-time signature key pair $(\mathsf{SK}', \mathsf{VK}')$, where $\mathsf{SK}' \neq \mathsf{SK}$ and $\mathsf{VK}' \neq \mathsf{VK}$.
2) Upon receiving a challenge ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma, \ell)$ which encrypts M_b where b is uniformly chosen at random from $\{0, 1\}$, modify $C = (C_1, C_2, \mathsf{VK}, \sigma, \ell)$ into $C' = (C_1, C_2, \mathsf{VK}', \sigma', \ell)$, where $\sigma' = \mathsf{Sig}_{\mathsf{SK}'}(H(C_1, C_2, \ell))$. (Notice that $C \neq C'$ as $\mathsf{VK}' \neq \mathsf{VK}$).
3) Send C' to the SEM-decryption oracle as a **SEM decryption** query to obtain a decryption V_1'. (Notice that V_1' is actually equal to s_1).
4) Send (C', V_1') to the User-decryption oracle as a **User decryption** query to obtain a decryption V_2'. (Notice that V_2' is actually equal to s_2).
5) Compute $M_b \equiv V_1' \oplus V_2'$.

4 Discussion

As noted in [3], the CBN scheme employs the technique used by Dodis and Katz [6] to give multiple encryption schemes chosen ciphertext security. However, we note that Dodis and Katz's technique is used incorrectly in the CBN scheme since the identity-based encryption C_1 and the public key encryption C_2 are not

properly bound together with the one-time signature σ and the verification key VK as shown in the above attack. We also note that the label l also does not play a proper role of preventing the malleability of ciphertext.

Using the verification key VK of the one-time signature as a label, we show that one can fix the CBN scheme. (This technique is actually presented in [6]). Note that the identity-based encryption and public key encryption schemes employed in the following construction satisfies $\mathsf{IBE.Dec}_{D_A}^{\mathsf{VK}}(\mathsf{IBE.Enc}_{\mathsf{params}}^{\mathsf{VK}}(\mathsf{ID}_A, M)) = M$ and $\mathsf{PKE.Dec}_{\mathsf{DK}}^{\mathsf{VK}}(\mathsf{PKE.Enc}_{\mathsf{EK}}^{\mathsf{VK}}(M)) = M$ respectively for any message M and label VK. Note also that Setup, Set-Private-Key, Set-Public-Key and Register-Public-Key algorithms are the same as those of the original CBN scheme.

- Encrypt: Inputs are a message $M \in \{0,1\}^n$, an identity ID_A and public key EK.
  1. Generate an one-time signature key pair (SK,VK) using SGen.
  2. Choose random $s_1 \in \{0,1\}^n$ and set $s_2 = M \oplus s_1$.
  3. Compute $C_1 = \mathsf{IBE.Enc}_{\mathsf{params}}^{\mathsf{VK}}(\mathsf{ID}_A, s_1)$.
  4. Compute $C_2 = \mathsf{PKE.Enc}_{\mathsf{EK}}^{\mathsf{VK}}(s_2)$.
  5. Compute an one-time signature $\sigma = \mathsf{Sig}_{\mathsf{SK}}(C_1, C_2)$.
  6. Output a ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma)$.
- SEM-Decrypt: Inputs are a ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma)$, an identity ID_A and SEM-key D_A.
  1. Check that ID_A is a legitimate user whose key is not revoked.
  2. Check that σ is a valid one-time signature on (C_1, C_2) with respect to VK by $\mathsf{Vfy}_{\mathsf{VK}}(\alpha, \sigma)$.
  3. Output $\perp$ if verification fails.
  4. Otherwise, output $V_1 = \mathsf{IBE.Dec}_{D_A}^{\mathsf{VK}}(C_1)$.
- User-Decrypt: Inputs are a ciphertext $C = (C_1, C_2, \mathsf{VK}, \sigma)$, the token V_1 from the SEM, and a secret DK.
  1. Check the validity of σ as in SEM-Decrypt.
  2. Output $\perp$ if verification fails.
  3. Otherwise, compute $V_2 = \mathsf{PKE.Dec}_{\mathsf{DK}}^{\mathsf{VK}}(C_2)$.
  4. Output the message $M = V_1 \oplus V_2$.

We emphasize that the important security requirement for the underlying public key and identity-based encryption schemes is that *each of their ciphertexts and labels should be bound in a non-malleable way.* – The use of labels in public key encryption is further elaborated in Section 7.2 of [7]).

It is not hard to see the above construction is secure assuming that an adversary for this scheme does not perform the public key replacement attack and the underlying primitives satisfy some strong security requirements. More precisely, we prove the following theorem.

Theorem 1. *The above modified CBN scheme is secure against chosen ciphertext attack without public key replacement if the underlying one-time signature scheme is unforgeable and the underlying identity-based encryption and public*

key schemes are semantically secure against chosen ciphertext attack. (The chosen ciphertext security of identity-based encryption here includes the attack where an adversary can make private key extraction queries, which is referred to as IND-ID-CCA in [2]).

Proof. Let B be an adversary that breaks the CCA-security of the modified CBN scheme. We assume that B does not conduct public key replacement attack. Let B' and B'' denote CCA-adversaries for the underlying public key encryption and identity-based encryption schemes used in the modified CBN scheme respectively. Also, let F denote an adversary (forger) for the one-time signature scheme.

Construction of B' (for breaking the underlying public key encryption). Suppose that B' is given a public key EK. B' runs $\mathsf{IBE.Set}$ to generate system parameters params and the $\mathsf{master\text{-}key}$ for the underlying identity-based encryption scheme. B' responds to B's queries as follows.

If B queries identities as private key extraction queries, B' generates corresponding private keys using the $\mathsf{master\text{-}key}$ and responds to B with them accordingly.

If B queries $((C'_1, C'_2, \mathsf{VK}', \sigma'), \mathsf{ID}'_A)$ to the $\mathsf{SEM\text{-}Decrypt}$ oracle, B' first checks whether σ' is a valid one-time signature on (C'_1, C'_2) with respect to VK'. If it is, B' generates a private key D'_A for the identity ID'_A using $\mathsf{master\text{-}key}$, decrypt C'_1 using D'_A and sends the result (decryption) back to B. Otherwise, B' rejects the query.

If B queries $((C'_1, C'_2, \mathsf{VK}', \sigma'), V'_1)$ to the $\mathsf{User\text{-}Decrypt}$ oracle, B' first checks whether σ' is a valid one-time signature on (C'_1, C'_2) with respect to VK'. If it is, B' forwards C'_2 as a decryption query to its decryption oracle to get a decryption V'_2, compute $M' = V'_1 \oplus V'_2$ and sends it back to B. Otherwise, B' rejects the query.

At some stage of the attack, B submits two equal-length plaintexts M_0, M_1 and an identity $\mathsf{ID_A}$. Upon receiving these, B' creates a challenge ciphertext for B as follows: B' selects a random string s_1 whose length is equal to M_0 (and hence M_1), computes $C_1 = \mathsf{IBE.Enc}^{\mathsf{VK}}_{\mathsf{params}}(\mathsf{ID}_A, s_1)$, computes $M'_0 = M_0 \oplus s_1$ and $M'_1 = M_1 \oplus s_1$, creates a one-time signature key pair $(\mathsf{SK}, \mathsf{VK})$ and queries M'_0 and M'_1 together with VK to its challenger to get $C_2 = \mathsf{PKE.Enc}^{\mathsf{VK}}_{\mathsf{EK}}(M'_b)$. B' also computes $\sigma = \mathsf{Sig}_{\mathsf{SK}}(C_1, C_2)$. Finally B' returns $C = (C_1, C_2, \mathsf{VK}, \sigma)$ as a challenge ciphertext for B.

B continues to make queries, each of which is of the form $(C', \mathsf{ID}'_A)(\neq (C, \mathsf{ID}_A))$, where $C' = (C'_1, C'_2, \mathsf{VK}', \sigma')$, to the $\mathsf{SEM\text{-}Decrypt}$ oracle. Upon receiving each query, B' does the following.

- If $\mathsf{VK}' \neq \mathsf{VK}$, B' rejects the query straight away.
- Else B' first checks whether σ' is a valid one-time signature on (C'_1, C'_2) with respect to VK'. If it is, B' generates a private key D'_A for the identity ID'_A using $\mathsf{master\text{-}key}$, decrypt C'_1 using D'_A and sends the result back to B. Otherwise, reject the query.

If B makes queries, each of which is of the form (C', V_1') where $C' = (C_1', C_2', \mathsf{VK}', \sigma')$ and $C' \neq C$ to the $\mathsf{User\text{-}Decrypt}$ oracle, B' performs the following.

- If $\mathsf{VK}' \neq \mathsf{VK}$, B' rejects the query straight away.
- Else if $\mathsf{VK}' = \mathsf{VK}$ and $C_2' \neq C_2$, B' first checks whether σ' is a valid one-time signature on (C_1', C_2') with respect to VK'. If it is, B' forwards C_2' as a decryption query to its decryption oracle to get a decryption $V_2' = \mathsf{PKE.Dec}_{\mathsf{DK}}^{\mathsf{VK}}(C_2')$ and sends $V_1' \oplus V_2'$ back to B. Otherwise, B' rejects the query.
- Else if $\mathsf{VK}' = \mathsf{VK}$ and $C_2' = C_2$, B' first checks whether σ' is a valid one-time signature on (C_1', C_2') with respect to VK'. If it is, B' stops the simulation. Otherwise, B' rejects the query. – $(*)$

When B outputs its guess $b' \in \{0, 1\}$, B' also outputs b' for its guess.

Construction of B'' (for breaking the underlying identity-based encryption). Suppose that B'' is given params of the identity-based encryption scheme. B'' runs $\mathsf{PKE.Gen}$ to generate a public/private key (EK,DK) for the underlying public key encryption scheme. B'' responds to B's queries as follows.

If B queries identities as private key extraction queries, B'' forwards them as private key extraction queries to its private key extraction oracle to get corresponding private keys and send back them to B.

If B queries $((C_1', C_2', \mathsf{VK}', \sigma'), \mathsf{ID}_A')$ to the $\mathsf{SEM\text{-}Decrypt}$ oracle, B'' first checks whether σ' is a valid one-time signature on (C_1', C_2') with respect to VK'. If it is, B'' forwards ID_A' as a private key extraction query to its private key extraction oracle to get a private key D_A', decrypt C_1' using D_A' and sends the result back to B. Otherwise, B'' rejects the query.

If B queries $((C_1', C_2', \mathsf{VK}', \sigma'), V_1')$ to the $\mathsf{User\text{-}Decrypt}$ oracle, B'' first checks whether σ' is a valid one-time signature on (C_1', C_2') with respect to VK'. If it is, B'' decrypts C_2' using the key DK to get a decryption V_2', compute $M' = V_1' \oplus V_2'$ and sends it back to B. Otherwise, B'' rejects the query.

Upon receiving two equal-length plaintexts M_0, M_1 and an identity $\mathsf{ID_A}$ from B, B'' creates a challenge ciphertext for B as follows: B'' first picks a random string s_2 whose length is equal to M_0 (and hence M_1) and computes $s_{1_0} = M_0 \oplus s_2$ and $s_{1_1} = M_1 \oplus s_2$. B'' then submits s_{1_0}, s_{1_1} and ID_A to its challenger to get $C_1 = \mathsf{IBE.Enc}_{\mathsf{params}}^{\mathsf{VK}}(\mathsf{ID}_A, s_{1_b})$ for $b \in \{0, 1\}$ chosen at random. B'' also creates a one-time signature key pair $(\mathsf{SK}, \mathsf{VK})$ and generates $C_2 = \mathsf{PKE.Enc}_{\mathsf{EK}}^{\mathsf{VK}}(s_2)$. B'' also computes $\sigma = \mathsf{Sig}_{\mathsf{SK}}(C_1, C_2)$. Finally B'' returns $C = (C_1, C_2, \mathsf{VK}, \sigma)$ as a challenge ciphertext for B.

B continues to make queries, each of which is of the form $(C', \mathsf{ID}_A')(\neq (C, \mathsf{ID}_A))$, where $C' = (C_1', C_2', \mathsf{VK}', \sigma')$, to the $\mathsf{SEM\text{-}Decrypt}$ oracle. Upon receiving each query, B'' does the following.

- If $\mathsf{VK}' \neq \mathsf{VK}$, B'' rejects the query straight away.
- Else if $\mathsf{VK}' = \mathsf{VK}$ and $\mathsf{ID}_A' \neq \mathsf{ID}_A$, B'' first checks whether σ' is a valid one-time signature on (C_1', C_2') with respect to VK'. If it is, B'' forwards ID_A' as a private key extraction query to its private key extraction oracle to get

a private key D'_A, decrypt C'_1 using D'_A and sends the result back to B. Otherwise, reject the query.
– Else if $\mathsf{VK}' = \mathsf{VK}$, $\mathsf{ID}'_A = \mathsf{ID}_A$ and $C'_1 = C_1$, B' first checks whether σ' is a valid one-time signature on (C'_1, C'_2) with respect to VK'. If it is, B'' stops the simulation. Otherwise, B' rejects the query. – $(**)$

If B makes queries, each of which is of the form (C', V'_1) where $C' = (C'_1, C'_2, \mathsf{VK}', \sigma')$ and $C' \neq C$ to the User-Decrypt oracle, B'' performs the following.

– If $\mathsf{VK}' \neq \mathsf{VK}$, B'' rejects the query straight away.
– Else B'' first checks whether σ' is a valid one-time signature on (C'_1, C'_2) with respect to VK'. If it is, B'' decrypts C'_2 to get a decryption V'_2 and sends $V'_1 \oplus V'_2$ back to B. Otherwise, B'' rejects the query.

When B outputs its guess $b' \in \{0, 1\}$, B'' also outputs b' for its guess.

Construction of F (for breaking the underlying one-time signature). Suppose that F is given a verification key (VK). F runs $\mathsf{IBE.Set}$ to generate system parameters params and the master-key for the underlying identity-based encryption scheme. F also runs $\mathsf{PKE.Gen}$ to generate a public/private key (EK,DK) for the underlying public key encryption scheme. Since F knows DK and master-key, F can easily answer B's private key extraction, SEM-Decrypt and User-Decrypt queries.

Upon receiving two equal-length plaintexts M_0, M_1 and an identity ID_A from B, F can create a challenge ciphertext for B as follows: F computes $C_1 = \mathsf{IBE.Enc}^{\mathsf{VK}}_{\mathsf{params}}(\mathsf{ID}_A, s_1)$ for some random string s_1 and $C_2 = \mathsf{PKE.Enc}^{\mathsf{VK}}_{\mathsf{EK}}(s_1 \oplus M_b)$ by choosing random $b \in \{0, 1\}$. F queries its one-time signature oracle to get a signature $\sigma = \mathsf{Sig}_{\mathsf{SK}}(C_1, C_2)$. Finally F returns $C = (C_1, C_2, \mathsf{VK}, \sigma)$ as a challenge ciphertext for B.

F continues to answer B's private key extraction, SEM-Decrypt and User-Decrypt queries using DK and master-key up to the point when B queries $C' = (C'_1, C'_2, \mathsf{VK}, \sigma')$, where $(C'_1, C'_2) \neq (C_1, C_2)$, to one of the SEM-Decrypt and User-Decrypt oracles. (Notice that happening this event means that σ' is a *new* valid signature on (C'_1, C'_2) with respect to VK). Finally F outputs σ' as a forgery.

Remarks on the cases $(*)$ and $(**)$. Notice that the probability that $(*)$ occurs is bounded by the success probability of the adversary F, i.e. the probability of successfully forging the one-time signature, and hence is small: In the case $(*)$, if $\mathsf{VK}' = \mathsf{VK}$ and $C'_2 = C_2$, C'_1 must be different from C_1 (otherwise, the queried ciphertext will be the same as the challenge ciphertext). Hence σ' becomes a new valid signature means on the new message (C_1, C'_2), which means breaking the security of the one-time signature. Similarly, the probability that $(**)$ occurs is small.

Summing up, B', B'' and F could be constructed using the adversary B and this proves the theorem. □

However, we do not know yet whether the modified CBN scheme remains secure under the chosen ciphertext attack with public key replacement. In general,

proving the security of the certificateless encryption (considering public key replacement attack) in the standard mode is a non-trivial task as discussed in [6]. This would be the case for the modified CBN scheme due to its generic nature of the underlying schemes that do not depend on the random oracles.

References

1. S.S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. In: *Proc. of ASIACRYPT '03*, LNCS 2894, pp. 452-473. Springer-Verlag, 2003.
2. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3): 586C615, 2003.
3. S.S.M. Chow, C. Boyd, and J.M.G. Nieto. Security-Mediated Certificateless Cryptography. In: *Proc. of PKC '06*, LNCS 3958, pp. 508-524. 2006. Springer-Verlag, 2006.
4. A. Dent and C. Kudla. On Proofs of Security for Certificateless Cryptosystems. IACR ePrint archive, report 348, 2005. Available at `http://eprint.iacr.org/2005/348`.
5. X. Ding and G. Tsudik. Simple Identity-Based Cryptography with Mediated RSA. In: *Proc. of CT-RSA '03*, LNCS 2612, pp. 193-210. Springer-Verlag, 2003.
6. Y. Dodis and J. Katz. Chosen-Ciphertext Security of Multiple Encryption. In: *Proc. of TCC '05*, LNCS 3378, pp. 188-209. Springer-Verlag, 2005.
7. V. Shoup. ISO 18033-2: An Emerging Standard for Public-Key Encryption, December 6, 2004. Available at `http://shoup.net/iso/`.