A Signature Scheme Based on Implicit and Explicit Certificates Against k-Traitors Collusion Attack

Tomasz Hyla, Jerzy Pejaś

A Signature Scheme based on Implicit and Explicit Certificates against *k*-traitors Collusion Attack

Tomasz Hyla[1], Jerzy Pejaś[1]

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

Abstract. In 2002, Mitsunari, Sakai and Kasahara formulated the Collusion Attack Algorithm with k traitors (known as k-CAA problem) and used it to develop the first traitor tracing scheme based on the bilinear pairings. Traitor tracing scheme is needed to discourage legitimate subscribers from sharing their secret keys to construct pirate decoders. In this paper, we propose a first signature scheme (IE-CBS-kCAA) based on k-CAA problem, which belongs to the fourth category of PKC schemes: the public key cryptography schemes based on an implicit and explicit certificates. The security analysis proves that our IE-CBS-kCAA scheme is secure against two game attacks in the random oracle model. The security is closely related to the difficulty of solving the modified k-CAA and discrete logarithm problems.

Keywords: ID-based signature schemes, implicit and explicit certificates-based public key cryptography, bilinear pairing, security analysis, k-CAA problem.

1 Introduction

Collusion Attack Algorithm with *k* traitors (*k*-CAA problem) is one of many hard computational problems used in public key cryptography based on bilinear pairings. Problem of *k* cooperating traitors has a great significance in group signature and encryption schemes. In this schemes, it is assumed that some unauthorized users (adversaries) may obtain some decryption or signature keys from a group of one or more authorized users (traitors). Then the adversaries can decrypt or sign data that they are not entitled to.

In 2002 S. Mitsunari, *et al.* [1] proposed an interesting traitor tracing scheme. The scheme was a first traitor tracing scheme using bilinear mappings. Also, in this paper they formulated *k*-CAA problem. The problem and its variants developed later [2] were useful during construction of many new encryption or signature schemes. The idea of encryption and digital signature schemes based on *k*-CAA problem is very similar to the idea of *k* cooperating traitors in global networking systems. In both cases, it is assumed, that if a group containing less then *k* traitors exists, who disclose their private keys and other secret information, then it is computationally infeasible to recreate or to

generate another private key belonging to an entity from outside the group. This condition in case of a traitor tracing system means that system is *k*-collusion resistant, if tracing succeeds and the adversary has fewer than *k* user keys at his disposal.

Sakai and Kasahara [3] in 2003 proposed one of the first encryption schemes based on *k*-CAA problem. Almost at the same time, in 2004 Zhang *et al.* [4] proposed an efficient ID-based short signature scheme (ZSS), which security relies on *k*-CAA problem. The ZSS scheme plays an important role in many pairing-based cryptographic systems (e.g. P. Barreto *et al*. [5]) and it is a starting point for security proofs in new signature schemes. ZSS scheme is existentially unforgeable under an adaptive chosen message in the random oracle model, although B. C. Hu *et al*. [6] demonstrated that the scheme is vulnerable to an attack called *message-and-key replacement attack*. Another example of ZSS extension was given by Du and Wen [7] in 2007.

The basic problem of ID-based schemes is the key escrow of a user's private key. Certificateless signature scheme (CLS scheme) introduced by S. Al-Riyami and K. Paterson [8] is a practical solution of the key escrow problem. However, most of the ID-based schemes do not satisfy Girault's level-3 security [9], which conventional public key infrastructure (PKI) can achieve. Such drawback has also a certificateless short signature scheme proposed in 2009 by Du and Wen [10], which is an extended version of their ID-based signature scheme [7]. This scheme was the first concrete certificateless signature construction based on *k*-CAA problem with Sakai-Kasahara key construction method.

In 2011 , J. K. Liu *et al.* [16] and J. Li *et al.* [14] presented certificate-based signature schemes and claimed their scheme are proven secure in the random oracle model using *k*-CAA assumption or its variations. In both schemes, Sakai-Kasahara construction of key generation is not used. Unfortunately, Cheng *et al*. [17] showed that J. K. Liu *et al.* scheme is insecure against a Type I adversary under a security model defined in [14], i.e., a Type I adversary can obtain a partial private key of a targeted user. Subsequently, Hung [15] reported how a Type I adversary (i.e. an uncertified entity) can successfully attack Li *et al*. scheme and extract singer's secret key and certificate of a target entity.

Recently, T. Hyla *et al.* [19] have introduced a new paradigm called Implicit and Explicit Certificates-Based Public Key Cryptography (IEC-PKC) and proposed a concrete implicit and explicit certificates-based encryption scheme (IE-CBE). In the IE-CBE construction, the implicit and explicit certificates are based on a short signature scheme given in [11], [16], which security depends on a k-CAA hard problem. To our best knowledge, an implicit and explicit certificates-based signature scheme based on *k*-CAA problem or its modification still does not exist. Hence, it is challenge and open problem to design such secure signature schemes under the proper security model.

Our contribution. In this paper, inspired by T. Hyla *et al.* [20] results, we propose the first IE-CBS-kCAA signature scheme using Sakai-Kasahara key construction with provable security against *k*-traitors Collusion Attack. The proposed scheme belongs to the public key cryptography schemes based on an implicit and explicit certificates. The main features of our scheme are presented below.

Firstly, a signature verification in IE-CBS-kCAA scheme can be carried out in two modes: in a first mode using an explicit certificate and in a second mode with an implicit

certificate; in the first mode, the verification can be made like in the PKI certificates. The verification in the second mode is similar to many cases of the certificate-based signature schemes and it is carried out without referencing to an explicit certificate. This type of dual nature of IE-CBS-kCAA scheme is a unique feature among other IEC-PKC (see T. Hyla *et al*. [21]) signature schemes.

Secondly, the proposed scheme possesses existential unforgeability against adaptive chosen-message and identity attacks (EUF-CMA) under the variation of the collusion attack algorithm with k-traitors (*k*-mCAA) and the discrete logarithm (DL) assumptions. Thirdly, the explicit and implicit certificates in IE-CBS-kCAA scheme are generated in two different algebraic groups. It is similar solution to the work [19], where implicit and explicit certificates belongs to the different groups. However, in both cases it is computationally hard to recreate implicit certificate using only explicit certificate and *vice versa.*

Lastly, the *Sign* algorithm in our scheme uses a random number. Therefore, IE-CBS-kCAA scheme is a randomized scheme. Randomization protects the scheme from known attacks. Definitely, the proposed scheme does not belong to short signature schemes, but it is computationally more efficient and has a similar signature length comparing to another (implicit) certificate-based signature schemes in [16], [20] with the similar security level.

Paper Organisation. The rest of this paper is organized as follows. In the next section, we briefly review some basic knowledge. Before presenting our results, we first present the notion of an implicit and explicit certificates-based signature scheme and its security model against two different types of attacks (see Section 3). In Section 4, the IE-CBS-kCAA randomized signature scheme from pairings is proposed, whose security in the random oracle, is analysed in Section 4. The efficiency of our scheme is discussed in Section 5. At last, we present our concluding remarks.

2 Preliminaries

In this section, we will review hard mathematical problems and security assumptions required in this paper. The definition and notation for an asymmetric bilinear map $\hat{e}: G_1 \times G_2 \to G_T$ can be found in [5].

Definition 1 (Discrete Logarithm (DL) problem). Given the generator $P \in G_1$ and $T \in G_1^*$ compute $a \in Z_p^*$ such that $T = aP$. The DL is (t, ε_{DL}) – hard if the success probability of any probabilistic t-polynomial-time algorithm A_{DL} solving the DL problem in G_1 is defined to be:

$$Adv_{DL}^{A} = \Pr\left\{ A_{DL}(P, aP) = a \,\middle|\, a \in Z_p^* \right\} < \varepsilon_{DL} \tag{1}$$

The DL assumption states that the probability Adv_{DL}^{A} is negligible for every probabilistic polynomial – time algorithm A.

Definition 2 (*k*-CAA problem, S. Mitsunari *et al.* [1]). For a positive integer *k* and $s \in Z_p^*$, $Q \in G_2$, given

$$\left\{ Q, Q_0 = sQ, h^*, h_1, \ldots, h_k \in Z_p^*,\ h^* \notin \{h_1, \ldots, h_k\}, \frac{1}{h_1 + s}Q, \ldots, \frac{1}{h_k + s}Q \right\},$$

compute a new pair $\left\{ h^*, \frac{1}{h^* + s}Q \right\}$. We say that the *k*-CAA is (t, $\varepsilon_{k\text{-}CAA}$)-hard if, for all *t*-time adversaries $A_{k\text{-}CAA}$, we have

$$Adv_{k-CAA}^{A} = \Pr\left\{ \begin{array}{l} A_{k-CAA}\left(Q, Q_0, \frac{1}{h_1 + s}Q, \ldots, \frac{1}{h_k + s}Q\right) = \frac{1}{h^* + s}Q \\ \left| s \in Z_p^*, Q \in G_2, h^*, h_1, \ldots, h_k \in Z_p^*, h^* \notin \{h_1, \ldots, h_k\} \right. \end{array} \right\} < \varepsilon_{k-CAA}$$

The *k*-CAA problem is believed to be hard, i.e., there is no polynomial time algorithm to solve it with non-negligible probability. For the needs of this paper we define a new variant of *k*-CAA problem hereinafter referred to as the *k*-mCAA problem (compare [22]).

Definition 3 (*k*-mCAA problem). For randomly picked values $s, r^*, h^*, r_1, h_1, \ldots, r_k, h_k \in Z_p^*$, $Q \in G_2$, given

$$\left\{ \begin{array}{l} Q, Q_0 = sQ, h^*, h_1, \ldots, h_k \in Z_p^*, h^* \notin \{h_1, \ldots, h_k\}, \\ r^*Q, r_1Q, \ldots, r_kQ,\ r^*Q \notin \{r_1Q, \ldots, r_kQ\}, \frac{1}{r_1h_1 + s}Q, \ldots, \frac{1}{r_kh_k + s}Q \end{array} \right\},$$

compute a new pair $\left\{ h^*, \frac{1}{r^*h^* + s}Q \right\}$. We say that the *k*-mCAA problem is (t, $\varepsilon_{k\text{-}mCAA}$)-hard if, for all *t*-time adversaries $A_{k\text{-}mCAA}$, we have

$$Adv_{k-mCAA}^{A} = \Pr\left\{ \begin{array}{l} A_{k-mCAA}\left(\begin{array}{l} Q, Q_0, r_1Q, \ldots, r_kQ, \\ \frac{1}{r_1h_1 + s}Q, \ldots, \frac{1}{r_kh_k + s}Q \end{array} \right) = \frac{1}{r^*h^* + s}Q \\ \left| \begin{array}{l} s \in Z_p^*, Q \in G_2, h^*, h_1, \ldots, h_k \in Z_p^*, \\ h^* \notin \{h_1, \ldots, h_k\},\ r^*Q \notin \{r_1Q, \ldots, r_kQ\} \end{array} \right. \end{array} \right\} < \varepsilon_{k-mCAA}.$$

The *k*-mCAA assumption states that the probability Adv_{k-mCAA}^{A} is negligible for every probabilistic polynomial-time algorithm *A*. It is worth to noting that the *k*-mCAA is hard to break, because even if *h* is known, the probability for finding a number $x \in Z_p^*$ such that $x = (s + r^* h)^{-1} \bmod p$ with two unknowns *s* and r^* is negligible and equal to $(p(p\text{-}1))^{-1}$).

Definition 3 was derived from the *k*-CAA3 problem definition, formulated by S. H. Islam *et al*. [22]. In contrast to the original definition, it was assumed that the values $r^*P, r_1P, \ldots, r_kP$ are input to the *k*-mCAA problem.

Let assume that $r^* = 1$ and $r_i = 1, (i = 1, \ldots, k)$. Then the *k-mCAA* problem is transformed into *k*-CAA problem. Thus, k-CAA problem can be seen as a special case of the *k*-mCAA problem. Similar, if $r_i = r^*, (i = 1, \ldots, k)$, then the *k*-mCAA problem is equivalent to the original *k*-CAA3 problem form [22].

3 Security model of IE-CBS-kCAA scheme

In this paper, we consider only one kind of security notion, existential unforgeability (EUF) under chosen-message attack (CMA) in the random oracle model (EUF-CMA). In this attack, an adversary, allowed to ask the signer to sign any message of its choice adaptively according to previous answers, should not be able to generate a new valid message-signature pair.

3.1 Adversaries and oracles

The security model of the proposed IE-CBS-kCAA scheme, hereinafter referred to as EUF-IECBS-kCAA -CMA, is defined by two games between challenger *C* and adversary *A*, assuming that the adversary chooses which game to play. In both cases, adversary $A = (A_1, A_2)$ is trying to break the EUF-CMA security of the IE-CBS-kCAA scheme, i.e., the formal model describing existential unforgeability. To describe these games, we use the widely accepted two types of adversaries with different capabilities: **Type I Adversary** and **Type II Adversary** (e.g., T. Hyla, *et al.* [20]).

Type I Adversary (A_1) is able to compromise the user's secret key or replace the user's public key, but is unable to gain TA's master secret key nor the user's partial private key issued by TA. We assume that adversary A_1 models the security against non-certified users and eavesdroppers, i.e., against the users, who are not registered and do not have certificates issued by the TA.

Type II Adversary (A_2) can obtain TA's master secret key and the user's implicit certificate, but cannot compromise the user's secret key nor replace her/his target public key. In this case, it is reasonable to consider attack scenarios that targets certified users, i.e., users who come into possession of a private/public key pair and explicit certificates before the master key *s* becomes known to the adversary.

The formal security model of the implicit and explicit certificates-based signature schemes divides the potential adversaries according to their attack power and classified the Type I/II adversary into three kinds (see Li, J., *et al.* [14] and Huang, X., *et al.* [12]): Normal Adversary, Strong Adversary and Super Adversary. The most power attacks are related to Super Type I/II Adversary, which may issue the following queries:

Create-User-Query. If a user identity *ID* has already been created, nothing is carried out by the oracle. Otherwise, challenger *C* runs the algorithms **Create-User** to obtain the secret value s_{ID} and the partial public key Pk_{ID}. Then it adds

$\langle ID, s_{ID}, Pk_{ID} \rangle$ to the L_U list. In this case, the user with identity *ID* is said to be created. In both cases, Pk_{ID} is returned.

Public-Key-Replacement-Query. If *ID* is created, the oracle takes as input a query $(CI_{ID}, Pk_{ID}, Pk'_{ID})$, finds the user *ID* in the list L_U and replaces the original user public key Pk_{ID} with $Pk'_{ID} = s'_{ID}P$. Otherwise, no action will be taken. Note that the adversary is not required to provide the secret value s'_{ID}.

Corruption-Query. This oracle takes as input a query *ID*. It browses the list L_U and if *ID* denotes the identity which has been created, the oracle outputs the secret key s_{ID}.

Implicit-Cert-Gen-Query. On input of an identity index (ID, Pk_{ID}), this oracle returns an implicit certificate $\overline{Sk}_{ID}$ whenever the user with identity index (ID, Pk_{ID}) has been created. Otherwise, a symbol $\perp$ is returned.

Explicit-Cert-Gen-Query. For a certificate request for a user with identity index (CI_{ID}, Pk_{ID}), this oracle returns an explicit certificate $Cert_{ID}$ and two additional components ($\underline{R}_{ID}$, $\underline{\underline{R}}_{ID}$). If the user with $CI_{ID}.ID$[1] is not created, the symbol $\perp$ is returned.

Super-Sign-Query. If *ID* has not been created, the oracle returns $\perp$. Otherwise, it takes as input a query $(m, CI_{ID}, Pk_{ID}, \underline{R}_{ID}, \underline{\underline{R}}_{ID})$, where *m* denotes the message to be signed, and then returns a valid signature σ_{ID} such that **Verify** (*params*, *m*, σ, CI_{ID}, Pk_{ID}, $\underline{R}_{ID}$, $\underline{\underline{R}}_{ID}$, $Cert_{ID}$) $\rightarrow$ *true*. Here Pk_{ID} denotes the user ID_S's current public key in the list L_U and can be replaced by A_1 or returned from the oracle **Create-User-Query**.

Remark 1. A Super Type II Adversary, who simulates the malicious certifier, is not allowed to make any requests to **Implicit-Cert-Gen-Query** and **Explicit-Cert-Gen-Query**.

3.2 Games against a Super Type I/II Adversary

To investigate the existential unforgeability of IE-CBS-kCAA scheme against Super Type I/II Adversary (A_1/A_2 in short) we can now define two games (**Game I** and **Game II**) between a challenger C and the two types of adversaries (A_1 and A_2, respectively).

Game I. This game is executed between challenger *C a*nd an adversary A_1 under an adaptively chosen message and chosen user's identity *ID*.

Setup. Challenger *C* executes algorithm **Setup** (1^k) $\rightarrow$ (*s*, *params*) in the IE-CBS-kCAA scheme to obtain the public parameter *params* and master secret key *s*. Adversary A_1 is given *params*, but the challenger *C* keeps the master secret key *s* secret.

---

[1] This notation means the filed *ID* of the user's certificate information CI_{ID}.

Queries. In this phase, A_1 can adaptively submit queries to following oracles defined in Section 3.1: **Create-User-Query**, **Implicit-Cert-Gen-Query**, **Explicit-Cert-Gen-Query**, **Public-Key-Replacement-Query**, **Corruption-Query** and **Super-Sign-Query**.

Forgery. Eventually, after some or all queries, adversary A_1 outputs a forgery $\left(\hat{m}, \hat{\sigma} = \left(\hat{h}, \hat{w}_1, \hat{w}_2, \hat{\Sigma}\right), ID, CI_{ID}, Pk_{ID}, \hat{\underline{R}}_{ID}, \hat{\underline{\underline{R}}}_{ID}, Cert_{ID}\right)$.

Constrains. Adversary A_1 wins the game if the forgery satisfies the following requirements:

(a) $\hat{\sigma}$ is a valid signature on the message $\hat{m}$ under the public key Pk_{ID} and the explicit certificate $Cert_{ID}$, i.e. **Verify** (*params*, $\hat{m}$, $\hat{\sigma}$, CI_{ID}, Pk_{ID}, $\hat{\underline{R}}_{ID}$, $\hat{\underline{\underline{R}}}_{ID}$, $Cert_{ID}$) $\rightarrow$ *true*. Here, Pk_{ID} is chosen by A_1 and might not be the one returned from **Create-User-Query** oracle.

(b) (ID, Pk_{ID}) and (CI_{ID}, Pk_{ID}) has never been submitted to respective oracles **Implicit-Cert-Gen-Query** and **Explicit-Cert-Gen-Query**.

(c) *ID* has never appeared as one of **Corruption-Query**.

(d) ($\hat{m}$, CI_{ID}, Pk_{ID}, $\hat{\underline{R}}_{ID}$, $\hat{\underline{\underline{R}}}_{ID}$) has never been submitted to oracle **Super-Sign-Query**.

The success probability that an adaptive chosen message and adversary A_1 with chosen identity index (ID, Pk_{ID}) wins the above game is defined as $Adv^{A_1}_{\text{IE-CBS-kCAA}}$.

Game II. In this Game an adversary A_2 with chosen identity index (ID, Pk_{ID}) interacts with its challenger *C* under an adaptively chosen message.

Setup. Challenger *C* executes algorithm **Setup** (1^k) $\rightarrow$ (*s*, *params*) in the IE-CBS-kCAA scheme to obtain the public parameter *params* and master secret key *s*. *C* then sends (*params*, *s*) to the adversary A_2.

Queries. In this phase, challenger *C* runs adversary A_2 can adaptively access the following oracles: **Create-User-Query**, **Public-Key-Replacement-Query**, **Corruption-Query** and **Super-Sign-Query**. The oracles **Implicit-Cert-Gen-Query** and **Explicit-Cert-Gen-Query** are not accessible and no longer needed, as adversary A_2, which simply holds the master key *s*, can now generate all user partial keys and certificates.

Forgery. At the end of this phase, after some or all queries, adversary A_2 outputs the forgery $\left(\hat{m}, \hat{\sigma} = \left(\hat{h}, \hat{w}_1, \hat{w}_2, \hat{\Sigma}\right), ID, CI_{ID}, Pk_{ID}, \hat{\underline{R}}_{ID}, \hat{\underline{\underline{R}}}_{ID}, Cert_{ID}\right)$.

Constrains. Adversary A_2 wins the game if the forgery satisfies the following requirements:

(a) $\hat{\sigma}$ is a valid signature on the message $\hat{m}$ under the public key Pk_{ID} and the explicit certificate $Cert_{ID}$, i.e. **Verify** (*params*, $\hat{m}$, $\hat{\sigma}$, CI_{ID}, Pk_{ID}, $\hat{\underline{R}}_{ID}$,

$\hat{\underline{R}}_{ID}$, $Cert_{ID}$) → *true*. Here, Pk_{ID} is the output returned by **Create-User-Query** oracle for *ID*.

(b) *ID* has never appeared as one of **Corruption-Query**.

(c) ($\hat{m}$, CI_{ID}, Pk_{ID}, $\hat{\underline{R}}_{ID}$, $\hat{\underline{R}}_{ID}$) has never been submitted to oracle **Super-Sign-Query**.

In this game, adversary A_2 may call the **Public-Key-Replacement-Query** oracle and obtain all secrets corresponding to identity indices other than (ID, Pk_{ID}).

The success probability that an adaptive chosen message and adversary A_2 with chosen identity index (ID, Pk_{ID}) wins the above game is defined as $Adv^{A_2}_{\text{IE-CBS-kCAA}}$.

Definition 4. An implicit and explicit certificate signature scheme IE-CBS-kCAA has existential unforgeability against chosen message attacks (EUF-IECBS-kCAA-CMA), if no probabilistic polynomial-time adversary has non-negligible probability to win Game I and Game II.

4 A new implicit and explicit certificates-based signature scheme IE-CBS-kCAA

The IE-CBS-kCAA scheme contains seven polynomial time algorithms: **Setup**, **Create-User**, **Implicit-Cert-Gen**, **Explicit-Cert-Gen**, **Set-Private-Key**, **Sign** and **Verify**. A detailed description of all algorithms of IE-CBS-kCAA scheme is presented below:

1. **Setup**: the system parameters are $params = \{G_1, G_2, G_T, \hat{e}, p, P, P_0, Q, Q_0, H_1, H_2\}$, where $|G_1| = |G_2| = |G_T| = p$ for some prime number $p \geq 2^k$ (k is the system security number), (P, Q) - generators of respectively G_1 and G_2 such that $\hat{e}(P, Q) = g$, $P_0 = sP$ and $Q_0 = sQ$ - system's master public keys with the master secret key $s \in Z_p^*$, $H_1, H_2 : \Gamma \times \{0,1\}^* \to Z_p$ are two secure cryptographic hash functions. The $\Gamma \in (0,1)^*$ means a string space that can be used to define a user with the identity *ID*. In the cases, when *ID* contains more information other than the identity we will mark it as *CI* (see below).

2. **Create-User** (*params*, ID_S): the user ID_S chooses a random number $s_{ID_S} \in Z_p^*$, sets s_{ID_S} as the secret key and produces the corresponding public key $Pk_{ID_S} = s_{ID_S} P$; the resulting public key is widely and freely distributed, e.g., the TA publishes them in its public repository.

3. **Implicit-Cert-Gen** (*params*, *s*, ID_S, Pk_{ID_S}): given ID_S presenting *S*'s identity, his public key Pk_{ID_S}, the trust authority *TA*:

(a) composes the user's certificate information CI_{ID_S}, including the TA's public keys (P_0, Q_0), identifiers ID_S and ID_{TA} of the user *S* and the *TA*, respectively, and the time period τ for which the information CI_{ID_S} is valid;

(b) randomly selects $r_{ID_S} \in Z_p^*$ and computes $(\underline{R}_{ID_S}, \underline{\underline{R}}_{ID_S}) = (r_{ID_S} P, r_{ID_S} Q)$;

(a) for Pk_{ID_S} and $\left(\underline{R}_{ID_S}, \underline{\underline{R}}_{ID_S}\right)$

(b) computes $q_{ID_S} = H_1(CI_{ID_S}, Pk_{ID_S}, \underline{R}_{ID_S}, \underline{\underline{R}}_{ID_S})$, generates *S*'s private key as:

$$\overline{Sk}_{ID_S} = \frac{1}{s + r_{ID_S} q_{ID_S}} Q \tag{2}$$

and transmits it to the user *S* secretly; in addition, TA sends also $(CI_{ID_S}, \underline{R}_{ID_S}, \underline{\underline{R}}_{ID_S})$.

4. **Explicit-Cert-Gen** (*params*, ID_S, *s*, r_{ID_S}, q_{ID_S}): TA authority, using parameters received from *S* and values calculated during execution **Implicit-Cert-Gen** algorithm, generates an explicit certificate $Cert_{ID_S}$ of a signer *S*:

(a) TA generates the explicit certificate for an entity *S*, which binds its identity with the public key components:

$$Cert_{ID_S} = \frac{1}{s + r_{ID_S} q_{ID_S}} P \tag{3}$$

(b) TA sends $Cert_{ID_S}$ to an entity *S*.

5. **Set-Private-Key** (*params*, CI_{ID_S}, s_{ID_S}, Pk_{ID_S}, $\underline{R}_{ID_S}$, $\underline{\underline{R}}_{ID_S}$, $\overline{Sk}_{ID_S}$): the user *S* checks if $\hat{e}(q_{ID_S} R_{ID_S} + P_0, \overline{Sk}_{ID_S}) = \hat{e}(P, Q) = g$, and then formulates his private key in the form $Sk_{ID_S} = (s_{ID_S}, \overline{Sk}_{ID_S})$.

6. **Sign** (*params*, *m*, CI_{ID_S}, (Sk_{ID_S}, Pk_{ID_S}, $\underline{R}_{ID}$, $\underline{\underline{R}}_{ID}$)): to sign a message $m \in \{0,1\}^*$, a signer *S* performs the following steps:

(a) pick two random numbers $k_1, k_2 \in_R Z_p^*$;

(b) computes a hash value $q_{ID_S} = H_1(CI_{ID_S}, Pk_{ID_S}, \underline{R}_{ID_S}, \underline{\underline{R}}_{ID_S})$ and then generates the signature $\sigma = (h, w_1, w_2, \Sigma)$, where $h = H_2(m, k_1 P, U, q_{ID_S})$,

$$\Sigma = \frac{k_1 - k_2^{-1} h}{k_1 h + s_{ID_S}} \overline{Sk}_{ID_S} \tag{4}$$

$w_1 = k_1 - h\, s_{ID_S} \pmod p$, $w_2 = k_2\left(k_1\, h + s_{ID_S}\right) \pmod p$, while $U = g^{\,k_1 k_2}$.

7. **Verify** (*params, m,* σ, CI_{ID}, Pk_{ID}, $\underline{R}_{ID}$, $\underline{\underline{R}}_{ID}$, $Cert_{ID}$): to verify the message/signature/certificate triple, i.e. $\left(m, \sigma = (h, w_1, w_2, \Sigma), Cert_{ID_S}\right)$, *V* performs the following steps:
   (a) computes a hash $q_{ID_S} = H_1\left(CI_{ID_S}, Pk_{ID_S}, \underline{R}_{ID_S}, \underline{\underline{R}}_{ID_S}\right)$ and then the values $U' = \hat{e}\left(q_{ID_S}\, \underline{R}_{ID} + P_0, \Sigma\right)^{w_2} \hat{e}\left(Cert_{ID_S}, q_{ID_S}\, \underline{\underline{R}}_{ID} + Q_0\right)^h$, $\overline{k_1 P} = w_1 P + h P_{ID_S}$;
   (b) if $h \equiv H_2\left(m, \overline{k_1 P}, U', q_{ID_S}\right)$ and $\hat{e}\left(Cert_{ID_S}, q_{ID_S}\, \underline{\underline{R}}_{ID} + Q_0\right) \equiv g$, then returns *accept*, else *reject*.

4.1 Correctness of the IE-CBS-kCAA scheme

Assume that digital signature σ and an explicit certificate $cert_{ID_S}$ have been generated using the **Sign** and **Explicit-Cert-Gen** algorithms, respectively. Therefore, σ is a valid signature on message *m* because it is accepted by the verification algorithm **Verify**:

$$
\begin{aligned}
U' &= \hat{e}\left(q_{ID_S}\, \underline{R}_{ID} + P_0, \Sigma\right)^{w_2} \hat{e}\left(Cert_{ID_S}, q_{ID_S}\, \underline{\underline{R}}_{ID} + Q_0\right)^h \\
&= \hat{e}\left(q_{ID_S}\, \underline{R}_{ID} + P_0, \frac{k_1 - k_2^{-1} h}{k_1\, h + s_{ID_S}}\, \overline{Sk}_{ID_S}\right)^{k_2\left(k_1\, h + s_{ID_S}\right)} \hat{e}\left(Cert_{ID_S}, q_{ID_S}\, \underline{\underline{R}}_{ID} + Q_0\right)^h \\
&= \hat{e}\left(q_{ID_S}\, \underline{R}_{ID} + P_0, (k_1 k_2 - h)\overline{Sk}_{ID_S}\right) \hat{e}\left(Cert_{ID_S}, q_{ID_S}\, \underline{\underline{R}}_{ID} + Q_0\right)^h \\
&= \hat{e}\left(\left(s + r_{ID_S} q_{ID_S}\right)P, (k_1 k_2 - h)\frac{1}{s + r_{ID_S} q_{ID_S}} Q\right) \\
&\qquad\qquad \hat{e}\left(\frac{1}{s + r_{ID_S} q_{ID_S}} P, h\left(s + r_{ID_S} q_{ID_S}\right)Q\right) \qquad (5) \\
&= \hat{e}(P, Q)^{k_1 k_2} \\
&= U
\end{aligned}
$$

Hence,

$$
\begin{aligned}
h' &= H_2\left(m, \overline{k_1 P}, U', q_{ID_S}\right) \\
&= H_2\left(m, wP + hP_{ID_S}, U, q_{ID_S}\right) \qquad (6) \\
&= h
\end{aligned}
$$

Furthermore, it is now easy to prove the correctness of the explicit certificate:

$$
\begin{aligned}
g' &= \hat{e}\left(Cert_{ID_S},\, q_{ID_S}\underline{\underline{R}}_{ID} + Q_0\right) \\
&= \hat{e}\left(\frac{1}{s + r_{ID_S} q_{ID_S}} P,\, \left(s + r_{ID_S} q_{ID_S}\right) Q\right) \\
&= \hat{e}(P, Q) \\
&= g
\end{aligned} \tag{7}
$$

4.2 Security analysis of IE-CBS-kCAA scheme

We prove the security of IE-CBS-kCAA scheme by using the approach of reducing the security of a higher-level construction to a lower-level primitive. More precisely we reduce the existence of an adversary breaking the protocol into an algorithm that was able to solve the respective *k*-mCAA or a discrete logarithm (DL) problem with non-negligible probability. In our reductions, we use the multiple forking lemma, proposed by Boldyreva and *et al.* [23], in the way similar to [20].

Lemma 1. Suppose that the hash functions H_1 and H_2 are random oracles, and in Game 1 against IE-CBS-kCAA scheme, adversary A_1 plays the role of an uncertified user. The proposed implicit and explicit certificates signature scheme IE-CBS-kCAA is existential unforgeable against a Super Type I adversary A_1 under the *k*-mCAA assumption.

Proof (sketch). Similarly to approach given in [20], our reduction was proceed into two steps. First, we described an intermediate algorithm B_1 (i.e. the wrapper) that interacts with the adversary A_1 and returned a side output. Second, we showed how to build a reduction algorithm R_1 that has launched the forking game $MF_{B,1}$ on the wrapper B_1. As a result, an algorithm R_1 was obtained one pairing equation with one unknowns and indeed returned the correct solution to the *k*-mCAA problem instance.

Algorithm R1 obtains two valid signature forgeries, each of them in the form $\hat{\sigma}_i = \left(\hat{m}, \hat{h}_i, \hat{w}_{1,i}, \hat{w}_{2,i}, \hat{\Sigma}_i, Pk_{ID}, \underline{R}_{ID}, \underline{\underline{R}}_{ID}, Cert_{ID}\right)$, ($i$ =0, 1) for the same message $\hat{m}$, the public key Pk_{ID}, the explicit certificate $Cert_{ID}$ and ($\underline{R}_{ID}$, $\underline{\underline{R}}_{ID}$). If both forgeries are valid, then R_1 obtains two sets of side-outputs σ_0, σ_1, where σ_i (for i = 0, 1) is of the form ($\hat{h}_i, \hat{U}_i, \hat{\Sigma}_i, \hat{w}_{2,i}, Pk_{ID}, \underline{R}_{ID}, \underline{\underline{R}}_{ID}$). Additionally, we assume that ($\hat{U}_0 = \hat{U}_1$). Based on these two sets of side-outputs σ_0, σ_1, the following equation is fulfilled

$$
\begin{aligned}
&\hat{e}\left(q_{ID}\underline{R}_{ID} + P_0, \hat{\Sigma}_0\right)^{\hat{w}_{2,0}} \hat{e}\left(Cert_{ID}, q_{ID}\underline{\underline{R}}_{ID} + Q_0\right)^{\hat{h}_0} = \\
&\qquad \hat{e}\left(q_{ID}\underline{R}_{ID} + P_0, \hat{\Sigma}_1\right)^{\hat{w}_{2,1}} \hat{e}\left(Cert_{ID}, q_{ID}\underline{\underline{R}}_{ID} + Q_0\right)^{\hat{h}_1}
\end{aligned} \tag{8}
$$

By making the suitable arrangements, the equation (10) can be converted to the form:

$$
\hat{e}\left(q_{ID}\underline{R}_{ID} + P_0, \hat{w}_{2,0}\hat{\Sigma}_0 + \hat{h}_0\overline{Sk}_{ID}\right) = \hat{e}\left(q_{ID}\underline{R}_{ID} + P_0, \hat{w}_{2,1}\hat{\Sigma}_0 + \hat{h}_1\overline{Sk}_{ID}\right) \tag{9}
$$

Finally, we get the solution to the *k*-mCAA problem challenge (see Definition 4):

$$\overline{Sk}_{ID} = \frac{\left(\hat{w}_{2,0}\,\hat{\Sigma}_0 - \hat{w}_{2,1}\,\hat{\Sigma}_1\right)}{\left(\hat{h}_0 - \hat{h}_1\right)} \tag{10}$$

Now, for the Game 2 implemented with Super Type 2 adversary, in which the adversary models a certified entity, we demand that a signer is honest and his tuple $\left(ID, Pk_{ID}, Cert_{ID}\right)$ has been previously registered with the TA. For this assumption, the following lemma can be proved in the random oracle model:

Lemma 2. Suppose that the hash functions H_1 and H_2 are random oracles. The proposed implicit and explicit certificates signature scheme IE-CBS-kCAA is existentially unforgeable against a Super Type II adversary under the *DL* problem.

The proof is similar to the proof of [20] and is omitted here.

Table 1. Performance comparison (based on [20])

Scheme	Type	Public key size	Signature size	Sign	Verify	Security level
LHMSW (J. Li *et al.* [14])	I-CBS	$\lvert G_1\rvert$	$2\lvert G_1\rvert$	$3\,M_G$	$3\,\hat{e}$	Normal A_1 and Normal A_2
LHZX (J. Li *et al.* [24])	I-CBS	$2\lvert G_1\rvert$	$\lvert G_1\rvert$	M_G	$\hat{e}\ +\ M_G$	Normal A_1 and Super A_2
CBSa (Kang *et al.* [13])	I-CBS	$\lvert G_1\rvert$	$3\lvert G_1\rvert$	$3\,M_G$	$3\,\hat{e}+2M_G$	Strong A_1 and Strong A_2
WMSH Scheme II (Wu, W., *et al.* [25])	I-CBS	$\lvert G_1\rvert$	$\lvert G_1\rvert+2\lvert Z_p\rvert$	$\hat{e}+4\,M_G$	$2\,\hat{e}+3M_G$	Super A_1 and Super A_2
IE-CBHS scheme (Hyla, T., *et al.* [20])	IE-CBS	$\lvert G_1\rvert$	$\lvert G_1\rvert+2\lvert Z_p\rvert$	$\hat{e}+3\,M_G$	$2\,\hat{e}+7M_G$	Super A_1 and Super A_2
Our IE-CBS-kCAA scheme	IE-CBS	$\lvert G_1\rvert$	$\lvert G_2\rvert+3\lvert Z_p\rvert$	$2\,M_G+P_{GT}$	$2\,\hat{e}+6M_G$	Super A_1 and Super A_2

5 Performance comparison

In this section, we compare our implicit and explicit certificates-based signature scheme IE-CBS-kCAA to other existing schemes with similar constructions. The comparison is based on results presented in [20]. Operations like: hashing, operations in Z_p^* (inversion, addition, multiplication), multiplication in G_T and addition in G_1 or G_2 can be omitted in efficiency comparison, because they are several orders of magnitude faster

when compared with pairings, scalar multiplications in G_1 or G_2 and exponentiations in G_T. In Table 1, proposed IE-CBS-kCAA is compared to other schemes ($|G_1|$ and $|Z_p|$ is the bit length of an element in G_2 and Z_p, M_G is a scalar multiplication in G_1 or G_1 and $\hat{e}$ is a bilinear pairing on $G_1 \times G_2$ and P_{GT} is exponentiation in G_T). Our proposed scheme has the same security level as WMSH Scheme II and IE-CBHS scheme. It requires the similar number of time-intensive operation when comparing to other schemes and is slightly faster than IE-CBHS scheme.

6 Conclusions

The paper contains the IE-CBS-kCAA signature scheme that has been built on a new paradigm called Implicit and Explicit Certificates-Based Public Key Cryptography (IEC-PKC) [19]. Using this paradigm we propose the first signature scheme based on the implicit and explicit certificates resistant against k-traitors collusion attacks with Sakai-Kasahara key construction. We proved that our scheme is existential unforgeable against the adaptive chosen message and identity attacks based on the variation of Collusion Attack Algorithm with *k* traitors (k-mCAA) and discrete logarithm (DL) assumptions in the random oracle model with Super Type I/II Adversaries.

The most time-consuming operation in a signature scheme from pairings is the computation of the pairing. Our scheme contains no one pairing operation in the signing phase and two pairing operation in the signature verification phase. Therefore, IE-CBS-kCAA scheme, when compared with other signature schemes (Table 1), has similar efficiency and is both more flexible, and more useful in practice.

References

1. Mitsunari, S., Sakai, R., Kasahara, M.: A new traitor tracing. IEICE Transations, vol. E85-A, no.2, pp.481-484, (2002)
2. Chen, L., Cheng, Z.: Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. In: Smart, N.P. (ed.) Cryptography and Coding: 10th IMA International Conference, Cirencester, UK, December 19-21, 2005. Proceedings. pp. 442–459. Springer, Berlin Heidelberg, (2005)
3. Sakai, R., Kasahara, M.: ID based Cryptosystems with Pairing on Elliptic Curve. http://eprint.iacr.org/2003/054, (2003)
4. Zhang, F., Safavi-Naini, R., Susilo, W.: An Efficient Signature Scheme from Bilinear Pairings and Its Applications. In: Bao, F., Deng, R., and Zhou, J. (eds.) Public Key Cryptography -- PKC 2004: Singapore, March 1-4, 2004, pp. 277–290. Springer, Berlin Heidelberg (2004).
5. Barreto, P.S.L.M., Libert, B., McCullagh, N., Quisquater, J.-J.: Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps. In: Roy, B. (ed.) Advances in Cryptology - ASIACRYPT 2005: Chennai, India, December 4-8, 2005, pp. 515–532. Springer, Berlin Heidelberg (2005)
6. Hu, B.C., Wong, D.S., Zhang, Z., Deng, X.: Certificateless signature: a new security model and an improved generic construction. Des. Codes Cryptogr. 42, 109–126 (2007)

7. Du, H., Wen, Q.: An Efficient Identity-Based Short Signature Scheme from Bilinear Pairings. In: 2007 International Conference on Computational Intelligence and Security (CIS 2007). pp. 725–729 (2007)
8. Al-Riyami, S. S., Paterson, K. G.: Certificateless public key cryptography. In: Chi-Sung Laih (Ed.), Advances in Cryptology – ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
9. Girault, M.: Self-certified public keys. In: Davies, D.W. (ed.) Advances in Cryptology --- EUROCRYPT '91: Brighton, UK, April 8--11, 1991. pp. 490–497. Springer, Berlin Heidelberg (1991)
10. Du, H., Wen, Q.: Efficient and provably-secure certificateless short signature scheme from bilinear pairings. Computer Standards and Interfaces, vol. 31, no. 2, pp. 390–394 (2009)
11. Huang, X., Susilo, W., Mu, Y., Zhang, F.: On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In: Desmedt, Y.G., Wang, H., Mu, Y., Li, Y. (Eds.) CANS 2005. LNCS, vol. 3810, pp. 13–25. Springer, Heidelberg (2005)
12. Huang, X., Mu, Y., Susilo W., Wong, D. S., Wu, W.: Certificateless Signatures: New Schemes and Security Models. The Computer Journal. vol. 55 no. 4, pp. 457-474 (2011)
13. G. Kang, J. H. Park, S. G. Hahn A Certificate-based Signature Scheme, in: Okamoto, T. (ed.): CT-RSA 2004. LNCS, vol. 2964, pp.99–111. Springer-Verlag (2004)
14. Li, J., Huang, X., Mu, Y., Susilo, W., Wu, Q.: Certificate-Based Signature: Security Model and Efficient Construction. In J. Lopez, P. Samarati, and J.L. Ferrer (Eds.): EuroPKI 2007. LNCS, vol. 4582, pp. 110–125, 2007. Springer-Verlag, Berlin Heidelberg (2007)
15. Hung, Y.-H., Huang, S.-S, Tseng, Y.-M.: A Short Certificate-based Signature Scheme with Provable Security. Information Technology and Control, vol. 45, no. 3, pp. 243-253 (2015)
16. Wu, W., Mu, Y., Susilo, W., Huang, X.: Certificate-based Signatures Revisited, Journal of Universal Computer Science, vol. 15, no. 8, pp.1659-1684 (2009)
17. Liu, J.K., Bao, F., Zhou, J.: Short and Efficient Certificate-Based Signature. In: Casares-Giner, V., Manzoni, P., and Pont, A. (eds.) NETWORKING 2011, Valencia, Spain, May 13, 2011, Revised Selected Papers. pp. 167–178. Springer Berlin Heidelberg (2011).
18. Cheng, L., Xiao, Y., Wang, G.: Cryptanalysis of a certificate-based on signature scheme. Procedia Engineering, vol. 29, 2821–2825 (2012)
19. Hyla, T., Maćków, W., Pejaś, J.: Implicit and Explicit Certificates-Based Encryption Scheme, K. Saeed and V. Snášel (Eds.): CISIM 2014. LNCS, vol. 8838, pp. 651–666. Springer-Verlag (2014)
20. Hyla, T., Pejaś, J.: A Hess-like Signature Scheme based on Implicit and Explicit Certificates. The Computer Journal (2016), doi: 10.1093/comjnl/bxw052, http://comjnl.oxfordjournals.org/cgi/reprint/bxw052
21. Hyla, T., Pejaś, J.: Non-standard Certification Models for Pairing Based Cryptography. In: Kobayashi, S., Piegat, A., Pejaś, J., El Fray, I., and Kacprzyk, J. (eds.) Hard and Soft Computing for Artificial Intelligence, Multimedia and Security. pp. 167–181. Springer International Publishing, Cham (2017)
22. Islam, S. H., Biswas, G. P.: An Efficient and Provably-secure Digital signature Scheme based on Elliptic Curve Bilinear Pairings. Theoretical and Applied Informatics, vol.24, no. 2, pp. 109-118 (2012)
23. Boldyreva, A., Palacio, A., Warinschi, B.: Secure proxy signature schemes for delegation of signing rights. Journal of Cryptology, vol. 25, issue 1, 57-115 (2012)
24. Li, J., Huang, X., Zhang, Y. and Xu, L.: An efficient short certificate-based signature scheme. J. Syst. Soft., vol. 85, 314–322 (2012)