Generic Construction of Certificateless Signature*

Dae Hyun Yum and Pil Joong Lee**

IS Lab., Dept. of Electronic and Electrical Eng., POSTECH, Republic of Korea
{dhyum, pjl}@postech.ac.kr
http://islab.postech.ac.kr

Abstract. To provide the binding between a user and his public key, traditional digital signature schemes use certificates that are signed by a trusted third party. While Shamir's identity-based signature scheme can dispense with certificates, the key escrow of a user's private key is inherent in the identity-based signature scheme. In Asiacrypt 2003, a new digital signature paradigm called the certificateless signature was introduced. The certificateless signature eliminates the need for certificates and does not suffer from the inherent key escrow problem. In this paper, we provide a generic secure construction of a certificateless signature. We also present an extended construction whose trust level is the same as that of a traditional public key signature scheme.

Keywords: Certificateless signature, identity-based signature, public-key signature.

1 Introduction

Public key signature and identity-based signature. A digital signature is one of the most important security primitives in modern cryptography. In a traditional public key signature scheme, methods to guarantee the authenticity of a public key are required, since the public key of the signer is actually a type of random string. To provide the binding between a signer and his public key, the traditional public key signature uses a certificate that is a digitally signed statement issued by the CA (Certification Authority). The need for public key infrastructure supporting certificates is considered the main difficulty in the deployment and management of public key signature schemes. While Shamir's identity-based signature scheme can dispense with certificates, the key escrow of a user's private key is inherent in the identity-based signature scheme [15]; a trusted third party called the PKG (Private Key Generator) manages the generation and distribution of the users' private keys.

* This research was supported by University IT Research Center Project and the Brain Korea 21 Project.

** (on leave at KT Research Center)

H. Wang et al. (Eds.): ACISP 2004, LNCS 3108, pp. 200–211, 2004.

Certificateless signature. In Asiacrypt 2003, the certificateless signature was proposed [1]. A certificateless signature scheme does not require the use of certificates and yet does not have the inherent key escrow problem of the identity-based signature scheme. Unlike the PKG in an identity-based signature scheme, the KGC (Key Generating Center) in a certificateless signature scheme does not have access to the user's private key. The KGC derives a partial private key from the user's identity and the master key. The user then combines the partial private key with some secret information to generate the actual private signing key. The system is not identity-based, because the public key is no longer computable from a user identity. However, no authentication of the public key is necessary and no certificate is required.

Our contribution. In this paper, we provide the generic secure construction of the certificateless signature scheme. While previous constructions are built from bilinear mappings that are implemented using Weil and Tate pairings on elliptic curves, our construction is built from general primitives: a public key signature scheme and an identity-based signature scheme. In addition, we present the extended construction that achieves trust level 3 in the hierarchy of [10].

Related work. In parallel with this work, we researched on the secure construction of certificateless encryption [16]. However, we could not extend the construction of certificateless encryption to the trust level 3.

2 Identity-Based Signature and Certificateless Signature

In this section, we review the definitions and security notions of identity-based signature schemes [6,8,12,15] and certificateless signature schemes [1].

2.1 Identity-Based Signature

Definition 1. *An identity-based signature scheme is a 4-tuple of polynomial time algorithms* $(\mathsf{IB\_Gen}, \mathsf{IB\_Ext}, \mathsf{IB\_Sign}, \mathsf{IB\_Vrfy})$ *such that:*

- $\mathsf{IB\_Gen}$*, the master key and parameter generation algorithm, is a probabilistic algorithm that takes as input a security parameter* 1^k*. It returns a master key* $IBSK^*$ *and a parameter list params.*
- $\mathsf{IB\_Ext}$*, the signing key issuance algorithm, is a deterministic algorithm that takes as input a user identity id and a master key* $IBSK^*$*. It returns the user id's private signing key* $IBSK_{id}$*.*
- $\mathsf{IB\_Sign}$*, the signing algorithm, is a probabilistic algorithm that takes as input a message M, a parameter list params, and a signing key* $IBSK_{id}$*.* $\mathsf{IB\_Sign}^{IBSK_{id}}_{params}(M)$ *returns a signature* α*.*
- $\mathsf{IB\_Vrfy}$*, the verification algorithm, is a deterministic algorithm that takes as input a message M, a user identity id, a parameter list params, and a signature* α*.* $\mathsf{IB\_Vrfy}_{params}(M, id, \alpha)$ *returns a bit b, where* $b = 1$ *means that the signature is accepted.*

In an identity-based signature scheme, IB_Gen and IB_Ext are performed by the PKG. A secret key $IBSK_{id}$ is given to a user id by the PKG through a secure channel. If $\mathsf{IB\_Vrfy}_{params}(M, id, \alpha) = 1$, we say that α is a valid signature of M by the user id. We require that all signatures output by $\mathsf{IB\_Sign}^{IBSK_{id}}_{params}(\cdot)$ are accepted as valid by $\mathsf{IB\_Vrfy}_{params}(\cdot, id, \cdot)$.

For security analysis, we define a key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ that returns a private signing key $IBSK_{id}$ on input id. We also give the adversary access to a signing oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)$ that returns $\mathsf{IB\_Sign}^{IBSK_{id}}_{params}(M)$ on input (M, id). The security goal of an identity-based signature scheme is existential unforgeability. This means that any PPT (probabilistic polynomial time) adversary A should have a negligible probability of generating a valid signature of a new message given access to the key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ and the signing oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)$. Naturally, A is considered successful if it forges a valid signature α of M by a user id where id was not queried to the key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ and (M, id) was not queried to the signing oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)$.

Definition 2. *Let Π_{IB} be an identity-based signature scheme. For any adversary A, we may perform the following experiment:*

$$(IBSK^*, params) \leftarrow \mathsf{IB\_Gen}(1^k);$$
$$(M, id, \alpha) \leftarrow A^{\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot),\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)}(params).$$

We say that A succeeds if $\mathsf{IB\_Vrfy}_{params}(M, id, \alpha) = 1$, id was never submitted to the key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$, and (M, id) was never submitted to the signing oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)$. Denote the probability of A's success by $\mathsf{Succ}_{A,\Pi_{IB}}(k)$. If for any PPT A, the success probability $\mathsf{Succ}_{A,\Pi_{IB}}(k)$ is negligible, we say that Π_{IB} is secure; that is, existentially unforgeable against chosen-message attacks.

2.2 Certificateless Signature

Definition 3. *A certificateless signature scheme is a* 7*-tuple of polynomial time algorithms* (CL_Gen, CL_Ext_Partial_Pri_Key, CL_Set_Sec_Val, CL_Set_Pri_Key, CL_Set_Pub_Key, CL_Sign, CL_Vrfy) *such that:*

- CL_Gen*, the master key and parameter generation algorithm, is a probabilistic algorithm that takes as input a security parameter 1^k. It returns a master key $CLSK^*$ and a parameter list params.*
- CL_Ext_Partial_Pri_Key*, the partial private key issuance algorithm, is a deterministic algorithm that takes as input a user identity id, a parameter list params, and a master key $CLSK^*$. It returns the user id's partial private key CLD_{id}.*
- CL_Set_Sec_Val*, the secret value setup algorithm, is a probabilistic algorithm that takes as input a parameter list params and a user identity id. It returns the user id's secret value CLS_{id}.*
- CL_Set_Pri_Key*, the signing key generation algorithm, is a deterministic algorithm that takes as input a parameter list params, the user id's partial*

private key CLD_{id}, and the user id's secret value CLS_{id}. It returns the user id's private signing key $CLSK_{id}$.

- $\mathsf{CL\_Set\_Pub\_Key}$*, the verification key generation algorithm, is a deterministic algorithm that takes as input a parameter list params, a user identity id, and the user id's secret value CLS_{id}. It returns the user id's public verification key $CLPK_{id}$.*
- $\mathsf{CL\_Sign}$*, the signing algorithm, is a probabilistic algorithm that takes as input a message M, a user identity id, a parameter list params, and the user id's private signing key $CLSK_{id}$.* $\mathsf{CL\_Sign}_{params}^{CLSK_{id}}(M)$ *returns a signature α.*
- $\mathsf{CL\_Vrfy}$*, the verification algorithm, is a deterministic algorithm that takes as input a parameter list params, the public verification key $CLPK_{id}$, a message M, a user identity id, and a signature α.* $\mathsf{CL\_Vrfy}_{params}^{CLPK_{id}}(M, id, \alpha)$ *returns a bit b, where $b = 1$ means that the signature is accepted.*

In a certificateless signature scheme, $\mathsf{CL\_Gen}$ and $\mathsf{CL\_Ext\_Partial\_Pri\_Key}$ are performed by a KGC. A partial private key CLD_{id} is given to a user id by the KGC through a secure channel. Since $\mathsf{CL\_Set\_Sec\_Val}$, $\mathsf{CL\_Set\_Pri\_Key}$, and $\mathsf{CL\_Set\_Pub\_Key}$ are executed by a user, the key escrow of the user's private key is not inherent in a certificateless signature scheme. We require that all signatures output by $\mathsf{CL\_Sign}_{params}^{CLSK_{id}}(\cdot)$ are accepted as valid by $\mathsf{CL\_Vrfy}_{params}^{CLPK_{id}}(\cdot, id, \cdot)$.

For security analysis, we extend the model of an identity-based signature scheme to allow an adversary to extract partial private keys, or private keys, or both, for identities of his choice. We must also consider the ability of the adversary to replace the public key of any entity with a value of his choice, because there is no certificate in a certificateless signature scheme. Five oracles can be accessed by the adversary. The first is a partial private key exposure oracle $\mathsf{O}_{\mathsf{Exp\_Partial}}^{\mathsf{CL}}(\cdot)$ that returns CLD_{id} on input a user identity id. The second is a private key exposure oracle $\mathsf{O}_{\mathsf{Exp\_Pri}}^{\mathsf{CL}}(\cdot)$ that returns $CLSK_{id}$ on input a user identity id if id's public key has not been replaced. The third is a public key broadcast oracle $\mathsf{O}_{\mathsf{Bro\_Pub}}^{\mathsf{CL}}(\cdot)$ that returns $CLPK_{id}$ on input a user identity id. The fourth is a public key replacement oracle $\mathsf{O}_{\mathsf{Rep\_Pub}}^{\mathsf{CL}}(\cdot, \cdot)$ that replaces the public key $CLPK_{id}$ for a user id with $CLPK'_{id}$ on input (id, $CLPK'_{id}$). The fifth is a signing oracle $\mathsf{O}_{\mathsf{Sign}}^{\mathsf{CL}}(\cdot, \cdot)$ that returns $\mathsf{CL\_Sign}_{params}^{CLSK_{id}}(M)$ on input (M, id).

The security of a certificateless signature scheme is against two different types of adversaries. The Type I adversary A_I has no access to the master key, but may replace public keys, extract partial private and private keys, and make signing queries. When A_I has replaced the public key of a user id and requests the user id's signature, we accept that the signing oracle's answer will be incorrect. We adopt this behavior of the signing oracle because we will construct certificateless signature schemes based on general primitives and do without any additional assumptions, such as the random oracle model [5]. However, we assume that the signing oracle's answer is correct, if A_I additionally submits the replaced public key ($CLPK'_{id}$) and the corresponding secret information (CLS'_{id} or $CLSK'_{id}$) to the signing oracle.[1] The Type II adversary A_{II} equipped with the master

[1] This model was used in [9,13].

key models a dishonest KGC and can generate partial private keys by himself. However, A_{II} is not allowed to replace public keys.

Definition 4. *Let Π_{CL} be a certificateless signature scheme. For any adversary A, we may perform the following experiment:*

$$(CLSK^*, params) \leftarrow \mathsf{CL\_Gen}(1^k);$$
$$(M, id, \alpha) \leftarrow A^{\mathsf{O}_1(\cdot),\mathsf{O}_2(\cdot),\mathsf{O}^{\mathsf{CL}}_{\mathsf{Exp\_Pri}},\mathsf{O}^{\mathsf{CL}}_{\mathsf{Bro\_Pub}},\mathsf{O}^{\mathsf{CL}}_{\mathsf{Sign}}(\cdot,\cdot)}(params, h).$$

where $h = \perp$, $\mathsf{O}_1(\cdot) = \mathsf{O}^{\mathsf{CL}}_{\mathsf{Exp\_Partial}}(\cdot)$, $\mathsf{O}_2(\cdot) = \mathsf{O}^{\mathsf{CL}}_{\mathsf{Rep\_Pub}}(\cdot,\cdot)$ for A_I and $h = CLSK^*$, $\mathsf{O}_1(\cdot) = \mathsf{O}_2(\cdot) = \perp$ for A_{II}. We say that A succeeds if $\mathsf{CL\_Vrfy}^{CLPK_{id}}_{params}(M, id, \alpha) = 1$ and A has followed the adversarial constraints. Denote the probability of A's success by $\mathsf{Succ}_{A,\Pi_{CL}}(k)$. If for any PPT A, the success probability $\mathsf{Succ}_{A,\Pi_{CL}}(k)$ is negligible, we say that Π_{CL} is secure; that is existentially unforgeable against chosen-message attacks.

If (M, id_{ch}, α) is the output of the adversary A_I, the identity id_{ch} cannot be submitted to the partial private key exposure oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Exp\_Partial}}(\cdot)$; $CLD_{id_{ch}}$ is securely given to the user id_{ch} by definition and can be deleted after generating the private signing key. However, A_I is allowed to replace the public key of id_{ch}. The exposure of $CLD_{id_{ch}}$ can be treated by the Type II adversary A_{II} who is equipped with the master key $CLSK^*$. For other restrictions on the two types of adversaries and security notions, refer to [1].

3 Generic Construction of Certificateless Signature

3.1 Generic Secure Construction

We provide the generic secure construction of certificateless signature based on public key signature and identity-based signature. Let Π_{PK} = (PK_Gen, PK_Sign, PK_Vrfy) be a public key signature scheme that is secure in the sense of [11] and Π_{IB} = (IB_Gen, IB_Ext, IB_Sign, IB_Vrfy) be a secure identity-based signature scheme. To avoid the key escrow problem of Π_{IB}, we will use the idea of sequential double signing. A secure certificateless signature scheme Ψ_{CL} = (CL_Gen, CL_Ext_Partial_Pri_Key, CL_Set_Sec_Val, CL_Set_Pri_Key, CL_Set_Pub_Key, CL_Sign, CL_Vrfy) can be constructed as in Table 1.

3.2 Security Analysis

The security of Ψ_{CL} in Table 1 can be proved by the security of Π_{PK} and Π_{IB}. If there is a Type I attacker A_I who can break Ψ_{CL}, we can construct the adversary A' against Π_{IB}. If there is a Type II attacker A_{II}, we can construct the adversary A'' against Π_{PK}.

Theorem 1. *Ψ_{CL} is a secure certificateless signature scheme if Π_{PK} and Π_{IB} are existentially unforgeable against chosen-message attacks.*

Table 1. Generic construction of certificateless signature.

$\mathsf{CL\_Gen}(1^k)$
$(IBSK^*, params) \leftarrow \mathsf{IB\_Gen}(1^k)$;
$CLSK^* \leftarrow IBSK^*$;
Return $(CLSK^*, params)$

$\mathsf{CL\_Ext\_Partial\_Pri\_Key}(id, params, CLSK^*)$
$IBSK_{id} \leftarrow \mathsf{IB\_Ext}(id, params, CLSK^*)$;
$CLD_{id} \leftarrow IBSK_{id}$;
Return CLD_{id}

$\mathsf{CL\_Set\_Pri\_Key}(params, CLD_{id}, CLS_{id})$
Parse CLS_{id} as (pk_{id}, sk_{id});
$CLSK_{id} \leftarrow (CLD_{id}, sk_{id})$;
Return $CLSK_{id}$

$\mathsf{CL\_Sign}(M, id, params, CLSK_{id})$
Parse $CLSK_{id}$ as (CLD_{id}, sk_{id});
$\alpha \leftarrow \mathsf{PK\_Sign}_{sk_{id}}(M)$;
$\beta \leftarrow \mathsf{IB\_Sign}^{CLSK_{id}}_{params}(\alpha, id)$;
Return $\langle \alpha, \beta \rangle$

$\mathsf{CL\_Set\_Sec\_Val}(params, id)$
$(pk_{id}, sk_{id}) \leftarrow \mathsf{PK\_Gen}(1^k)$;
$CLS_{id} \leftarrow (pk_{id}, sk_{id})$;
Return CLS_{id}

$\mathsf{CL\_Set\_Pub\_Key}(params, id, CLS_{id})$
Parse CLS_{id} as (pk_{id}, sk_{id});
$CLPK_{id} \leftarrow pk_{id}$;
Return $CLPK_{id}$

$\mathsf{CL\_Vrfy}(params, CLPK_{id}, M, id, \langle \alpha, \beta \rangle)$
$b_1 \leftarrow \mathsf{IB\_Vrfy}_{params}(\alpha, id, \beta)$;
$b_2 \leftarrow \mathsf{PK\_Vrfy}_{CLPK_{id}}(M, \alpha)$;
$b \leftarrow b_1 \& b_2$; // bitwise AND
Return b

Proof. (Sketch) Let A_I be a Type I attacker who can break Ψ_{CL}. Suppose that A_I has a success probability ϵ and runs in time t. We show how to construct from A_I an adversary A' against Π_{IB}. At the beginning, A' is given by a Π_{IB} challenger a parameter list $params$ and two oracles: the key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ and the signing oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)$. Let $\Pi_{PK} = (\mathsf{PK\_Gen}, \mathsf{PK\_Sign}, \mathsf{PK\_Vrfy})$ be a secure public key signature scheme that is chosen by A'. Then Ψ_{CL} is well-defined from Π_{IB} and Π_{PK}. To run A_I, A' simulates the $\mathsf{CL\_Gen}(1^k)$ by supplying A_I with $params$. A' keeps a list $L = \{(id, IBSK_{id}, pk_{id}, sk_{id})\}$ where $IBSK_{id}$ is the output of $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ and (pk_{id}, sk_{id}) is an output of $\mathsf{PK\_Gen}(1^k)$. The list L need not be made in advance and is computed according to the A_I's queries. A' responds to A_I's oracle queries as follows.

- Partial private key exposure oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Exp\_Partial}}(\cdot)$ queries: Suppose that the request is on a user identity id.
  1. When the list L contains $(id, IBSK_{id}, pk_{id}, sk_{id})$, A' checks to determine whether $IBSK_{id} = \perp$ or not. If $IBSK_{id} \neq \perp$, A' returns $CLD_{id} = IBSK_{id}$ to A_I. If $IBSK_{id} = \perp$, A' sends id to the Π_{IB} key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ and obtains $IBSK_{id}$. A' writes $IBSK_{id}$ in the list L and returns $CLD_{id} = IBSK_{id}$ to A_I.
  2. When the list L does not contain $(id, IBSK_{id}, pk_{id}, sk_{id})$, A' sends id to the Π_{IB} key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ and obtains $IBSK_{id}$. A' sets $pk_{id} = sk_{id} = \perp$. The element $(id, IBSK_{id}, pk_{id}, sk_{id})$ is added to the list L. A' returns $CLD_{id} = IBSK_{id}$ to A_I.

- Private key exposure oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Exp\_Pri}}(\cdot)$ queries: Suppose that the request is on a user identity id.
  1. When the list L contains (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' checks $IBSK_{id} = \perp$ and $pk_{id} = \perp$. If $IBSK_{id} = \perp$, A' sets $IBSK_{id} = \mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(id)$. If $pk_{id} = \perp$, A' runs $\mathsf{PK\_Gen}(1^k)$ to obtain (pk_{id}, sk_{id}) and saves these values in the list L. A' returns $CLSK_{id} = (IBSK_{id}, sk_{id})$.
  2. When the list L does not contain (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' adds the element (id, $IBSK_{id}$, pk_{id}, sk_{id}) to the list L by sending id to the Π_{ID} key exposure oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Exp}}(\cdot)$ and running $\mathsf{PK\_Gen}(1^k)$. A' returns $CLSK_{id} = (IBSK_{id}, sk_{id})$.
- Public key broadcast oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Bro\_Pub}}(\cdot)$ queries: Suppose that the request is on a user identity id.
  1. When the list L contains (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' checks to determine whether $pk_{id} = \perp$ or not. If $pk_{id} \neq \perp$, A' returns $CLPK_{id} = pk_{id}$. Otherwise, A' runs $\mathsf{PK\_Gen}(1^k)$ and obtains (pk_{id}, sk_{id}). A' saves (pk_{id}, sk_{id}) in the list L and returns $CLPK_{id} = pk_{id}$.
  2. When the list L does not contain (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' sets $IBSK_{id} = \perp$ and runs $\mathsf{PK\_Gen}(1^k)$ to obtain (pk_{id}, sk_{id}). A' adds (id, $IBSK_{id}$, pk_{id}, sk_{id}) to the list L and returns $CLPK_{id} = pk_{id}$.
- Public key replacement oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Rep\_Pub}}(\cdot,\cdot)$ queries: Suppose that A_I asks with an input (id, $CLPK_{id}$).
  1. When the list L contains (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' sets $pk_{id} = CLPK_{id}$ and $sk_{id} = \perp$.
  2. When the list L does not contain (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' sets $IBSK_{id} = \perp$, $pk_{id} = CLPK_{id}$, $sk_{id} = \perp$ and adds the element (id, $IBSK_{id}$, pk_{id}, sk_{id}) to the list L.
- Signing oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Sign}}(\cdot,\cdot)$ queries: Suppose that A_I asks with an input (M, id).
  1. When the list L contains (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' checks to determine whether $pk_{id} = \perp$ or not. If $pk_{id} = \perp$, A' runs $\mathsf{PK\_Gen}(1^k)$ to get (pk_{id}, sk_{id}) and saves these values in the list L.
     a) A' checks whether $sk_{id} = \perp$ or not. If $sk_{id} = \perp$, i.e., the public key for the user id has been replaced by A_I, A' returns a random signature $\langle \alpha, \beta \rangle$.
     b) A' computes $\alpha = \mathsf{PK\_Sign}_{sk_{id}}(M)$. A' checks whether $IBSK_{id} = \perp$ or not. If $IBSK_{id} = \perp$, A' sends (α, id) to the Π_{IB} signing oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)$. Let β be the output of $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\alpha, id)$. A' returns $\langle \alpha, \beta \rangle$ to A_I. If $IBSK_{id} \neq \perp$, A' computes $\beta = \mathsf{IB\_Sign}^{IBSK_{id}}_{params}(\alpha, id)$. A' returns $\langle \alpha, \beta \rangle$ to A_I.
  2. When the list L does not contain (id, $IBSK_{id}$, pk_{id}, sk_{id}), A' sets $IBSK_{id} = \perp$ and runs $\mathsf{PK\_Gen}(1^k)$ to obtain (pk_{id}, sk_{id}). A' adds (id, $IBSK_{id}$, pk_{id}, sk_{id}) to the list L and computes $\alpha = \mathsf{PK\_Sign}_{sk_{id}}(M)$. A' sends (α, id) to the Π_{IB} signing oracle $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\cdot,\cdot)$. Let β be the output of $\mathsf{O}^{\mathsf{IB}}_{\mathsf{Sign}}(\alpha, id)$. A' returns $\langle \alpha, \beta \rangle$ to A_I.

When the Type I attacker A_I outputs $(M, id, \langle \alpha, \beta \rangle)$, A' outputs (α, id, β) to the Π_{IB} challenger. Since the A_I's view is identical to its view in the real attack, the success probability of A' is also ϵ and A' runs in time $O(time(t))$.

Let A_{II} be a Type II attacker who can break Ψ_{CL}. Suppose that A_{II} has a success probability ϵ, runs in time t, and makes queries on l users, i.e., (id_1, id_2, $\cdots$, id_l). We show how to construct from A_{II} an adversary A'' against Π_{PK}. At the beginning, A'' is given by a Π_{PK} challenger a public key pk and the signing oracle $\mathsf{O}^{\mathsf{PK}}_{\mathsf{Sign}}(\cdot)$. Let $\Pi_{IB} = (\mathsf{IB\_Gen}, \mathsf{IB\_Ext}, \mathsf{IB\_Sign}, \mathsf{IB\_Vrfy})$ be a secure identity-based signature scheme that is chosen by A''. Then Ψ_{CL} is well-defined from Π_{PK} and Π_{IB}. To simulate $\mathsf{CL\_Gen}(1^k)$, A'' runs $\mathsf{IB\_Gen}(1^k)$ to obtain ($IBSK^*$, $params$) and sets $CLSK^* = IBSK^*$. A'' gives ($params$, $CLSK^*$) to A_{II} since A_{II} has access to the master key. As before, A'' keeps a list $L = \{(id, IBSK_{id}, pk_{id}, sk_{id})\}$ where $IBSK_{id}$ is the output of $\mathsf{IB\_Ext}(id, params, IBSK^*)$ and (pk_{id}, sk_{id}) is an output of $\mathsf{PK\_Gen}(1^k)$. A'' chooses a random index $j \in \{1, \cdots, l\}$ and sets $IBSK_{id_j} = \mathsf{IB\_Ext}(id_j, params, IBSK^*)$, $pk_{id_j} = pk$, $sk_{id_j} = \perp$. The element $(id_j, IBSK_{id_j}, pk_{id_j}, sk_{id_j})$ is added to the list L. The remainder of the list L is computed according to the A_{II}'s queries. A'' responds to A_{II}'s oracle queries as follows.

- Private key exposure oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Exp\_Pri}}(\cdot)$ queries: Suppose that the request is on a user identity id. If $id = id_j$, A'' aborts.
  1. When the list L contains $(id, IBSK_{id}, pk_{id}, sk_{id})$, A'' checks $IBSK_{id} = \perp$ and $pk_{id} = \perp$. If $IBSK_{id} = \perp$, A'' sets $IBSK_{id} = \mathsf{IB\_Ext}(id, params, IBSK^*)$. If $pk_{id} = \perp$, A'' runs $\mathsf{PK\_Gen}(1^k)$ to obtain (pk_{id}, sk_{id}) and saves these values in the list L. A'' returns $CLSK_{id} = (IBSK_{id}, sk_{id})$.
  2. When the list L does not contain $(id, IBSK_{id}, pk_{id}, sk_{id})$, A'' adds the element $(id, IBSK_{id}, pk_{id}, sk_{id})$ to the list L by setting $IBSK_{id} = \mathsf{IB\_Ext}(id, params, IBSK^*)$ and running $\mathsf{PK\_Gen}(1^k)$. A'' returns $CLSK_{id} = (IBSK_{id}, sk_{id})$.
- Public key broadcast oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Bro\_Pub}}(\cdot)$ queries: Suppose that the request is on a user identity id.
  1. When the list L contains $(id, IBSK_{id}, pk_{id}, sk_{id})$, A'' checks to determine whether $pk_{id} = \perp$ or not. If $pk_{id} \neq \perp$, A'' returns $CLPK_{id} = pk_{id}$. Otherwise, A'' runs $\mathsf{PK\_Gen}(1^k)$ and obtains (pk_{id}, sk_{id}). A'' saves (pk_{id}, sk_{id}) in the list L and returns $CLPK_{id} = pk_{id}$.
  2. When the list L does not contain $(id, IBSK_{id}, pk_{id}, sk_{id})$, A'' sets $IBSK_{id} = \perp$ and runs $\mathsf{PK\_Gen}(1^k)$ to obtain (pk_{id}, sk_{id}). A'' adds $(id, IBSK_{id}, pk_{id}, sk_{id})$ to the list L and returns $CLPK_{id} = pk_{id}$.
- Signing oracle $\mathsf{O}^{\mathsf{CL}}_{\mathsf{Sign}}(\cdot, \cdot)$ queries: Suppose that A_{II} asks with an input (M, id).
  1. When the list L contains $(id, IBSK_{id}, pk_{id}, sk_{id})$, A'' checks to determine whether $IBSK_{id} = \perp$ and $pk_{id} = \perp$. If $IBSK_{id} = \perp$, A'' sets $IBSK_{id} = \mathsf{IB\_Ext}(id, params, IBSK^*)$. If $pk_{id} = \perp$, A'' runs $\mathsf{PK\_Gen}(1^k)$ to get (pk_{id}, sk_{id}) and saves these values in the list L. Now, A'' checks to see whether $sk_{id} = \perp$ or not.

a) If $sk_{id} = \perp$, A'' sends M to the Π_{PK} signing oracle $\mathsf{O}^{\mathsf{PK}}_{\mathsf{Sign}}(\cdot)$. Let α be the output of $\mathsf{O}^{\mathsf{PK}}_{\mathsf{Sign}}(M)$. A'' computes $\beta = \mathsf{IB\_Sign}^{IBSK_{id}}_{params}(\alpha)$. A'' returns $\langle \alpha, \beta \rangle$ to A_{II}.
b) Otherwise, A'' computes $\alpha = \mathsf{PK\_Sign}_{sk_{id}}(M)$ and $\beta = \mathsf{IB\_Sign}^{IBSK_{id}}_{params}(\alpha, id)$. A'' returns $\langle \alpha, \beta \rangle$ to A_{II}.

2. When the list L does not contain $(id, IBSK_{id}, pk_{id}, sk_{id})$, A'' sets $IBSK_{id} = \mathsf{IB\_Ext}(id, params, IBSK^*)$ and runs $\mathsf{PK\_Gen}(1^k)$ to obtain (pk_{id}, sk_{id}). A'' adds $(id, IBSK_{id}, pk_{id}, sk_{id})$ to the list L. A'' computes $\alpha = \mathsf{PK\_Sign}_{sk_{id}}(M)$ and $\beta = \mathsf{IB\_Sign}^{IBSK_{id}}_{params}(\alpha, id)$. A'' returns $\langle \alpha, \beta \rangle$ to A_{II}.

When the Type II attacker A_{II} outputs $(M, id, \langle \alpha, \beta \rangle)$, A'' outputs (M, α) to the Π_{PK} challenger. If A'' does not abort during simulation, the A_{II}'s view is identical to its view in the real attack. Since the index j is chosen randomly, the probability that A'' does not abort during simulation is $1/l$. Hence, the success probability of A'' is at least ϵ/l and A'' runs in time $O(time(t))$. Q.E.D. □

3.3 Extended Construction

The public key cryptosystem can be classified into three trust levels according to the trust assumption of the TTP (Trusted Third Party) [10]:

- At level 1, the TTP knows (or can easily compute) the users' private keys and therefore can impersonate any user at any time without being detected.
- At level 2, the TTP does not know (or cannot easily compute) the users' private keys. Nevertheless, the TTP can still impersonate a user by generating a false public key (or a false certificate).
- At level 3, the TTP does not know (or cannot easily compute) the users' private keys. Moreover, it can be proved that the TTP generates false public keys of users if it does so.

In a traditional public key signature scheme, if the CA forges certificates, the CA can be identified as having misbehaved through the existence of two valid certificates for the same identity. However, a false public key can be created by the KGC without being detected in the certificateless signature scheme Ψ_{CL}, since a new public key can be created by both the legitimate user and the KGC. While the traditional public key signature achieves trust level 3, the certificateless signature reaches only trust level 2.

At this point, our question is how the certificateless signature can achieve trust level 3. In other words, can we prevent the dishonest KGC from issuing two valid partial private keys for one user? We can use a simple technique to bind a user identity id and his public key $CLPK_{id}$, that was employed in [1].[2]

[2] This binding technique cannot raise a certificateless encryption scheme to trust level 3 as opposed to the claim in [1]. In a signature scheme, the existence of two different, working public keys for a single user can be proved by the corresponding two valid signatures; a digital signature is universally verifiable with the corresponding

Table 2. Generic construction of certificateless signature of trust level 3.

$\mathsf{CL\_Gen}(1^k)$
$(IBSK^*, params) \leftarrow \mathsf{IB\_Gen}(1^k)$;
$CLSK^* \leftarrow IBSK^*$;
Return $(CLSK^*, params)$

$\mathsf{CL\_Set\_Sec\_Val}(params, id)$
$(pk_{id}, sk_{id}) \leftarrow \mathsf{PK\_Gen}(1^k)$;
$CLS_{id} \leftarrow (pk_{id}, sk_{id})$;
Return CLS_{id}

$\mathsf{CL\_Set\_Pub\_Key}(params, id, CLS_{id})$
Parse CLS_{id} as (pk_{id}, sk_{id});
$CLPK_{id} \leftarrow pk_{id}$;
Return $CLPK_{id}$

$\mathsf{CL\_Ext\_Partial\_Pri\_Key}(id, CLPK_{id}, params, CLSK^*)$
$IBSK_{id} \leftarrow \mathsf{IB\_Ext}(id||CLPK_{id}, params, CLSK^*)$;
$CLD_{id} \leftarrow IBSK_{id}$;
Return CLD_{id}

$\mathsf{CL\_Set\_Pri\_Key}(params, CLD_{id}, CLS_{id})$
Parse CLS_{id} as (pk_{id}, sk_{id});
$CLSK_{id} \leftarrow (CLD_{id}, sk_{id})$;
Return $CLSK_{id}$

$\mathsf{CL\_Sign}(M, id, params, CLSK_{id})$
Parse $CLSK_{id}$ as (CLD_{id}, sk_{id});
$\alpha \leftarrow \mathsf{PK\_Sign}_{sk_{id}}(M)$;
$\beta \leftarrow \mathsf{IB\_Sign}^{CLSK_{id}}_{params}(\alpha, id||CLPK_{id})$;
Return $\langle \alpha, \beta \rangle$

$\mathsf{CL\_Vrfy}(params, CLPK_{id}, M, id, \langle \alpha, \beta \rangle)$
$b_1 \leftarrow \mathsf{IB\_Vrfy}_{params}(\alpha, id||CLPK_{id}, \beta)$;
$b_2 \leftarrow \mathsf{PK\_Vrfy}_{CLPK_{id}}(M, \alpha)$;
$b \leftarrow b_1 \& b_2$;
Return b

public key. However, two ciphertexts do not guarantee the validity of the two public keys in an encryption scheme; without the corresponding private keys, we cannot check whether the ciphertexts are correct or not. To guarantee the existence of two different, working public encryption keys for a single user, the private keys or the partial private keys must be presented (or provided in a zero knowledge manner) in a certificateless encryption scheme. Note that we do not expect the dishonest KGC to reveal this information.

This technique reduces the degree of trust that users need to have in the KGC and raises the certificateless signature to trust level 3. A minor drawback of this technique is that the input of CL_Ext_Partial_Pri_Key includes $CLPK_{id}$ and hence, CL_Set_Pub_Key should be executed before the KGC runs CL_Ext_Partial_Pri_Key.

Let Π_{PK} = (PK_Gen, PK_Sign, PK_Vrfy) be a secure public key signature scheme and Π_{IB} = (IB_Gen, IB_Ext, IB_Sign, IB_Vrfy) be a secure identity-based signature scheme. The construction of a trust level 3 certificateless signature scheme Λ_{CL} = (CL_Gen, CL_Ext_Partial_Pri_Key, CL_Set_Sec_Val, CL_Set_Pri_Key, CL_Set_Pub_Key, CL_Sign, CL_Vrfy) is shown in Table 2. Note that the existence of two different, working public keys for a single user will identity the KGC as having misbehaved in issuing both corresponding partial private keys, since the underlying Π_{IB} is a secure identity-based signature scheme and only the KGC knows the master key $IBSK^*$.

4 Concluding Remarks

A certificateless signature is a new digital signature paradigm that simplifies the public key infrastructure. A certificateless signature retains the efficiency of Shamir's identity-based signature while it does not suffer from the inherent private key escrow problem. We provided a generic secure construction of a certificateless signature in a more general manner. Moreover, the extended construction achieves trust level 3, the same level as is enjoyed in a traditional signature scheme.

Acknowledgment. The authors would like to thank Yong Ho Hwang for his help in preparing the final version of this paper.

References

1. S. S. Al-Riyami and K. G. Peterson, "Certificateless public key cryptography," Asiacrypt 2003, LNCS Vol. 2894, pp. 452-474, 2003.
2. M. Bellare, A. Desai, D. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption: analysis of the DES modes of operation," FOCS 1997, IEEE, 1997.
3. D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," Crypto 2001, LNCS Vol. 2139, pp. 213-229, 2001.
4. D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003.
5. M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," 1st ACM Conf. on Computer and Communications Security, pp. 62-73, 1993.
6. J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," PKC 2003, LNCS Vol. 2567, pp. 18-30, 2003.
7. L. C. Guillou and J. J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," Eurocrypt 1988, LNCS Vol. 330, pp. 123-128, 1988.

8. L. C. Guillou and J. J. Quisquater "A "paradoxical" identity-based signature scheme resulting from zero-knowledge," Crypto 1988, LNCS Vol. 403, pp. 216-231, 1988.
9. C. Gentry, "Certificate-based encryption and the certificate revocation problem," Eurocrypt 2003, LNCS Vol. 2656, pp. 272-293, 2003.
10. M. Girault, "Self-certified public keys," Eurocrypt 1991, LNCS Vol. 547, pp. 490-497, 1992.
11. S. Goldwasswer, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," SIAM J. Computing, Vol 7, No 2, pp. 281-308, 1988.
12. F. Hess, "Efficient identity based signature schemes based on pairings," SAC 2002, LNCS Vol. 2595, pp. 310-324, 2003.
13. G. Kang, J. H. Park, and S. G. Hahn "A certificate-based signature scheme," CT-RSA 2004, LNCS Vol. 2964, pp. 99-111, 2004.
14. K. G. Paterson, "ID-based signatures from pairings on elliptic curves," Electronics Letters Vol. 38 (18), pp. 1025-1026, 2002.
15. A. Shamir, "Identity-based cryptosystems and signature schemes," Crypto 1984, LNCS Vol. 196, pp. 47-53, 1984.
16. D. H. Yum and P. J. Lee, "Generic construction of certificateless encryption," The 2004 International Conference on Computational Science and its Applications, Assisi (Perugia, Italy), May 14 - May 17, 2004.