Implicit and Explicit Certificates-Based Encryption Scheme

Tomasz Hyla, Witold Maćków, Jerzy Pejaś

Implicit and Explicit Certificates-based Encryption Scheme

Tomasz Hyla[1], Witold Maćków[1], Jerzy Pejaś[1],

[1] West Pomeranian University of Technology, Szczecin
Faculty of Computer Science and Information Technology, Poland
{thyla, wmackow, jpejas}@zut.edu.pl

Abstract. Certificate-based encryption (CBE) combines traditional public-key encryption and certificateless encryption. However, it does suffer to the Denial of Decryption (DoD) attack called by Liu and Au. To capture this attack, they introduced a new paradigm called self-generated-certificate public key cryptography. In this paper we show that the problem of DoD attack can be solved with a new implicit and explicit certificates-based public key cryptography paradigm. More importantly, we propose a concrete implicit and explicit certificate-based encryption (IE-CBE) scheme that defends against DoD attack. This new scheme is enhanced version of CBE scheme and preserves all its advantages, i.e., every user is given by the trusted authority an implicit certificate as a part of a private key and generates his own secret key and corresponding public key. In addition, in the IE-CBE scheme trusted authority has to generate an explicit certificate for a user with some identity and a public key. We prove that our scheme is IND-CCA2$^-$ and DoD-Free secure in the random oracle model as hard is to solve *p*-BDHI and *k*-CCA problems.

Keywords: pairing based cryptography, implicit certificate, explicit certificate, encryption scheme, random oracle model

1 Introduction

In Asiacrypt 2003, S. Al-Riyami and K. Paterson [1] introduced a new cryptographic paradigm called Certificateless Encryption (CLE). The CLE scheme is an intermediate step between Identity-Based Encryption (IBE) schemes and Public Key Encryption (PKE) schemes based on traditional public key cryptography (see [1, 2, 3, 4, 5]). In the CLE schemes, a Trusted Authority (TA) is involved in issuing user partial private keys computed from TA's master secret. The user also independently generates an additional secret value and calculates both the private and corresponding public keys. Even if a TA knows the user's partial private key, impersonation is impossible.

In PKE approach, the message sender needs to retrieve the authenticated parameters from the Certificate Authority (CA), the user's public key, and the certificate signed by the CA. In CLE, the message sender also needs to retrieve the authenticated parameters

Corresponding author: Tomasz Hyla, e-mail address: thyla@zut.edu.pl, full postal address: West Pomeranian University of Technology in Szczecin, Faculty of Computer Science and Information Technology, ul. Zolnierska 52, 71-210 Szczecin, Poland, phone: (+48) 449 56 62

from the TA and the user's public key, but not any certificate [5]. On the one side this last CLE feature allows to eliminate the third-party queries for the certificate, but on other side the lack of a certificate does not allow to identify the proper public key. As a result, the sender may choose a wrong public key, or even use another one which is never owned by the intended recipient.

Liu J. K., *et al.* [6] were the first to notice that a CLE schemes did not prevent a sender from encrypting a message using an incorrect public key and termed this feature as a Denial of Decryption (DoD) attack, since this possibly denies the recipient's opportunity to get a correct decryption result. In DoD attack the adversary cannot gain any secret information, but any authorised user is also not able to decrypt this information and get the normal service. The adversary can succeed to launch this attack since there is no checking whether the public key is associated with the proper person or not.

Unfortunately, the certificate-based encryption (CBE) schemes introduced by Gentry in 2003 [7] also do not resist the DoD attacks. Each user in the CBE scheme achieves a certificate from a TA. However, this certificate is a part of a private key, so that certificate is implicit and should be kept in secrecy. The secrecy of the implicit certificate means that the encrypting subject implicitly assumes existence of a certificate related to the recipient of an encrypted message. However, is this assumption correct in any case? No, because CBE scheme did not prevent a sender from encrypting a message using a public key which does not correspond to the recipient's identity *ID* for which the message is intended.

In the literature a few solution of the DoD problem exists (e.g., [6], [8, 9]). One of the firsts belongs to Liu J. K., *et al.* [6], which propose the idea of self-generated-certificate public key encryption (SGC-PKE) to address this problem. Same as CLE and CBE schemes, the TA in SGC-PKE scheme is trusted to only issue a partial private key after user's authentication. The underlying idea for the construction of SGC-PKE scheme consists of asking the recipient to use one partial private key to certify (to sign) the public key and only then to share a correct copy of the public key, while the second one to decrypt the ciphertext received from the sender. As a result, there are two full private keys, one for CLE and the other for certificateless signature (CLS).

It is noteworthy that other SGC-PKE scheme given by Lai, J. and Kou, K. [8] essentially instantiates above generic construction of Liu J. K., *et al.* [6]. In Lai-Kou's scheme the receiver and the TA must undertake a protocol before the receiver can sign its identity and public keys using private key. This last operation means that the receiver creates a digital self-generated certificate which binds the receiver's encryption key to its identity.

Dent, A.W. [9] describes the certificate-chain certificateless encryption scheme that combines a SGC-PKE approach with a traditional public-key encryption scheme PKE. This scheme demonstrates that a PKI-based public-key encryption scheme with a certificate generated by the CA (Certificate Authority) can be used to instantiate a BSS certificateless encryption scheme [10] with receiver self-generated certificate.

The above-mentioned schemes have one fundamental advantage: they allow for the authentication of the receiver's identity and its public key. Therefore, if a sender wishes

to encrypt a message, then the sender first checks whether the certificate correctly authenticates the encryption key for the receiver's identity. This procedure resembles a traditional public key encryption systems based on Public Key Infrastructure (PKI): the message senders still need to retrieve and verify the self-generated certificates. The only difference from the PKE approach is another certification process including the issuance and management of certificates. In SGC-PKE, the certificate is self-generated and managed by the receiver, while in the PKE, it is generated and managed by the CA. This last features cannot be rather treated as an advantage of SGC-PKE compared with PKE, because such SGC-PKE schemes do not allow building global encryption systems.

1.1 Our contribution

In this paper, we introduce a new paradigm called Implicit and Explicit Certificates-Based Public Key Cryptography (IEC-PKC) to defend against the DoD attack and propose a concrete encryption scheme (IE-CBE). This scheme preserves all advantages of Certificate-Based Public Key Cryptography (CB-PKC), i.e., every user is given, by the TA, an implicit certificate as a part of a private key and generates his own secret key as well as corresponding public key. In addition, in the IE-CBE scheme the TA has to generate an explicit certificate for a user with some identity and a public key. The purpose of this explicit certificate is similar both to the self-generated certificate in SGC-PKE and the one in traditional PKC. However, the main difference is that in SGC-PKE schemes two secret keys are randomly generated, while in IE-SK-CBE only one. The implicit and explicit certificates should be related with each other in such a way that no one, even the entity of those certificates and their issuer (TA authority) should not be able to recreate an implicit certificate using the explicit certificate.

1.2 Paper Organisation

The remainder of this paper is organized as follows. In Section 2 we present a formal definition for the Implicit and Explicit Certificate-Based Encryption (IE-CBE) scheme and its security model. In Section 3, we present IE-CBE Scheme based on Sakai-Kasahara encryption scheme [3, 4] derived from CBE [12] schemes and provide a formal security proof of it in Section 4. The paper ends with conclusions.

2 An Implicit and Explicit Certificate-based Encryption Scheme

2.1 Generic IE-CBE Encryption Scheme

In this section, we present a formal definition for the IE-CBE scheme. The three main entities involved in an IE-CBE scheme are a sender, a receiver and a trusted authority chosen by the sender. The scheme uses bilinear pairings [15] and using notions similar to those presented by S. Al-Riyami, et al. [1].

Definition 1. An implicit and explicit certificate-based encryption scheme (IE-CBE) is the 7-tuple of algorithms which are defined in Table 1.

Table 1. Generic IE-CBE encryption scheme definition

Algorithm	Input	Output	Run by
Setup	1^k	*params*, P_0, s_{TA}	*TA*; s_{TA} is secret
Create-User	*params*, P_0, ID_R	$s_{2_{ID_R}}$, Pk_{ID_R}, $\overline{CI}_{ID_R}$	User; $s_{2_{ID_R}}$ is secret
Extract-Partial-Private-Key	*params*, P_0, s_{TA}, $\overline{CI}_{ID_R}$	Sk'_{ID_R}, CI_{ID_R}	TA for each user
Certificate-Generate	*params*, P_0, s_{TA}, CI_{ID_R}	$Cert_{ID_R}$	TA for each user.
Set-Private-Key	*params*, $\overline{CI}_{ID_R}$, Sk'_{ID_R}, $s_{2_{ID_R}}$	Sk_{ID_R}	User; Sk_{ID_R} is secret
Encrypt	*params*, CI_{ID_R}, $Cert_{ID_R}$, m	$C^m_{ID_R}$ or $\perp$	User that encrypts m
Decrypt	*params*, CI_{ID_R}, Sk_{ID_R}, $C^m_{ID_R}$	*m*	User that decrypts m

Notations:

1^k	security parameter	$C^m_{ID_R}$	a ciphertext
$Cert_{ID_R}$	an user's certificate	$\overline{CI}_{ID_R}$	user's partial certificate information (includes Pk_{ID_R}, P_0, ID_{TA}, ID_R,)
CI_{ID_R}	a full certificate information (includes Pk_{ID_R}, P_0, ID_{TA}, ID_R, τ) of user with ID_R		
ID_R	receiver identity	ID_{TA}	trusted authority identity
m	a plaintext $m \in (0,1)^n$	*n*	number of bits
params	system parameters	P_0	master public keys
$s_{2_{ID_R}}$	a secret key value	s_{TA}	master private key
$Sk_{ID_R} = (s_{2_{ID_R}}, \overline{Sk}_{ID_R})$	the full user's private key	Sk'_{ID_S}	a blinded partial private key
$\overline{Sk}_{ID_R}$	an unblinded value of Sk'_{ID_R}	τ	time period for which the information in CI_{ID_R} is valid
$\perp$	not valid symbol		

It is required that algorithms from Table 1 must satisfy the standard consistency constraint, i.e., for all $m \in \{0,1\}^n$, **Decrypt**($C^m_{ID_R}$, *params*, CI_{ID_R}, Sk_{ID_R}) = *m*, where **Encrypt**(*m*, *params*, CI_{ID_R}, $Cert_{ID_R}$) → $C^m_{ID_R}$, **Certificate-Generate** (s_{TA}, P_0, *params*,

$\overline{CI}_{ID_R}$) → ($Cert_{ID_R}$, CI_{ID_R}) and (Pk_{ID_R} , Sk_{ID_R} , $Cert_{ID_R}$) is a valid public/private certified key pair.

2.2 Security Model

The security model should appropriately describe the real-world security needs to demonstrate that the scheme resists all practical attacks, but the model should not be so powerful that it would require to use overly complex and inefficient schemes in order to meet the security notions [9]. We require the IND-CCA2 [6], [8, 9], [12] notion of security for the encryption scheme. This captures the notion that no attacker can determine any information about a message from a ciphertext even, if they can obtain the decryptions of any other adaptively prepared ciphertext.

The security model of IE-CBE scheme is modified version of the models proposed by S. Al-Riyami and K. Paterson [1], A. Dent [9], Lai, J., Kou, K. [8] and J. K. Liu, et al. [6]. According to these models, there are two types of adversaries. **Type I** adversary is an uncertified user, who is allowed to impersonate an arbitrary victim by changing his public key with other public key of his own choice, that the sender uses to encrypt messages, but does not have access to the TA's master-key. It can also obtain partial and full secret keys of arbitrary identities, and the certificates of all users except the certificate for the forged certificate information of the victim. **Type II** adversary is a malicious TA that is equipped with master-key and can compute the master public key value maliciously (see [9], [11]), but is not allowed to replace public keys. The main goal of Type II adversary is to impersonate a victim with a given public key and without access to the corresponding secret private key chosen by the victim.

Typically, it is expected that the decryption oracle should be able to correctly respond to decryption queries made on identities whose public keys have been replaced by the Type I adversary and for which oracle does not know the corresponding private keys. However, such security model is to strong and does not reflect an attacker's real-life capabilities [9, 10]. In our IE-CIBE scheme we assumed that the challenger is not forced to attempt to decrypt ciphertext for which the public key has been replaced, if the corresponding secret key is not known. It is known as **Type I**$^{-}$ adversary [6].

A security model is typically presented as a game played between an arbitrary (probabilistic polynomial-time, PPT) adversary *A* representing given an encryption scheme and a challenger (who represents a new algorithm *B* which uses *A* as a subroutine and supplies the answers to *A*'s oracle queries). The challenger keeps a list of users in the system and all TA-issued certificates, their real public/private key pairs, and the public key value that the sender associates with each user. The adversary interacts with the challenger via a series of oracles which force the challenger to perform certain operations and model the different ways that the adversary can interact with the system.

Definition 2 (IND-CCA2^{-} security, compare [6], [8, 9], [12]). The IE-CBE encryption scheme is said to be IND-CCA2^{-} secure if no PPT adversary A of Type I^{-} or Type II has a non-negligible advantage in the following game played against the challenger:

Setup. The challenger *C* takes a security parameter 1^k and runs the *Setup* (1^k) algorithm. It gives *A* the resulting system parameters *params* and a random TA public key P_0. If *A* is of Type I⁻, the challenger keeps the master secret key s_{TA} to itself. Otherwise, it gives s_{TA} to *A* and additionally, a random public key Pk_* of some user.

Phase 1. In this phase, the adversary *A* can adaptively issue queries to the following oracles:

- *CreateUser-Query*(ID_R). On input an user's identity ID_R, the challenger first generates his public key $Pk_{ID_R} = (X_{ID_R}, Y_{ID_R}, Z_{ID_R}, R_{ID_R})$. If a user with identity index (ID_R, Pk_{ID_R}) is already created, then challenger responds with the public key Pk_{ID_R} associated with the identity ID_R. Otherwise, the challenger calculates the full private key Sk_{ID_R} and composes the certificate information CI_{ID_R}. Finally, the challenger calculates the explicit certificate $Cert_{ID_R}$ and outputs Pk_{ID_R} and CI_{ID_R} to *A*. The tuple ($\overline{ID}_R$, Sk_{ID_S} , Pk_{ID_R} , $Cert_{ID_S}$, CI_{ID_R}) is added to the $Users_{list}$ list and the user with identity $\overline{ID}_R = (ID_R, Pk_{ID_R})$ is said to be created. We assume that other oracles defined below only respond to an identity which has been created.
- *Cert-Generate-Query*($\overline{ID}_R$, CI_{ID_R}). (*This oracle is applicable to Type I⁻ adversary*.) When adversary *A* queries a user with identity $\overline{ID}_R$ and the certificate information CI_{ID_R}, the challenger *C* returns the certificate $Cert_{ID_R}$ to *A*. If the identity $\overline{ID}_R \notin User_{list}$, the symbol $\perp$ is returned.
- *Extract-Partial-Private-Key-Query*($\overline{ID}_R$, CI_{ID_R}). (*This oracle is applicable to Type I⁻ adversary*.). On input of an identity index $\overline{ID}_R$ supplied by an adversary, challenger *C* returns a partial key $\overline{Sk}_{ID_R}$ whenever the user with identity index $\overline{ID}_{ID_R}$ has been created. Otherwise, a symbol $\perp$ is returned.
- *Private-Key-Extract-Query*($\overline{ID}_R$). (*This oracle is applicable to Type I⁻ adversary*.). On receiving a query for an identity index $\overline{ID}_R$, challenger *C* responds with the private key Sk_{ID_R}. If the identity $\overline{ID}_R$ has no associated private key or the user's public key has been replaced, the challenger *C* returns a symbol $\perp$.
- *Public-Key-Replace-Query*($\overline{ID}_R$, Pk'_{ID_R}). (*This oracle is applicable to Type I⁻ adversary*.) This oracle takes an identity $\overline{ID}_R$ and allows adversary *A* to replace a public key Pk_{ID_R} with a new value Pk'_{ID_R} chosen by him.

– *Certificate-Replace-Query*($\overline{ID}_R$, $Cert'_{ID_R}$). (*This oracle is applicable to Type I⁻ adversary*.) This oracle acts as *Public-Key-Replace-Query*, but this time the adversary *A* is able replace a previous certificate $Cert_{ID_R}$ with a new value $Cert'_{ID_R}$ chosen by him.
– *Decryption-Oracle*($\overline{ID}_R$, CI_{ID_R} , $C^m_{ID_R}$). This oracle takes as input an identity $\overline{ID}_R$, the user's certificate information CI_{ID_R} and the ciphertext $C^m_{ID_R}$ for some message *m* and returns the decrypted plaintext. If the user's public key has been replaced, it requires an additional input of the corresponding secret key for the decryption. If this secret key is unknown to oracle, then a symbol $\perp$ is returned (only in the case of Type I⁻ adversary).

Challenge. When the adversary *A* decides that Phase 1 is over, it outputs and submits two message (m_0, m_1), together with an identity $\overline{ID}_*$ of uncorrupted secret key and the corresponding certificate information CI_{ID_*}. If *A* is of Type II adversary, it is allowed additionally to generate the master public key P'_0 of the TA different then its correctly generated static master public key P_0 and some state information [9], [11]. All information prepared by the adversary *A* are sent to the challenger *C*. The challenger picks a random bit $\beta \in \{0,1\}$ and computes $C^{m_\beta}_{ID_*}$, the encryption of the message m_β under the current public key Pk_{ID_*} for ID_*. If this ciphertext is correct, the challenger *C* sends $C^{m_\beta}_{ID_*}$ as the challenge to the adversary *A*. Otherwise the challenger *C* outputs $\perp$ and *A* loses the game.

Phase 2. In this phase, the adversary *A* may adaptively query the same oracles as in the Phase 1. In any moment it terminates game and outputs a guess $\beta' \in \{0,1\}$.

Guess. The adversary *A* wins this security game if $\beta \neq \beta'$ and the following restrictions are fulfilled:

– in Phase 2, the *A* cannot use *Decryption-Oracle* ($\overline{ID}_*$, CI_{ID_*} , $C^{m_\beta}_{ID_*}$) for the tuple ($\overline{ID}_*$, CI_{ID_*}) under which the message m_β was encrypted;
– in Phase 1, the adversary *A* of Type I⁻ cannot submit $\overline{ID}_*$ and/or CI_{ID_*} to *Cert-Generate-Query*, *Extract-Partial-Private-Key-Query* and *Private-Key-Extract-Query*;
– if *A* is Type II, the identity $\overline{ID}_*$ has not been submitted to *Private-Key-Extract-Query*.

The adversary's advantage is defined to be $Adv^{IND-CCA^-}_{IE-CIBE}(A) = \left|\Pr[\beta = \beta'] - 1/2\right|$ and the scheme IC-CIBE is said to be secure against the adversary *A* of Type I⁻ and II if this advantage is negligible.

For security, in addition to IND-CCA2^{-}, we require the IE-CIBE encryption scheme to be DoD-Free. The formal security model for DoD attacks is defined as a game played between the challenger and a PPT adversary (DoD adversary), which has the same power as the adversary *A* of a Type I^{-}.

Definition 3 (DoD-Free Security, see [6], [9]). We say that IE-CIBE encryption scheme is DoD-Free secure if no PPT adversary *A* has a non-negligible advantage in the following game played against the challenger:

Setup. The challenger *C* takes a security parameter 1^k and runs the *Setup* (1^k) algorithm. It gives *A* the resulting system parameters *params* and a random public key P_0 of the TA. The challenger keeps the master secret key s_{TA} to itself.

Queries. In this phase, the adversary *A* can adaptively issue queries to the same oracles which are given in Phase 1 to the adversary *A* of a Type I^{-} (see Definition 2).

Challenge. When the adversary *A* decides that Phase 1 is over, it outputs message m_* together with an identity $\overline{ID}_*$ and the corresponding certificate information CI_{ID_*}. All information are sent to the challenger C, which computes $C_{ID_*}^{m_*}$, the encryption of the message m_* under the current public key Pk_{ID_*} for ID_*. If the output of the encryption is $\perp$, then *A* immediately losses the game. Otherwise, it outputs $C_{ID_*}^{m_*}$.

Constrains. The adversary *A* wins the game if the following requirements are fulfilled:

- the ciphertext $C_{ID_*}^{m_*}$ computed in Challenge phase is not $\perp$;
- the output of the *Decrypt*($C_{ID_*}^{m_*}$, *params*, CI_{ID_*}, Sk_{ID_*}) is not equal m_* for the tuple ($\overline{ID}_*$, CI_{ID_*}) under which the message m_* was encrypted;
- the adversary has not been submitted $\overline{ID}_*$ and/or CI_{ID_*} to *Cert-Generate-Query*, *Extract-Partial-Private-Key-Query* and *Private-Key-Extract-Query*.

The DoD adversary's advantage is defined to be $Adv_{IE-CIBE}^{DoD-Free}(A) = \Pr[A\ wins]$.

The IND-CCA2^{-} security model of IE-CIBE is a little different from the definition of the chosen ciphertext security model given in [8], [10], [12]. First, it contains two new queries on an explicit certificate extraction and its replacement, i.e., *Cert-Generate-Query* and *Certificate-Replace-Query* (the Type I^{-} adversary only), respectively. Second, the Type II adversary is challenged on a random partial public key of a user and the TA public key of its choice. Note that the Type II adversary is not required to show its knowledge of the matching private keys corresponding to these public keys. When using the IND-CCA2^{-} and DoD-Free games we can define the security for the IE-CIBE scheme.

Definition 4. The IE-CBE encryption scheme is said to be secure if it is both IND-CCA2^{-} secure and DoD-Free secure.

3 IE-CBE Scheme Based on Sakai-Kasahara encryption scheme

The IE-CBE scheme is constructed on the Sakai-Kasahara identity-based encryption scheme [3, 4] and is similar to the certificate-based encryption (CBE) scheme given by Y. Lu and J. Li [12].

3.1 Full Implicit and Explicit Certificate-Based encryption scheme (IE-CBE)

The proposed IE-SK-CBE scheme consists of eight algorithms: **Setup**, **Create-User**, **Extract-Partial-Private-Key**, **Certificate-Generate**, **Set-Public-Key**, **Set-Private-Key**, **Encrypt** and **Decrypt**:

Setup. For given security parameters 1^k and two cyclic groups $(G_1, +)$ and $(G_2, \times)$ of the same prime order $q>2^k$, a trusted authority (TA):

(a) generates P being a generator of G_1 and chooses the bilinear admissible pairing given as $\hat{e}: G_1 \times G_1 \to G_2$ (e.g., [1, 12, 15]);

(b) picks a random main key $s_{TA} \in_R Z_q^*$;

(c) calculates the public key $P_0 = \left(\overline{P}_0, \widetilde{P}_0\right)$, where $\overline{P}_0 = s_{TA} P$ and $\widetilde{P} = s_{TA} s_{TA} P$;

(d) selects five secure hash functions: $H_1 : \{0,1\}^* \to Z_q^*$, $H_2 : \{0, 1\}^* \times G_1^3 \to Z_q^*$, where notation G_1^3 is the Cartesian product of groups G_1 defined as $G_1^3 = G_1 \times G_1 \times G_1$, $H_3 : G_1 \times G_2 \times G_1 \to \{0,1\}^n$, and $H_4 : \{0, 1\}^n \to \{0, 1\}^n$ for some integer $n>0$, where n is plaintext message $m \in \{0,1\}^n$ length in bits.

The message space is $M=\{0, 1\}^n$, while the ciphertext space is $C= G_1^* \times \{0,1\}^n$.

Create-User. Decrypting entity R generates a key material that contains R private key and a partial public key.

(a) R chooses two secret random values $s_{1_{ID_R}}, s_{2_{ID_R}} \in_R Z_q^*$;

(b) R calculates a public key $Pk_{ID_R} = \left(X_{ID_R}, Y_{ID_R}, Z_{ID_R}\right)$, where $X_{ID_R} = s_{2_{ID_R}} P$, $Y_{ID_R} = s_{2_{ID_R}} \overline{P}_0$ and $Z_{ID_R} = s_{2_{ID_R}} \widetilde{P}_0$;

(c) R calculates parameters $userParams = \{UP_1, UP_2, X_{ID_R}, ID_R\}$, where $UP_1 = s_{1_{ID_R}} \left(s_{2_{ID_R}} + \overline{q}_{ID_R}\right)^{-1} X_{ID_R}$ for $\overline{q}_{ID_R} = H_1\left(\overline{CI}_{ID_R}\right)$ and $UP_2 = s_{1_{ID_R}} P$;

Remark. The UP_1 value, as in the traditional PKC, proves by R to the TA the possession of secret key $s_{2_{ID_R}}$ corresponding to public key Pk_{ID_R} (see **Extract-Partial-Private-Key** algorithm).

(d) R composes the well-formed (i.e., using syntax rules specified by some data specification language like ASN.1 or XML) partial certificate information $\overline{CI}_{ID_R}$,

filling it with desired values including the *userParams*, the public keys (P_0 , Pk_{ID_R}) and identities for both the subject *R* and the TA;

(e) *R* sends $\overline{CT}_{ID_R}$ to TA.

Extract-Partial-Private-Key. *TA* authority calculates a blinded partial private key of an entity *R*:

(a) *TA* verifies and registers ID_R; if entity *R* is already registered, then TA omits registration and goes into key renewal mode;

(b) *TA* based on X_{ID_R} calculates Y_{ID_R} and Z_{ID_R}, then compares them with the content of $\overline{CT}_{ID_R}$, subsequently calculates a $\overline{q}_{ID_R} = H_1\left(\overline{CT}_{ID_R}\right)$ and verifies if equation $\hat{e}\left(UP_1, X_{ID_R} + \overline{q}_{ID_R} P\right) = \hat{e}\left(X_{ID_R}, UP_2\right)$ is true; if it is false, the algorithm is ended; if it is true, TA has a proof, that identity ID_R is related to secret key $s_{2_{ID_R}}$ and to X_{ID_R}, Y_{ID_R}, Z_{ID_R};

(c) *TA* composes the full user certificate information CI_{ID_R}, including the public keys Pk_{ID_R} and P_0, identifiers of the user *R* and the *TA*, and the time period τ for which this information CI_{ID_R} is valid;

(d) TA calculates a blinded partial private key $Sk'_{ID_R} = \left(s_{TA} + q_{ID_R}\right)^{-1} s_{TA} UP_1$, where $q_{ID_R} = H_1\left(CT_{ID_R}\right)$ and together with CI_{ID_R} sends it to entity *R*.

Certificate-Generate. TA authority, using parameters received from *R* and values calculated during execution **Extract-Partial-Private-Key** algorithm, generates an explicit certificate $Cert_{ID_R}$ of an entity *R*.

(a) TA generates a certificate for an entity *R*, which binds identity with public key components:

$$Cert_{ID_R} = \frac{1}{s_{TA} + q_{ID_R}} P \tag{1}$$

(b) TA sends $Cert_{ID_R}$ to an entity R.

Set-Private-Key. An entity R calculates a full private key Sk_{ID_R}.

(a) *R* verifies correctness of Sk'_{ID_R}:

$$\hat{e}\left(Sk'_{ID_R}, Y_{ID_R} + q_{ID_R} X_{ID_R} + \overline{q}_{ID_R}\left(\overline{P}_0 + q_{ID_R} P\right)\right) = \hat{e}\left(P, s_{1_{ID_S}} s_{2_{ID_S}} \overline{P}_0\right) \tag{2}$$

(b) *R* calculates a second part of the private key:

$$\overline{Sk}_{ID_R} = s_{1_{ID_R}}^{-1}\left(s_{2_{ID_R}} + \overline{q}_{ID_R}\right) S'_k = \frac{1}{s_{TA} + q_{ID_R}} Y_{ID_R} \tag{3}$$

(c) *R* formulates a private key for entity *R* in the form: $Sk_{ID_R} = \left(s_{2_{ID_R}}, \overline{Sk}_{ID_R}\right)$.

Encrypt. To encrypt the message $m \in \{0,1\}^n$, the sender *S*:

(a) calculates $q_{ID_R} = H_1\left(CT_{ID_R}\right)$, and then verifies the authenticity of the certificate $Cert_{ID_R}$:

$$\hat{e}\left(Cert_{ID_R}, q_{ID_R} Y_{ID_R} + Z_{ID_R}\right) = \hat{e}\left(P, Y_{ID_R}\right) \tag{4}$$

$$\hat{e}\left(X_{ID_R}, \tilde{\overline{P}}_0\right) = \hat{e}\left(Y_{ID_R}, \overline{P}_0\right) = \hat{e}\left(Z_{ID_R}, P\right) \tag{5}$$

Remark. When equations (4) and (5) are true, then components of public key Pk_{ID_R} are authentic, which implies that a public key Pk_{ID_R} belonging to the entity with an identity ID_R is authentic.

(b) if the verification result from previous step is positive, then *S* chooses a random number $v \in \{0,1\}^n$ and calculates:

$$r = H_2\left(v, m, ID_R, Pk_{ID_R}\right) \tag{6}$$

$$U = r\left(\overline{P}_0 + q_{ID_R} P\right) \tag{7}$$

$$k = H_3\left(U, \hat{e}\left(Cert_{ID_R}, r\left(Z_{ID_R} + q_{ID_R} Y_{ID_R}\right)\right), r\left(Y_{ID_R} + q_{ID_R} X_{ID_R}\right)\right) \tag{8}$$

$$V = v \oplus k\,,\ W = m \oplus H_4\left(v\right) \tag{9}$$

(c) *S* creates the ciphertext $C = \left(U, V, W\right)$ and sends it to a recipient *R*.

Decrypt. A decryption entity R reconstruct message *m* using ciphertext *C*.

(a) *R* calculates:

$$k' = H_3\left(U, \hat{e}\left(\overline{Sk}_{ID_R}, U\right), s_{2_{ID_R}} U\right) \tag{10}$$

$$v' = V \oplus k' \tag{11}$$

$$m' = W \oplus H_4\left(v'\right) \tag{12}$$

$$r' = H_2\left(v', m', ID_R, Pk_{ID_R}\right) \tag{13}$$

(b) if $U \neq r'\left(H_1\left(CT_{ID_R}\right)P + \overline{P}_0\right)$, then decryption process is incorrect, otherwise *m'* is a correct plain text corresponding to the ciphertext $C = \left(U, V, W\right)$.

3.2 IE-CBE scheme correctness

Assume that the ciphertext $C=(U,V,W)$, the partial private key Sk_{ID_R} and the explicit certificate $Cert_{ID_S}$ were generated using the **Encrypt**, **Certificate-Generate** and **Extract-Partial-Private-Key** algorithms, respectively. Hence, combining equation (8) with equations (1), (3) and (10) shows the following:

$$\begin{aligned} k &= H_3\left(U, \hat{e}\left(Cert_{ID_R}, r\left(Z_{ID_R} + q_{ID_R} Y_{ID_R}\right)\right), r\left(Y_{ID_R} + q_{ID_R} X_{ID_R}\right)\right) = \\ &H_3\left(U, \hat{e}\left(\frac{1}{s_{TA} + q_{ID_R}} P, s_{2_{ID_R}} s_{TA} r\left(\overline{P}_0 + q_{ID_R} P\right)\right), s_{2_{ID_R}} r\left(\overline{P}_0 + q_{ID_R} P\right)\right) = \\ &H_3\left(U, \hat{e}\left(\frac{1}{s_{TA} + q_{ID_R}} s_{2_{ID_R}} s_{TA} P, r\left(\overline{P}_0 + q_{ID_R} P\right)\right), s_{2_{ID_R}} U\right) = \\ &H_3\left(U, \hat{e}\left(\overline{Sk}_{ID_R}, U\right), s_{2_{ID_R}} U\right) = k' \end{aligned} \tag{14}$$

Furthermore, it is now easy to prove the correctness of equations (4):

$$\begin{aligned} &\hat{e}\left(Cert_{ID_R}, q_{ID_R} Y_{ID_R} + Z_{ID_R}\right) = \\ &\hat{e}\left(\frac{1}{s_{TA} + q_{ID_S}} P, \left(s_{TA} + q_{ID_R}\right) Y_{ID_R}\right) = \hat{e}\left(P, Y_{ID_R}\right) \end{aligned} \tag{15}$$

3.3 IE-CBE scheme modification

Any certificate-based encryption (CBE) scheme contains an implicit certificate that is a part of a private key. Hence, it seems to be naturally to modify any particular encryption scheme based on both explicit and implicit certificate and produce a CBE scheme that can be proven secure.

Assume that this is possible in our case and we may remove the **Certificate-Generate** algorithm form IE-CBE scheme. The resulting scheme is a new scheme based on an implicit certificate (let's name it I-CBE, Implicit Certificate-Based Encryption scheme). Introduced change requires to remove certificate verification (Eq. 4) in the algorithm **Encrypt** and modify equation (8), which will be as follows (compare with Eq. 15):

$$k = H_3\left(U, \hat{e}\left(P, Y_{ID_R}\right), r\left(Y_{ID_R} + q_{ID_R} X_{ID_R}\right)\right) \tag{16}$$

Remark. It is easy to notice that a certificate $Cert_{ID_R}$ in relation with the scheme I-CBE plays in the IE-CBE scheme a similar role to self-generated certificate in relation with the underlying CL-PKE scheme in SGC-PKE scheme ([6], [8, 9]).

4 IE-CBE scheme security

In the IE-CBE construction, the implicit and explicit certificates are based on a short signature scheme given in [12, 16] that security depends on a *k*-CAA hard problem (see Definition 1). It means that if adversary is not able to counterfeit an explicit certificate, then it is not possible to execute a DoD attack and IE-CBE scheme is secure as hard is to solve *k*-CCA problem. Because IE-CBE scheme depends on the underlying I-CBE scheme complemented with an algorithm **Certificate-Generate**, hence it is natural to divide its security proof into two phases: in the first it must be shown that I-CBE scheme is IND-CCA2$^-$ secure and in the second that IE-CBE scheme is DoD free.

A similar approach was used for a security model of SGC-PKE scheme [6], [8, 9], where they first examine the security of CL-PKE from which the SGC-PKE developed, and then consider the DoD-Free security. In our case, we construct the IE-CBE encryption scheme from an implicit certificate-based encryption (I-CBE) scheme and an explicit certificate built on a short signature defined in [16]. The security of resulting IE-CBE scheme needs to show that the requirements of Definition 4 are met and thus following Theorem should hold.

Theorem. The IE-CBE scheme is IND-CCA2$^-$ and DoD-Free secure in the random oracle model.

To prove the above theorem, we first prove the IND-CCA2 security of the IE-CBE scheme (Lemma 1 and 2) and then show that IE-CBE scheme is DoD-Free (Lemma 3).

Lemma 1. The IE-CBE scheme is IND-CCA2$^-$ secure if IND-CCA2$^-$ secure is the underlying I-CBE scheme.

Proof. The definition of IE-CBE given in Section 3.1 is the same as the definition of I-CBE from Section 3.3, except for **Certificate-Generate** algorithm which is used to generate the explicit certificates. This certificates have no influence on the semantic security of I-CBE scheme (see equation (16)), but provide the DoD-Free feature of IE-CBE scheme only (compare Lemma 2). Hence, it is clear that IND-CCA2$^-$ security of I-CBE scheme implies IND-CCA2$^-$ security of IE-CBE scheme.

Lemma 2. In the random oracle model, the I-CBE scheme is IND-CCA2$^-$ secure under the *p*-BDHI assumption (*p*-BDHI problem, Boneh D., Boyen X. [14]).

The proof of Lemma 2 is similar to the proof of [13] and it is run on the basis of the IND-CCA2$^-$ game (see Definition 2), in which the oracle **Cert-Generate-Query** is not accessible and no longer needed, as challenger *C* cannot now generate certificates and the adversary cannot use them in any operation. Due to its length the proof is not included here.

Besides the IND-CCA2$^-$ security property, we require additionally IE-CBE scheme to be DoD-Free secure. The condition that IE-CBE scheme should meet are defined in Lemma given below.

Lemma 3. The IE-CBE scheme is DoD-Fee secure, assuming that the implicit and explicit certificates are existential unforgeable.

Proof. In IE-CBE scheme, the implicit and explicit certificates are short signatures computed using a signature scheme considered in [16]. According to Theorem 3 of [16] this signature scheme is existentially unforgeable under chosen message attack (EUF-CMA) in the random oracle model, assuming that *k*-CCA problem (*k*-CAA problem, Mitsunari S., et al. [13]) is believed to be computationally hard.

We now consider the DoD-Free game implemented with a Type I⁻ adversary *A* (see Definition 3), in which the adversary *A* models an uncertified entity. Suppose that algorithm *F* is a forger that breaks the short signatures. We wish to construct another algorithm *B* that uses *A* with algorithm *F* to solve the *k*-CAA problem. The algorithm *B* receives the *k*-CAA instance (a challenge) with $P, \overline{P}_0, \widetilde{P} \in G_1$, $h_1, h_2, \ldots, h_k \in Z_q^*$ and $\frac{1}{h_1 + s_{TA}} P, \ldots, \frac{1}{h_k + s_{TA}} P$. Its goal is to compute a pair $\left(h^*, \frac{1}{h^* + s_{TA}} P \right)$ for some $h^* \notin \{h_1, \ldots, h_k\}$. As the algorithm *B* has access to the signing-oracle, hence *B* can answer all oracle queries given by *A*, including the queries for the implicit and explicit certificate signing.

When the queries phase of DoD-Free game is over, then the adversary *A* submits message m_* and an identity $\overline{ID}_*$ to the *B*. An adversary *A* wins if following conditions hold (compare [6], [8]):

(a) the certificate $Cert_{ID_*}$ of CI_{ID_*} (with ID_* and Pk_{ID_*}) is valid (see Eq. 4);

(b) $Decrypt(C_{ID_*}^{m_*}, params, CI_{ID_*}, Sk_{ID_*}) \neq m_*$, where $Encrypt(m_*, params, CI_{ID_*}, Cert_{ID_*})$;

(c) the adversary never makes *Cert-Generate-Query*, *Extract-Partial-Private-Key-Query* and *Private-Key-Extract-Query* for $\overline{ID}_*$ and/or CI_{ID_*}.

Due to the correctness of IE-CBE scheme (see Section 3.2), the equality $Decrypt(C_{ID_*}^{m_*}, params, CI_{ID_*}, Sk_{ID_*}) = m_*$ holds always if the condition (a) is satisfied. Because the hash function H_1 is collision-resistant and an adversary cannot find another distinct certificate information CI'_{ID_*} that is in collision with CI_{ID_*}, then the veracity of the condition (a) implies the public key Pk_{ID_*} associated with the certificate $Cert_{ID_*}$ (and the identity $\overline{ID}_*$) has not been replaced. Hence, if the condition (b) holds, the public key Pk_{ID_*} had to be replaced. This means that $Encrypt(m_*, params, CI_{ID_*}, Cert_{ID_*}) \neq \perp$ and thus form the conditions (a) and (c) follows that the certificate $Cert_{ID_*}$ is a successful forgery. Consequently, challenger *B* can compute a group element $\left(s_{TA} + h^*\right)^{-1} P$, where $h^* = q_{ID_*} = H_1\left(CT_{ID_*}\right)$, which is the solution of the *k*-CAA problem.

This ends the proof. □

5 Conclusions

This paper contains an encryption scheme IE-CBE that has been built on a new paradigm called Implicit and Explicit Certificates-Based Public Key Cryptography (IEC-PKC). The idea of this paradigm is similar to Self-Generated-Certificate Public Cryptosystem (SGC-PC) paradigm given in [6, 8] and provides a mechanism for strong authentication of the user's identity, its public key and relationship between these two elements. Moreover, any encryption scheme with this mechanism should be immune to the DoD attack. Our way of achieving this authentication mechanism is different from that used in SGC-PKE: we allow the TA to sign the user's identity and public key, instead of the user signing the self-certificate with TA-issued partial private key.

However, this explicit certificate is closely related to implicit certificate and its role is only technical (compare Eq. 14, 15 and 16). Following this approach we make formally analysis of the IE-CBE scheme security in the random oracle model and prove that the scheme is IND-CCA2 and DoD-Free secure, assuming p-BDHI and k-CCA problems to be computationally hard.

Our future works will focus on applying approach presented in this paper to our group encryption scheme CIBE-GAS [17, 18].

Acknowledgment. This scientific research work is supported by National Centre for Research and Development (NCBiR) of Poland (grant No PBS1/B3/11/2012) in 2012-2015.

References

1. Al-Riyami S., Paterson K.: Certificateless public key cryptography. Advances in Cryptology - AsiaCrypt, LNCS, Vol. 2894, 2003, pp. 452–473, Springer, Verlag (2003)
2. Boneh D., Franklin M.: Identity-Based Encryption from the Weil pairing. Advances in Cryptology – CRYPT01, LNCS, Vol. 2139, Springer, pp. 213–229, (2001)
3. Chen L., Cheng Z.: *Security proof of Sakai-Kasahar's identity-based encryption scheme*. In Proceedings of Cryptography and Coding, LNCS, Vol. 3796, pp. 442-459, Springer-Verlag (2005)
4. Sakai, R., Kasahara, M.: ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054, (2003)
5. Chow, S., S., M.: Certificateless Encryption. In M. Joye and G. Neven (Eds.) Identity-Based Cryptography, pp. 135-155, IOS Press, (2009)
6. Liu, J., K., Au, M., H., Susilo, W.: Self-Generated-Certificate Public Key Cryptography and certificateless signature/encryption scheme in the standard model: extended abstract. In Feng Bao and Steven Miller (Eds.), ASIACCS 2007, pp. 273–283, ACM Press, (2007)
7. Gentry, G.: Certificate-based encryption and the certificate revocation problem. In E. Biham (Ed.): Eurocrypt 2003, LNCS, Vol. 2656, , pp. 272–293, Springer, (2003)

8. Lai, J., Kou, K.: Self-generated-certificate public key encryption without pairing. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 476–489. Springer, Heidelberg (2007)
9. Dent, A., W.: A Brief Introduction to Certificateless Encryption Schemes and Their Infrastructures. Martinelli F. and Preneel B. (Eds.): EuroPKI 2009, LNCS , Vol. 6391, pp. 1–16, Springer, (2010)
10. Baek, J., Safavi-Naini, R., Susilo, W.: *Certificateless public key encryption without pairing*. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, Vol. 3650, pp. 134–148, Springer, Heidelberg, 2005
11. Au, M., H., Chen, J., Liu, J., K., Mu, Y., Wong, D., S., Yang, G.: Malicious KGC Attacks in Certificateless Cryptography. In: ASIACCS (2007), pp. 302-311 (2007)
12. Lu, Y., Li, J.: Constructing Efficient Certificate-based Encryption with Paring, Journal of Computers, Vol. 4, No. 1, January, (2009)
13. Mitsunari, S., Sakai, R., Kasahara, M.: A new traitor tracing, IEICE Transactions, Vol. E85-A, No.2, pp.481-484, (2002)
14. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In Proceedings of Advances in Cryptology – Eurocrypt *2004*, LNCS, Vol. 3027, pp. 223-238, Springer-Verlag, (2004)
15. Lynn, B.: On the implementation of pairing-based cryptosystems. PhD Thesis, Stanford University (2007)
16. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from bilinear pairings and its applications. In Public Key Cryptography – PKC'04, Singapore, LNCS 2947, pp.277–290, (2004)
17. Hyla, T., Pejaś, J.: A practical certificate and identity based encryption scheme and related security architecture. In: Saeed, K., Chaki, N., Cortesi, A., Wierzchon, S. (eds.), CISIM 2013. LNCS, vol. 8104, pp. 178-193. Springer-Verlag, (2013)
18. Hyla, T., Pejaś, J.: Certificate-Based Encryption Scheme with General Access Structure. In: Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S. (eds.) CISIM 2012. LNCS, vol. 7564, pp. 41–55. Springer, Heidelberg (2012)