Journal of Universal Computer Science, vol. 15, no. 8 (2009), 1659-1684
submitted: 25/11/08, accepted: 24/4/09, appeared: 28/4/09 © J.UCS

Certificate-based Signatures Revisited[1]

Wei Wu
(Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
ww986@uow.edu.au)

Yi Mu
(Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
ymu@uow.edu.au)

Willy Susilo
(Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
wsusilo@uow.edu.au)

Xinyi Huang
(Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
xh068@uow.edu.au)

Abstract: Certificate-based encryption was introduced in Eurocrypt'03 to solve the certificate management problem in public key encryption. Recently, this idea was extended to certificate-based signatures. Several new schemes and security models of certificate-based signatures have been proposed. In this paper, we first take a closer look at the certificate-based signature by comparing it with digital signatures in other popular public key systems. We introduce a new security model of certificate-based signature, which defines several new types of adversaries against certificate-based signatures, along with the security model of certificate-based signatures against them. The new model is clearer and more elaborated compared with other existing ones. We then investigate the relationship between certificate-based signatures and certificateless signatures, and propose a generic construction of certificate-based signatures. We prove that the generic construction is secure (in the random oracle model) against all types of adversaries defined in this paper, assuming the underlying certificateless signatures satisfying certain security notions. Based on our generic construction, we are able to construct new certificate-based signature schemes, which are more efficient in comparison with other schemes with similar security levels.
Key Words: certificate-based signatures, certificateless signatures, security model, generic construction, concrete scheme

---
[1] This is the full version of our paper presented at WISA 2008 [Wu et al. 2008].

Category: K.6.5, E.3

1 Introduction

In a public-key cryptosystem, each user has a pair of keys: public key and private key. The public key is always published and publicly accessible, while the private key is kept secret by the owner. The central problem in a public key system is to prove that a public key is genuine and authentic, and has not been tampered with or replaced by a malicious third party. The usual approach to ensure the authenticity of a public key is to use a certificate. A (digital) certificate is a signature of a trusted certificate authority (CA) that binds together the identity of an entity A, its public key PK and other information. This kind of systems is referred as public key infrastructure (PKI). The PKI is generally considered to be costly to use and manage.

Shamir [Shamir 1985] introduced the concept of identity-based public key cryptography (or, ID-PKC for short), where the original motivation is to ease certificate management in the e-mail system. A user's public key in ID-PKC is some unique information about the identity of the user (e.g., email address). The private key in ID-PKC is generated by a trusted third party called Private Key Generator (PKG) who holds a master key. Thus, key escrow is an inherent problem in this kind of ID-PKC (e.g., [Shamir 1985, Boneh and Franklin 2001]), as the PKG knows any user's private key. The key escrow problem could be partially solved by the introduction of multiple PKGs and the use of threshold techniques, which requires extra communications and infrastructures.

Al-Riyami and Paterson proposed a new paradigm called certificateless public key cryptography [Al-Riyami and Paterson 2003] (or, CL-PKC for short), where the original motivation is to find a public key system that does not require the use of certificates and does not have the key escrow problem. Each entity in CL-PKC holds two secrets: a secret value and a partial private key. The secret value SV is generated by the entity itself, while a third party Key Generating Center (KGC), holding a master key, generates the partial private key PPK from the user's identity information [2]. The entity's actual private key is the output of some function with the input SV and PPK. This way, KGC does not know the actual private key and the key escrow problem is eliminated. The entity can use the actual private key to generate the public key, which is no longer only computed from the identity. This makes the certificateless system non-identity-based. The entity's public key could be available to other entities

[2] In Section 5.1 of [Al-Riyami and Paterson 2003], the authors sketched an alternative partial private key generation technique. In this paper, when we mention a cryptographic protocol in CL-PKC, we mean it is a protocol with the classic private key generation technique used in Section 4.1 of [Al-Riyami and Paterson 2003], which has been adopted by most researchers in CL-PKC.

by transmitting it along with messages (for example, in a signing application) or by placing it in a public directory (this would be more appropriate for an encryption setting). However, there is no certificate to ensure the authenticity of the entity's public key in CL-PKC. Therefore, it is necessary to assume that an adversary is able to replace the entity's public key with a false key of its choice, which is also known as *key replacement attack* [Huang et al. 2005]. One assumption in CL-PKC is that KGC never mounts the key replacement attack. In the traditional PKI, however, one does not need to make the same assumption on the third party CA, who if replaces the entity's public key with a false key of its choice, can be implicitly proved due to the existence of the certificate for that false key.

In Eurocrypt 2003, Gentry [Gentry 2003] introduced the notion of certificate-based encryption. As in the traditional PKI, each client generates its own public/private key pair and requests a certificate from the CA. The difference is that, a certificate in the certificate-based cryptography, or more generally, a signature from the third party acts not only as a certificate (as in the traditional PKI) but also as a decryption key (as in ID-PKC and CL-PKC). The sender can encrypt a message without obtaining explicit information other than the recipient's public key and the parameters of CA. To decrypt a message, a keyholder needs both its secret key and an up-to-date certificate from its CA (or a signature from an authority). Therefore, CA does not need to make the certificate status information available among the whole system, and only needs to contact the certificate holder for revocation and update. As the sender is not required to verify the certificate of the recipient's public key, the sender could be duped to encrypt messages with an uncertified public key. This could be due to the recipient has not yet had his/her public key certified, or the encryption key that the sender holds is not the recipient's authentic public key. In this sense, certificate-based encryption works in a similar way to certificateless encryption, but the difference is that certificates do exist in certificate-based encryption.

Certificate-based cryptography was introduced to solve the certificate management problem in the traditional PKI, but only in the scenario of encryption. The notion of certificate-based encryption was extended to certificate-based signature in [Kang et al. 2004, Li et al. 2007]. However, as mentioned in [Gentry 2003], if we only consider signing and verification signatures in a public key cryptosystem, then the certificate management problem is not as challenging as in the scenario of encryption and decryption. For example, the signer can send its public key and the proof of certificate status to the verifier simultaneously with its signature, thus the verifier can obtain the certificate without referring to a public directory or issuing a third-party query to CA. This, however, will require more bandwidth for signature transmitting. Public key cryptosystems like ID-PKC [Shamir 1985, Boneh and Franklin 2001] and CL-

PKC [Al-Riyami and Paterson 2003] can eliminate the certificate management problem as one can directly use the entity A's public key to verify signatures, without checking the certificate of A's public key. However, this is achieved at the cost of assuming certain trust on the authority, who is able to impersonate any user in an undetectable way. In certificate-based cryptosystem, certificate management problem can be eased in a different way. To generate valid certificate-based signatures of a user with the identity information ID and the public key PK, one needs two pieces of secret information, namely a valid certificate of (ID, PK) and the secret key of PK. In other words, a valid certificate-based signature ensures the existence of a valid certificate. Thus, the signer does not need to send the certificate along with the message and the signature. This is achieved at the cost of more computational operations in signature verification, which implies the verification of the certificate. If one replaces PK with PK' and generates a valid signature under ID and PK', he/she must have a certificate of (ID, PK'). This can prove that the third party CA is dishonest, as there is only one party with the ability to generate certificates. Therefore, the third party in certificate-based signatures has the Trust Level 3 in the definition in [Girault 1991], which is similar as CA in the traditional PKI and a few constructions of identity-based signatures [Bellare et al. 2004, Galindo et al. 2006]. To summarize, (1) The authority in certificate-based signatures and traditional PKI-based signatures is at Trust Level 3 in the definition given in [Girault 1991], which is higher than the authority in the ID-PKC and the CL-PKC, and (2) To ease the problem of certificate management, certificate-based signatures consume (in general) less bandwidth in signature transmitting but might require more computational cost than traditional-PKI-based signatures.

1.1 Related Work

Kang, Park and Hahn proposed the notion and the first construction of certificate-based signatures in [Kang et al. 2004], by extending the idea of certificate-based encryption in [Gentry 2003]. That is, to generate a valid signature under the public key PK, the entity needs to know both the corresponding private key SK and the up-to-date certificate of PK. To verify a claimed signature, one only needs the signer's public key and the parameter of CA (particularly, no need to check the certificate of that public key). As the verifier is not required to check the certificate about a claimed public key, key replacement attacks also exist in certificate-based cryptography. Key replacement attacks in certificate-based signatures were first addressed in [Kang et al. 2004] and formally defined in [Li et al. 2007]. As introduced in [Li et al. 2007], adversaries in certificate-based signatures can be divided into two types: **CB-**$\mathcal{A}_I$ and **CB-**$\mathcal{A}_{II}$. **CB-**$\mathcal{A}_I$ can replace any entity's public key PK with a new public key PK' chosen by itself, and is trying to forge a valid signature under PK' whose certificate is not available to **CB-**$\mathcal{A}_I$. **CB-**$\mathcal{A}_{II}$ has the knowledge of CA's master key and thus can generate the certificate for any user. **CB-**$\mathcal{A}_{II}$ is trying to forge a valid signature under an entity's authentic public key PK (that is, PK is chosen by that entity), whose private key is not available to **CB-**$\mathcal{A}_{II}$. In addition to the security models, a certificate-based signature scheme secure against key replacement attacks was also proposed in [Li et al. 2007]. Very recently, Liu *et al.* proposed two new certificate-based signature schemes [Liu et al. 2008]. The first one does not require any pairing operation and the security of their second scheme can be proved without random oracles. Some variants of certificate-based signatures (e.g., certificate-based proxy signature [Kang et al. 2004] and certificate-based linkable ring signature [Au et al. 2007B]) have also been proposed.

1.2 Motivations and Contributions

As mentioned in [Al-Riyami and Paterson 2003], certificate-based cryptography and certificateless cryptography are quite similar and there could be a possible method to convert a certificateless cryptographical protocol to a certificate-based cryptographical protocol. In particular, there are four similarities in certificateless signatures and certificate-based signatures.

1. In both public key cryptosystems, signature signing requires two pieces of secret information. In CL-PKC, one needs a partial private key and a secret value of a public key to produce valid signatures under that public key. Similarly, in certificate-based cryptosystem, one needs the certificate and the private key of a public key to generate valid signatures under that public key.

2. The partial private key is generated by KGC in CL-PKC. The certificate is generated by Certifier in certificate-based cryptosystem.

3. The secret value corresponding to a public key is chosen by the user in CL-PKC. The private key is also chosen by the user in certificate-based cryptosystem.

4. In both public key cryptosystems, explicit verification of the authenticity of a public key is not required when one verifies the validity of signatures under that public key.

Motivated by those similarities, we believe that certificate-based signatures and certificateless signatures are closely related, and the investigation of the relationship between those two notions is worthwhile. The contributions of this paper are twofold.

1. *New Security Models of Certificate-based Signatures*
A reasonable and elaborated security model is necessary for constructing provably secure cryptographic protocols. For example, although the key replacement attack has been widely accepted in certificateless cryptography, there is no consensus on the precise meaning of that term in the early research of certificateless cryptography and several certificateless signature schemes have been broken [Al-Riyami and Paterson 2003, Gorantla and Saxena 2005, Hu et al. 2006, Huang et al. 2005, Park 2006, Yum and Lee 2004, Zhang and Feng 2006]. Although some security models [Kang et al. 2004, Li et al. 2007] have been proposed so far, the security definition of certificate-based signatures is not satisfactory, especially in the exact meaning of key replacement attacks. In this paper, we provide elaborated definitions of certificate-based signatures, which will allow us to establish a systematic approach for constructing and proving secure certificate-based signature schemes. Our definitions are inspired by and modified from the security notions in certificateless signatures. This is not only because certificateless signatures and certificate-based signatures are analogous in many ways, but also due to the fact that security definitions of certificateless signatures have been formalized recently.

2. *Generic Construction of Certificate-based Signatures from Certificateless Signatures*
After giving new security models of certificate-based signatures, we propose a generic construction of certificate-based signatures which is secure in the proposed models. We show how to build a certificate-based signature scheme from a certificateless signature scheme, by treating partial private keys in certificateless signatures as certificates in certificate-based signatures. Our method can be used to build certificate-based signature schemes secure (in the random oracle model) against any type of adversaries defined in this paper, assuming that the underlying certificateless signature schemes satisfy certain security notions. We also give two concrete examples of our generic construction and compare them with other existing ones. From essentially the same idea, the generic construction of certificate-based encryption from certificateless encryption has been proposed in [Al-Riyami and Paterson 2005], but a recent work in [Kang and Park 2005] shows a flaw in the security proof of [Al-Riyami and Paterson 2005]. Our construction does not have that flaw as we use different techniques in the conversion.

Organization of Our Paper
The outline of a certificate-based signature (CBS) scheme is presented in the next section, where the description of the oracles are also given. We then redefine the security of CBS against different types of attacks in Section 3. The generic construction of certificate-based signatures from certificateless signatures is proposed in Section 4. Section 5 demonstrates the application of our generic construction by showing two concrete certificate-based signature schemes. Fi-

nally, Section 6 concludes the paper.

2 Certificate-Based Signatures

In this section, we will first review the definitions of certificate-based signatures. Then, we will describe oracles used in our security model.

2.1 Syntax of Certificate-Based Signatures

In a certificate-based cryptosystem, a certificate generator, which is called as the "certifier", will first generate the system parameter and a master public/private key pair. The certifier will use that key pair to generate certificates for users in the system. Users then will generate their own public/secret key pairs and contact the certifier to obtain the corresponding certificates. A user can use the secret key and the certificate to generate a signature on a message. In this case, that user is also called as the signer. A signature recipient is called as the verifier if he/she performs the signature verification.

A certificate-based signature (CBS) scheme consists of the following five algorithms:

1. $\mathsf{CB\text{-}Setup}(1^k) \rightarrow (CB\text{-}msk,\ CB\text{-}mpk,\ CB\text{-}params)$. By taking as input a security parameter 1^k, the certifier runs the algorithm $\mathsf{CB\text{-}Setup}$ to generate the certifier's master secret key $CB\text{-}msk$, master public key $CB\text{-}mpk$ and the system parameter $CB\text{-}params$. $CB\text{-}params$ includes the description of a string space $\varGamma$, which can be any subset of $\{0,1\}^*$.

2. $\mathsf{CB\text{-}UserKeyGen}(CB\text{-}mpk,\ CB\text{-}params,\ \mathsf{ID}) \rightarrow (CB\text{-}SK_{\mathsf{ID}},\ CB\text{-}PK_{\mathsf{ID}})$. The user with the identity information ID runs the algorithm $\mathsf{CB\text{-}UserKeyGen}$ to generate the user ID's secret/public key pair $(CB\text{-}SK_{\mathsf{ID}},\ CB\text{-}PK_{\mathsf{ID}}) \in \mathcal{SK}^{\mathcal{CB}} \times \mathcal{PK}^{\mathcal{CB}}$, by taking as input $CB\text{-}mpk$ and $CB\text{-}params$. Here, $\mathcal{SK}^{\mathcal{CB}}$ denotes the set of valid secret key values and $\mathcal{PK}^{\mathcal{CB}}$ denotes the set of valid public key values.

3. $\mathsf{CB\text{-}CertGen}(CB\text{-}msk,\ CB\text{-}mpk,\ CB\text{-}params,\ \mathsf{ID},\ CB\text{-}PK_{\mathsf{ID}}) \rightarrow Cert_{\mathsf{ID}}$. The certifier runs the algorithm $\mathsf{CB\text{-}CertGen}$ to generate the certificate $Cert_{\mathsf{ID}}$, by taking as input $CB\text{-}msk$, $CB\text{-}mpk$, $CB\text{-}params$, ID and its public key $CB\text{-}PK_{\mathsf{ID}}$.

4. $\mathsf{CB\text{-}Sign}(m, CB\text{-}params, CB\text{-}mpk, \mathsf{ID}, Cert_{\mathsf{ID}}, CB\text{-}SK_{\mathsf{ID}}, CB\text{-}PK_{\mathsf{ID}}) \rightarrow CB\text{-}\sigma$. The prospective signer runs the algorithm $\mathsf{CB\text{-}Sign}$ to generate the signature $CB\text{-}\sigma$, by taking as input a message m, $CB\text{-}params$, $CB\text{-}mpk$, the user's identity ID, its $Cert_{\mathsf{ID}}$ and key pair $(CB\text{-}SK_{\mathsf{ID}},\ CB\text{-}PK_{\mathsf{ID}})$.

5. $\mathsf{CB\text{-}Verify}(m, CB\text{-}\sigma, CB\text{-}mpk, CB\text{-}params, \mathsf{ID}, CB\text{-}PK_{\mathsf{ID}}) \rightarrow \{true, false\}$. Anyone can run the algorithm CB-Verify to check the validity of the signature. By taking as input a message/signature pair $(m, CB\text{-}\sigma)$, ID, $CB\text{-}PK_{\mathsf{ID}}$, $CB\text{-}mpk$, $CB\text{-}params$, this algorithm outputs $true$ if $CB\text{-}\sigma$ is ID's valid signature on m. Otherwise, this algorithm outputs $false$.

Correctness. Signatures generated by the algorithm CB-Sign can pass through the verification in CB-Verify. That is,
$\mathsf{CB\text{-}Verify}(m, \mathsf{CB\text{-}Sign}(m, CB\text{-}params, CB\text{-}mpk, \mathsf{ID}, Cert_{\mathsf{ID}}, CB\text{-}SK_{\mathsf{ID}}, CB\text{-}PK_{\mathsf{ID}}), CB\text{-}mpk, CB\text{-}params, \mathsf{ID}, CB\text{-}PK_{\mathsf{ID}}) = true$.

2.2 Adversaries and Oracles

We now describe the oracles which will be used in the security model of certificate-based signatures in this paper. We first give a brief description of adversaries in certificate-based signatures. Formal definitions of these adversaries will be given in Section 3.

The essential security of a certificate-based signature scheme requires that one can generate a valid signature under the public key $CB\text{-}PK_{\mathsf{ID}}$ *if and only if* having the knowledge of both $Cert_{\mathsf{ID}}$ and $CB\text{-}SK_{\mathsf{ID}}$. In other words, one cannot generate a valid signature with only $Cert_{\mathsf{ID}}$ or $CB\text{-}SK_{\mathsf{ID}}$. As introduced in [Li et al. 2007], adversaries in certificate-based signatures can be divided into two types: **CB-**$\mathcal{A}_I$ and **CB-**$\mathcal{A}_{II}$. Type I adversary **CB-**$\mathcal{A}_I$ simulates the scenario where the adversary (anyone except the certifier) is allowed to replace public keys of any entities, but is not allowed to obtain the target user's certificate $Cert_{\mathsf{ID}}$. Type II adversary **CB-**$\mathcal{A}_{II}$ simulates a malicious certifier who is able to produce certificates but is assumed not to replace the target user's public key. We will use the following oracles to simulate potential attacking scenarios. In the remainder of this paper, we write $\alpha \leftarrow \beta$ to denote the algorithmic action of assigning the value β to α.

1. $\mathcal{O}^{\mathsf{CB-UserCreate}}$: This oracle receives an input $\mathsf{ID} \in \Gamma$ and outputs the public key of user ID. It maintains two lists $L1^{PK}$ and $L2^{PK}$, which are initially empty and are used to record the information for each user ID. $L1^{PK}$=$\{(\mathsf{ID}, CB\text{-}SK_{\mathsf{ID}}, CB\text{-}PK_{\mathsf{ID}})\}$ provides the information about user ID's secret key and public key when it is created. $L2^{PK}$=$\{(\mathsf{ID}, CB\text{-}\overline{PK}_{\mathsf{ID}})\}$ provides the information of ID's current public key, which is denoted as $CB\text{-}\overline{PK}_{\mathsf{ID}}$.

   (a) For a fresh input ID, the oracle runs the algorithms CB-UserKeyGen to obtain the secret key $CB\text{-}SK_{\mathsf{ID}}$ and public key $CB\text{-}PK_{\mathsf{ID}}$. It then adds $(\mathsf{ID}, CB\text{-}SK_{\mathsf{ID}}, CB\text{-}PK_{\mathsf{ID}})$ to $L1^{PK}$ and $(\mathsf{ID}, CB\text{-}\overline{PK}_{\mathsf{ID}})$ to $L2^{PK}$ where $CB\text{-}\overline{PK}_{\mathsf{ID}} \leftarrow CB\text{-}PK_{\mathsf{ID}}$. After that, it outputs $CB\text{-}PK_{\mathsf{ID}}$. In this case, ID is said to be *created*. Here we assume that other oracles (which will be defined later) only respond to the identity which has been created.

(b) Otherwise, ID has already been created. The oracle will search ID in $L1^{PK}$ and return $CB\text{-}PK_{\mathsf{ID}}$ as the output.

2. $\mathcal{O}^{\mathsf{CB-PKReplace}}$: For a public key replacement query (ID, CB-PK) $\in \Gamma \times \mathcal{PK}^{\mathcal{CB}}$, this oracle finds the user ID in the list $L2^{PK}$, sets $CB\text{-}\overline{PK}_{\mathsf{ID}} \leftarrow$ CB-PK and updates the corresponding pair with (ID, $CB\text{-}\overline{PK}_{\mathsf{ID}}$).
3. $\mathcal{O}^{\mathsf{CB-Corruption}}$: This oracle takes as input a query ID. It browses the list $L1^{PK}$ and outputs the secret key $CB\text{-}SK_{\mathsf{ID}}$.
4. $\mathcal{O}^{\mathsf{CB-CertGen}}$: For a certificate request for (ID, CB-PK) $\in \Gamma \times \mathcal{PK}^{\mathcal{CB}}$, this oracle runs the algorithm CB-CertGen and returns the certificate for (ID, CB-PK).
5. $\mathcal{O}^{\mathsf{CB-Sign}}$: Considering different levels of the signing power the challenger may have, this oracle can be further divided into following three types:
   (a) $\mathcal{O}^{\mathsf{CB-NormalSign}}$: This oracle takes as input a query (ID, m), and outputs a signature $CB\text{-}\sigma$ such that $true = \mathsf{CB-Verify}(m,\ CB\text{-}\sigma,\ CB\text{-}params,$ ID, $CB\text{-}PK_{\mathsf{ID}}$, $CB\text{-}mpk)$. Here $CB\text{-}PK_{\mathsf{ID}}$ is ID's public key in the list $L1^{PK}$.

   (b) $\mathcal{O}^{\mathsf{CB-StrongSign}}$: This oracle takes as input a query $(\mathsf{ID}, m, coin)$, where m denotes the message to be signed, and $coin \in \{1, 2\}$. It acts differently according to the value of *coin*. If $coin = 1$, this oracle works the same as $\mathcal{O}^{\mathsf{CB-NormalSign}}$. Otherwise $coin = 2$, this oracle first checks the list $L1^{PK}$ and $L2^{PK}$ to obtain ID's original public key $CB\text{-}PK_{\mathsf{ID}}$ and ID's current public key $CB\text{-}\overline{PK}_{\mathsf{ID}}$. If $CB\text{-}\overline{PK}_{\mathsf{ID}} = CB\text{-}PK_{\mathsf{ID}}$, this oracle works as same as $\mathcal{O}^{\mathsf{CB-NormalSign}}$. Otherwise, $\mathcal{O}^{\mathsf{CB-StrongSign}}$ will ask the adversary to supply the secret key $CB\text{-}\overline{SK}_{\mathsf{ID}}$ corresponding to $CB\text{-}\overline{PK}_{\mathsf{ID}}$. After that, this oracle uses $CB\text{-}\overline{SK}_{\mathsf{ID}}$ and the certificate for (ID,$CB\text{-}\overline{PK}_{\mathsf{ID}}$) to generate the signature $CB\text{-}\sigma$, which will be returned as the answer.

   (c) $\mathcal{O}^{\mathsf{CB-SuperSign}}$: For a query (ID, m), this oracle first finds ID's current public key $CB\text{-}\overline{PK}_{\mathsf{ID}}$ in $L2^{PK}$. This oracle then outputs a signature σ such that $true = \mathsf{CB-Verify}(m, \sigma,\ CB\text{-}params, \mathsf{ID},\ CB\text{-}\overline{PK}_{\mathsf{ID}},\ CB\text{-}mpk)$.

Remark. A Type II adversary **CB-**$\mathcal{A}_{II}$, who simulates the malicious certifier, is not allowed to make any requests to $\mathcal{O}^{\mathsf{CB-CertGen}}$.

3 Security Models of Certificate-based Signatures

In this section, we will define security models of certificate-based signatures. Our models follow the standard methods: each security notion is defined by the game between the adversary and the challenger, which consists of several oracles defined in Section 2.2. In our definition, the notation $\{Q_1, Q_2, \cdots, Q_n\} \nrightarrow \{\mathcal{O}^1, \mathcal{O}^2, \cdots, \mathcal{O}^n\}$ denotes that "No query $Q \in \{Q_1, Q_2, \cdots, Q_n\}$ can be submitted to any oracle $\mathcal{O} \in \{\mathcal{O}^1, \mathcal{O}^2, \cdots, \mathcal{O}^n\}$. A $(t, q_{UC}, q_{PKR}, q_C, q_{CG}, q_S)$ adversary refers to the adversary who runs in polynomial time t, makes at most q_{UC} queries to $\mathcal{O}^{\mathsf{CB-UserCreate}}$, q_{PKR} queries to $\mathcal{O}^{\mathsf{CB-PKReplace}}$, q_C queries to $\mathcal{O}^{\mathsf{CB-Corruption}}$, q_{CG} queries to $\mathcal{O}^{\mathsf{CB-CertGen}}$, q_S queries to $\mathcal{O}^{\mathsf{CB-Sign}} \in \{\mathcal{O}^{\mathsf{CB-NormalSign}}, \mathcal{O}^{\mathsf{CB-StrongSign}}, \mathcal{O}^{\mathsf{CB-SuperSign}}\}$.

The definition in this section is inspired by [Huang et al. 2007], which provides a new classification of potential adversaries against certificateless signatures. The security models in [Huang et al. 2007] not only include previous security definitions of certificateless signatures, but also introduce new types of adversaries. Following the definitions in [Huang et al. 2007], we classify the potential adversaries in certificate-based signatures according to their attack power. They are Normal Adversary, Strong Adversary and Super Adversary. Combined with the known type I adversary and type II adversary in certificate-based signatures, we now define the security of certificate-based signatures in different attack scenarios and relate them to prior definitions.

3.1 Security Against Normal Type I Adversary

We first define the Normal Type I adversary in certificate-based signatures, which is denoted as **Normal-CB-$\mathcal{A}_I$**. The essential attacking scenario of Normal-CB-$\mathcal{A}_I$ is that the adversary can obtain some message/signature pairs $(m_i, CB\text{-}\sigma_i)$ which are generated by the target user using its own secret key and certificate. Our definition described below is inspired by and modified from the definition of Normal Type I adversary against certificateless signatures in [Huang et al. 2007].

Initial: The challenger runs the algorithm $\mathsf{CB\text{-}Setup}$, returns $CB\text{-}params$ and $CB\text{-}mpk$ to $\mathcal{A}_I$.
Queries: In this phase, $\mathcal{A}_I$ can adaptively make requests to $\mathcal{O}^{\mathsf{CB-UserCreate}}$, $\mathcal{O}^{\mathsf{CB-PKReplace}}$, $\mathcal{O}^{\mathsf{CB-Corruption}}$, $\mathcal{O}^{\mathsf{CB-CertGen}}$, $\mathcal{O}^{\mathsf{CB-NormalSign}}$.
Output: After all queries, $\mathcal{A}_I$ outputs a forgery $(m^*, CB\text{-}\sigma^*, \mathsf{ID}^*)$. Let $CB\text{-}\overline{PK}_{\mathsf{ID}^*}$ be the current public key of ID^* in $L2^{PK}$.
Restrictions: We say $\mathcal{A}_I$ wins the game if the forgery satisfies the following requirements: (1) $true = \mathsf{CB{-}Verify}(m^*, CB\text{-}\sigma^*, CB\text{-}params, \mathsf{ID}^*, CB\text{-}\overline{PK}_{\mathsf{ID}^*}, CB\text{-}mpk)$; (2) $(\mathsf{ID}^*, m^*) \nrightarrow \mathcal{O}^{\mathsf{CB-NormalSign}}$; (3) $(\mathsf{ID}^*, CB\text{-}\overline{PK}_{\mathsf{ID}^*}) \nrightarrow \mathcal{O}^{\mathsf{CB-CertGen}}$ and (4) $\mathsf{ID}^* \nrightarrow \mathcal{O}^{\mathsf{CB-Corruption}}$.

The success probability that an adaptive chosen message and chosen identity adversary Normal-CB-$\mathcal{A}_I$ wins the above game is denoted as $Succ^{cma,cida}_{\mathcal{A}_I,normal}$. We say a certificate-based signature scheme is secure against a $(t, q_{UC}, q_{PKR}, q_C, q_{CG}, q_S)$ Normal-CB-$\mathcal{A}_I$ if $Succ^{cma,cida}_{\mathcal{A}_I,normal}$ is negligible.

Remark. Our definition is similar to that in [Li et al. 2007], but with two improvements. Firstly, we allow the adversary to replace any user's public key, while the adversary in [Li et al. 2007] can only replace the target user's public key. The other improvement is that the adversary in our model is allowed to obtain certificates of (ID, $CB\text{-}PK$)s chosen by itself. This is different from the adversary in [Li et al. 2007] who can only obtain certificates of original public keys generated by the challenger.

3.2 Security Against Strong Type I Adversary

In this section, we boost the attack power of Normal Type I adversary and define the Strong Type I adversary: **Strong-CB-**$\mathcal{A}_I$. Strong-CB-$\mathcal{A}_I$ is more powerful than Normal-CB-$\mathcal{A}_I$ in the sense that Strong-CB-$\mathcal{A}_I$ can access the oracle $\mathcal{O}^{\mathsf{CB-StrongSign}}$. Apart from that, Strong-CB-$\mathcal{A}_I$ is allowed to corrupt the target user ID^*'s original secret key. The attacking scenario is similar to those in certificateless signatures defined in [Hu et al. 2006, Zhang et al. 2006], and is formally defined as below.

The game between the challenger and a Strong-CB-$\mathcal{A}_I$ is very similar to that defined in Section 3.1, but with two differences: (1) In the phase **Queries**, Strong-CB-$\mathcal{A}_I$ have access to $\mathcal{O}^{\mathsf{CB-StrongSign}}$ rather than $\mathcal{O}^{\mathsf{CB-NormalSign}}$ and (2) In **Restrictions**, $(\mathsf{ID}^*, m^*) \nrightarrow \mathcal{O}^{\mathsf{CB-StrongSign}}$ and ID^* can appear as a query to $\mathcal{O}^{\mathsf{CB-Corruption}}$.

The success probability that an adaptive chosen message and chosen identity adversary Strong-CB-$\mathcal{A}_I$ wins the above game is denoted as $Succ^{cma,cida}_{\mathcal{A}_I,strong}$. We say a certificate-based signature scheme is secure against a $(t, q_{UC}, q_{PKR}, q_C, q_{CG}, q_S)$ Strong-CB-$\mathcal{A}_I$ if $Succ^{cma,cida}_{\mathcal{A}_I,strong}$ is negligible.

3.3 Security Against Super Type I Adversary

In this section, we will define the Super Type I adversary, which is denoted as **Super-CB-**$\mathcal{A}_I$. Super-CB-$\mathcal{A}_I$ is more powerful than Strong-CB-$\mathcal{A}_I$ (and hence, more powerful than Normal-CB-$\mathcal{A}_I$) in the sense that Super-CB-$\mathcal{A}_I$ has access to $\mathcal{O}^{\mathsf{CB-SuperSign}}$. That is, Super-CB-$\mathcal{A}_I$ is allowed to obtain a valid signature under the public key chosen by itself without providing the corresponding secret key, which makes it the strongest Type I adversary. This is similar to the Super Type I adversary in certificateless signatures defined in [Huang et al. 2007].

The game between the challenger and a Super-CB-$\mathcal{A}_I$ is very similar to that defined in Section 3.2, but with two differences: (1) In the phase **Queries**, Super-CB-$\mathcal{A}_I$ is allowed to have access to $\mathcal{O}^{\mathsf{CB-SuperSign}}$ rather than $\mathcal{O}^{\mathsf{CB-StrongSign}}$ and (2) In **Restrictions**, $(\mathsf{ID}^*, m^*) \nrightarrow \mathcal{O}^{\mathsf{CB-SuperSign}}$.

The success probability of an adaptively chosen message and chosen identity adversary Super-CB-$\mathcal{A}_I$ wins the above game is denoted as $Succ^{cma,cida}_{\mathcal{A}_I,super}$. We say a certificate-based signature scheme is secure against a $(t, q_{UC}, q_{PKR}, q_C, q_{CG}, q_S)$ Super-CB-$\mathcal{A}_I$ if $Succ^{cma,cida}_{\mathcal{A}_I,super}$ is negligible.

3.4 Security Against Type II Adversary

In certificate-based signatures, a type II adversary **CB-**$\mathcal{A}_{II}$ simulates the certifier who is equipped with the master secret key and might engage in adversarial activities like eavesdropping on signatures and making signing queries. Similar to the type I adversary, CB-$\mathcal{A}_{II}$ could be also classified into **Normal-CB-**$\mathcal{A}_{II}$, **Strong-CB-**$\mathcal{A}_{II}$, **Super-CB-**$\mathcal{A}_{II}$, which has access to $\mathcal{O}^{\mathsf{CB-NormalSign}}$, $\mathcal{O}^{\mathsf{CB-StrongSign}}$, $\mathcal{O}^{\mathsf{CB-SuperSign}}$, respectively. However, there is no need to particularly define Strong-CB-$\mathcal{A}_{II}$. $\mathcal{O}^{\mathsf{CB-StrongSign}}$ can answer queries either by using $\mathcal{O}^{\mathsf{CB-NormalSign}}$ (then $\mathcal{O}^{\mathsf{CB-StrongSign}}$ is the same as $\mathcal{O}^{\mathsf{CB-NormalSign}}$), or signing the message with the corresponding secret key provided by the adversary. Note that, CB-$\mathcal{A}_{II}$ has the master secret key, and thus can calculate any user's certificate. If he has the secret key as well, CB-$\mathcal{A}_{II}$ can generate the signature by himself and $\mathcal{O}^{\mathsf{CB-StrongSign}}$ becomes useless. Therefore, for a type II adversary CB-$\mathcal{A}_{II}$, it is sufficient to define only two types of adversaries, namely Normal-CB-$\mathcal{A}_{II}$ and Super-CB-$\mathcal{A}_{II}$. The definition of those two types of adversaries is described as follows.

Initial: The challenger runs the algorithm $\mathsf{CB\text{-}Setup}$ and returns the system parameters *CB-params*, master secret key *CB-msk* and master public key *CB-mpk* to $\mathcal{A}_{II}$.

Queries: $\mathcal{A}_{II}$ can adaptively make requests to $\mathcal{O}^{\mathsf{CB-UserCreate}}$, $\mathcal{O}^{\mathsf{CB-PKReplace}}$, $\mathcal{O}^{\mathsf{CB-Corruption}}$ and $\mathcal{O}^{\mathsf{CB-Sign}}$, where $\mathcal{O}^{\mathsf{CB-Sign}} \in \{\mathcal{O}^{\mathsf{CB-NormalSign}}, \mathcal{O}^{\mathsf{CB-SuperSign}}\}$.

Output: After all queries, $\mathcal{A}_{II}$ outputs a forgery $(m^*, CB\text{-}\sigma^*, \mathsf{ID}^*)$.

Restrictions: We say $\mathcal{A}_{II}$ wins the game if the forgery satisfies the requirements as following: (1) $true \leftarrow \mathsf{CB-Verify}(m, CB\text{-}\sigma^*, CB\text{-}params, \mathsf{ID}^*, CB\text{-}PK_{\mathsf{ID}^*}, CB\text{-}mpk)$. Here $CB\text{-}PK_{\mathsf{ID}^*}$ is the original public key in $L1^{PK}$; (2) $(\mathsf{ID}^*, m^*) \nrightarrow \mathcal{O}^{\mathsf{CB-Sign}}$; and (3) $\mathsf{ID}^* \nrightarrow \mathcal{O}^{\mathsf{CB-Corruption}}$.

The success probability that an adaptive chosen message and chosen identity adversary CB-$\mathcal{A}_{II}$ wins the above game is denoted as $Succ^{cma,cida}_{\mathcal{A}_{II},type}$, where $type \in \{normal, super\}$. We say a certificate-based signature scheme is secure against a $(t, q_{UC}, q_{PKR}, q_C, q_S)$ CB-$\mathcal{A}_{II}$if $Succ^{cma,cida}_{\mathcal{A}_{II},type}$ is negligible. Here, $\mathcal{O}^{\mathsf{CB-Sign}}$ will be $\mathcal{O}^{\mathsf{CB-NormalSign}}$ if $type = normal$. Otherwise, $\mathcal{O}^{\mathsf{CB-Sign}}$ is $\mathcal{O}^{\mathsf{CB-SuperSign}}$.

3.5 Security Against Malicious-but-Passive Type II Adversary

We now define a more powerful type II adversary, who is allowed to generate the system parameter and the master secret/public key. This assumes that the third party certifier have already been malicious at the very beginning of the setup stage of the system, rather than being only given the parameter and the master secret/public key honestly generated by the challenger. Such attacks are first introduced to certificateless cryptosystems in [Au et al. 2007A]. In addition to this, even though we say that the certifier is malicious, we also assume (as in [Au et al. 2007A]) that the certifier is passive, in the sense that the certifier would not actively replace the user's public key or corrupt the user's secret key. It is shown in [Au et al. 2007A] that the malicious-but-passive third party KGC in certificateless cryptosystems like [Al-Riyami and Paterson 2003] can have its master key pair specifically generated so that all the encrypted messages for the target victim can also be decrypted by the KGC. The security model of certificate-based encryption in [Gentry 2003] also captures the essence of those attacks. The security of certificate-based signatures against a malicious-but-passive Type II adversary is defined by the following game:

Initial: The challenger executes $\mathcal{A}_{II}$ on the security parameter 1^k. $\mathcal{A}_{II}$ returns the system parameters *CB-params* and master public key *CB-mpk*.

Queries: A malicious-but-passive $\mathcal{A}_{II}$ can make queries to all oracles except $\mathcal{O}^{\mathsf{CB-CertGen}}$. Since *CB-params* and *CB-mpk* are generated by $\mathcal{A}_{II}$, $\mathcal{O}^{\mathsf{CB-UserCreate}}$ and $\mathcal{O}^{\mathsf{CB-PKReplace}}$ have to be modified as following:

$\mathcal{O}^{\mathsf{CB-UserCreate}}$: As defined in Sec. 2.2, this oracle receives an input $\mathsf{ID} \in \Gamma$ and outputs the public key $CB\text{-}PK_{\mathsf{ID}}$. After obtaining $CB\text{-}PK_{\mathsf{ID}}$, a malicious-but-passive Type II adversary must provide ID's certificate $Cert_{\mathsf{ID}}$ for (ID, $CB\text{-}PK_{\mathsf{ID}}$). This oracle then adds (ID, $CB\text{-}SK_{\mathsf{ID}}$, $CB\text{-}PK_{\mathsf{ID}}$, $Cert_{\mathsf{ID}}$) to $L1^{PK}$, and (ID, $CB\text{-}PK_{\mathsf{ID}}$, $Cert_{\mathsf{ID}}$) to $L2^{PK}$.

$\mathcal{O}^{\mathsf{CB-PKReplace}}$: For a malicious-but-passive Type II adversary, the input to this oracle should be (ID, $\mathsf{CB\text{-}PK}$, $Cert$) where $Cert$ is the corresponding certificate of $\mathsf{CB\text{-}PK}$ under the identity ID. This oracle searches $L2^{PK}$, finds a record related to ID and sets $CB\text{-}\overline{PK}_{\mathsf{ID}} \leftarrow \mathsf{CB\text{-}PK}$ and $\overline{Cert}_{\mathsf{ID}} \leftarrow Cert$. It then updates the related tuple with (ID, $CB\text{-}\overline{PK}_{\mathsf{ID}}$, $\overline{Cert}_{\mathsf{ID}}$).

Output and **Restrictions**: Same as those defined in Sec. 3.4.

The success probability that an adaptive chosen message and chosen identity malicious-but-passive Type II adversary **CB-**$\mathcal{A}_{II}$ wins the above game is denoted as $Succ^{cma,cida}_{MP-\mathcal{A}_{II},type}$, where $type \in \{normal, super\}$. We say a certificate-based signature scheme is secure against a $(t, q_{UC}, q_{PKR}, q_C, q_S)$ malicious-but-passive CB-$\mathcal{A}_{II}$ if $Succ^{cma,cida}_{MP-\mathcal{A}_{II},type}$ is negligible. Here, $\mathcal{O}^{\mathsf{CB-Sign}}$ will be $\mathcal{O}^{\mathsf{CB-NormalSign}}$ if $type = normal$. Otherwise, $\mathcal{O}^{\mathsf{CB-Sign}}$ is $\mathcal{O}^{\mathsf{CB-SuperSign}}$.

4 Generic Construction of Certificate-based Signatures

In this section, we will introduce a generic method to construct certificate-based signatures. Our construction is based on certificateless signatures whose description is as below.

4.1 Syntax of Certificateless Signatures

A certificateless signature (CLS) scheme is defined by six algorithms: CL-Setup (generates KGC's key pair ($CL\text{-}msk$, $CL\text{-}mpk$) and system's parameter), CL-PPKExtract (generates a user ID's partial private key $CL\text{-}PPK_{ID}$), CL-SSValue (generates a user ID's secret value $CL\text{-}SV_{ID}$), CL-SPKey (generates a user ID's public key $CL\text{-}PK_{ID}$), CL-Sign (generates a certificateless signature $CL\text{-}\sigma$ using $CL\text{-}PPK_{ID}$ and $CL\text{-}SV_{ID}$) and CL-Verify (outputs $true$ if a given signature is valid, or $false$ otherwise). As one can see, to distinguish from the identity information in the certificate-based system (which is denoted as ID), we use the notion ID to denote the identity information in the certificateless system. For other notations, we put the prefix "CL-" to indicate that they are in the certificateless cryptosystem. Please refer to [Huang et al. 2007] for the formal definition of each algorithm.

4.2 Generic Construction: CLS-2-CBS

In this section, we show how to convert a certificateless signature scheme into a certificate-based signature scheme. In our construction, we need a hash function $H : \Gamma \times \mathcal{PK}^{\mathcal{CB}} \rightarrow \mathcal{ID}^{\mathcal{CL}}$. Here, Γ is the identity information space in the certificate-based system, $\mathcal{PK}^{\mathcal{CB}}$ is the public key space in certificate-based system and $\mathcal{ID}^{\mathcal{CL}}$ denotes the space of identities in the certificateless cryptosystem[3].

Let CLS be the certificateless signature scheme described in Section 4.1. We now describe the generic construction CLS-2-CBS.

1. CB-Setup$(1^k) \rightarrow (CB\text{-}msk, CB\text{-}mpk, CB\text{-}params)$.
   (a) Run algorithm CL-Setup(1^k) of CLS to obtain $CL\text{-}params$, $CL\text{-}msk$ and $CL\text{-}mpk$. For the security parameter k, we assume that the public key size in a certificateless cryptosystem is at least 2^k;
   (b) Set $CB\text{-}params$ by extending $CL\text{-}params$ to include the description of Γ;

[3] Here, we use the hash function H to "connect" two identities in certificate-based signatures and certificateless signatures. This is different from the technique in the generic construction of certificate-based encryption proposed in [Al-Riyami and Paterson 2005]. A recent work [Kang and Park 2005] pointed out a flaw of security proof in [Al-Riyami and Paterson 2005]. That flaw does not exist in our construction, the details of which will be shown in the security proof later.

(c) $(CB\text{-}msk,\ CB\text{-}mpk) \leftarrow (CL\text{-}msk,\ CL\text{-}mpk)$.

2. $\mathsf{CB\text{-}UserCreate}(CB\text{-}mpk,\ CB\text{-}params, \mathsf{ID} \in \Gamma) \rightarrow (CB\text{-}SK_{\mathsf{ID}},\ CB\text{-}PK_{\mathsf{ID}})$.
   (a) $CL\text{-}mpk \leftarrow CB\text{-}mpk$;
   (b) Extract $CL\text{-}params$ from $CB\text{-}params$;
   (c) $CB\text{-}SK_{\mathsf{ID}} \leftarrow \mathsf{CL\text{-}SSValue}(CL\text{-}mpk,\ CL\text{-}params)$;
   (d) $CB\text{-}PK_{\mathsf{ID}} \leftarrow \mathsf{CL\text{-}SPKey}(CL\text{-}mpk,\ CL\text{-}params,\ CB\text{-}SK_{\mathsf{ID}})$.
3. $\mathsf{CB\text{-}CertGen}(CB\text{-}msk,\ CB\text{-}mpk,\ CB\text{-}params,\ \mathsf{ID} \in \Gamma,\ CB\text{-}PK_{\mathsf{ID}}) \rightarrow Cert_{\mathsf{ID}}$.
   (a) $(CL\text{-}msk,\ CL\text{-}mpk) \leftarrow (CB\text{-}msk,\ CB\text{-}mpk)$;
   (b) Extract $CL\text{-}params$ from $CB\text{-}params$;
   (c) $H(\mathsf{ID},\ CB\text{-}PK_{\mathsf{ID}}) \rightarrow ID \in \mathcal{ID}^{\mathcal{CL}}$;
   (d) $Cert_{\mathsf{ID}} \leftarrow \mathsf{CL\text{-}PPKExtract}(CL\text{-}msk,\ CL\text{-}mpk,\ CL\text{-}params, ID)$.
4. $\mathsf{CB\text{-}Sign}(m, CB\text{-}params, CB\text{-}mpk, \mathsf{ID}, Cert_{\mathsf{ID}}, CB\text{-}SK_{\mathsf{ID}}, CB\text{-}PK_{\mathsf{ID}}) \rightarrow CB\text{-}\sigma$.
   (a) Extract $CL\text{-}params$ from $CB\text{-}params$;
   (b) $CL\text{-}mpk \leftarrow CB\text{-}mpk$;
   (c) $H(\mathsf{ID},\ CB\text{-}PK_{\mathsf{ID}}) \rightarrow ID \in \mathcal{ID}^{\mathcal{CL}}$;
   (d) $(CL\text{-}SV_{ID},\ CL\text{-}PK_{ID}) \leftarrow (CB\text{-}SK_{\mathsf{ID}},\ CB\text{-}PK_{\mathsf{ID}})$ and $CL\text{-}PPK_{ID} \leftarrow Cert_{\mathsf{ID}}$;
   (e) $CB\text{-}\sigma \leftarrow \mathsf{CL\text{-}Sign}(m,\ CL\text{-}params,\ CL\text{-}mpk,\ ID,\ CL\text{-}SV_{ID},\ CL\text{-}PK_{ID},\ CL\text{-}PPK_{ID})$. One can see that the signature size of $CB\text{-}\sigma$ is the same as that in the underlying certificateless signature scheme.
5. $\mathsf{CB\text{-}Verify}(CB\text{-}params,\ CB\text{-}mpk,\ \mathsf{ID},\ CB\text{-}PK_{\mathsf{ID}},\ (m,\ CB\text{-}\sigma)) \rightarrow \{true,\ false\}$.
   (a) Extract $CL\text{-}params$ from $CB\text{-}params$;
   (b) $CL\text{-}mpk \leftarrow CB\text{-}mpk$;
   (c) $H(\mathsf{ID},\ CB\text{-}PK_{\mathsf{ID}}) \rightarrow ID \in \mathcal{ID}^{\mathcal{CL}}$;
   (d) $CL\text{-}PK_{ID} \leftarrow CB\text{-}PK_{\mathsf{ID}}$;
   (e) $CL\text{-}\sigma \leftarrow CB\text{-}\sigma$.
   (f) Output $\mathsf{CL\text{-}Verify}(CL\text{-}mpk,\ CL\text{-}params, ID,\ CL\text{-}PK_{ID},\ (m,\ CL\text{-}\sigma))$.

Correctness. We show that any certificate-based signature produced by CB-Sign will pass through the verification in CB-Verify.

In our construction, a certificate-based signature is the output of the algorithm CL-Sign in the certificateless system, and algorithm CB-Verify also employs the verification algorithm CL-Verify in the certificateless system. To show the correctness of our construction, it suffices to show that under the same $CL\text{-}params$ and $CL\text{-}mpk$, a certificateless signature produced by using the secret value $CL\text{-}SV_{ID}$ and the partial private key $CL\text{-}PPK_{ID}$ will pass through the check using the corresponding identity ID and its public key $CL\text{-}PK_{ID}$. This is ensured by the correctness of the underlying certificateless signature scheme, that is, for any signature $CL\text{-}\sigma$ produced by CL-Sign(m, $CL\text{-}params$, $CL\text{-}mpk$, ID, $CL\text{-}SV_{ID}$, $CL\text{-}PK_{ID}$, $CL\text{-}PPK_{ID}$), CL-Verify($CL\text{-}mpk$, $CL\text{-}params, ID$, $CL\text{-}PK_{ID}$, (m, $CL\text{-}\sigma$)) will output *true*. Therefore, for any signature output by CB-Sign defined in our construction, the algorithm CB-Verify will always output *true*.

Security Analysis

Theorem 1. *[Security of* CLS-2-CBS*]* CLS-2-CBS *is secure (in the random oracle model) against adversaries defined in Section 3, assuming the underlying certificateless signature scheme* CLS *satisfying certain security requirements.*

The proof of Theorem 1 consists of several lemmas, which demonstrate the security relationship between our generic construction CLS-2-CBS and its underlying certificateless signature scheme CLS. Please refer to [Huang et al. 2007] for security definitions of CLS.

Lemma 2 Security against Normal-CB-$\mathcal{A}_I$. CLS-2-CBS *is secure (in the random oracle model) against* ***Normal-CB-***$\mathcal{A}_I$ *defined in Section 3.1, if* CLS *is secure against* ***Normal-CL-***$\mathcal{A}_I$ *defined in [Huang et al. 2007].*

Proof. In the proof, we will regard hash function H as the random oracle and show that if there is a Normal-CB-$\mathcal{A}_I$ who can forge a valid certificate-based signature of CLS-2-CBS with non-negligible probability, then there exists a Normal-CL-$\mathcal{A}_I$ who can use Normal-CB-$\mathcal{A}_I$ to forge a valid certificateless signature of CLS with almost the same probability.

In our proof, the challenger of a Normal-CB-$\mathcal{A}_I$ is the Normal-CL-$\mathcal{A}_I$ against the underlying CLS, who can make requests to its own challenger CL-Challenger. CL-Challenger is made up of several oracles as follows: $\mathcal{O}^{\mathsf{CL-UserCreate}}$ (creates users in the certificateless cryptosystem), $\mathcal{O}^{\mathsf{CL-Corruption}}$ (returns secret values of created users), $\mathcal{O}^{\mathsf{CL-PPKExtract}}$ (returns partial private keys of created users), $\mathcal{O}^{\mathsf{CL-PKReplace}}$ (replaces public keys of created users with the value provided by the adversary) and $\mathcal{O}^{\mathsf{CL-NormalSign}}$ (returns certificateless signatures on messages chosen by the adversary). Please refer to [Huang et al. 2007] for the formal definition of each oracle. The description of our proof is as follows.

Initial: The $\mathsf{CL\text{-}Challenger}$ runs the algorithm $\mathsf{CL\text{-}Setup}$ of CLS and feeds the Normal-CL-$\mathcal{A}_I$ with $CL\text{-}mpk$ and $CL\text{-}params$. Normal-CL-$\mathcal{A}_I$ then returns $CB\text{-}mpk$ and $CB\text{-}params$ to Normal-CB-$\mathcal{A}_I$ where $CB\text{-}mpk$ is defined to be $CL\text{-}mpk$ and $CB\text{-}params$ is defined by extending $CL\text{-}params$ to include the description of Γ. Before Normal-CB-$\mathcal{A}_I$ submits any queries, Normal-CL-$\mathcal{A}_I$ asks $\mathsf{CL\text{-}Challenger}$ to create q_{uc} users in the certificateless cryptosystem. Here q_{uc} is the number of queries Normal-CB-$\mathcal{A}_I$ issues to $\mathcal{O}^{\mathsf{CB-UserCreate}}$. Normal-CL-$\mathcal{A}_I$ then records the information as $(ID_i, CL\text{-}PK_{ID_i})$, $i = 1, 2, \cdots, q_{uc}$ in the list CL^{PK}. Here $ID_i \in \mathcal{ID}^{\mathcal{CL}}$, $CL\text{-}PK_{ID_i}$ is the original public key of ID_i in certificateless system.
Queries: As defined in Section 3.1, Normal-CB-$\mathcal{A}_I$ can issue queries to following oracles. We now show how Normal-CL-$\mathcal{A}_I$ can answer these queries.

– $\mathcal{RO}$: In the proof, the hash function H is viewed as the random oracle $\mathcal{RO}$. For a fresh input $(\mathsf{ID},\mathsf{CB\text{-}PK}) \in \Gamma \times \mathcal{PK}^{CB}$, the output of $\mathcal{RO}$ is a random element ID in $\mathcal{ID}^{CL}$. Normal-CL-$\mathcal{A}_I$ maintains a list $L^{\mathcal{RO}}$ consisting of $(\mathsf{ID},\mathsf{CB\text{-}PK},ID)$.

– $\mathcal{O}^{\mathsf{CB-UserCreate}}$: At any time Normal-CB-$\mathcal{A}_I$ can request to create the user $\mathsf{ID}_i \in \Gamma$ and expect to obtain ID_i's public key $CB\text{-}PK_{\mathsf{ID}_i}$. In response to such queries:

1. For the i^{th} fresh query ID_i, Normal-CL-$\mathcal{A}_I$ first checks the list CL^{PK} and finds the i^{th} pair $(ID_i, CL\text{-}PK_{ID_i})$. Normal-CL-$\mathcal{A}_I$ then sets $CB\text{-}PK_{\mathsf{ID}_i} \leftarrow CL\text{-}PK_{ID_i}$, $H(\mathsf{ID}_i, CB\text{-}PK_{\mathsf{ID}_i}) = ID_i$ and adds $(\mathsf{ID}_i, CB\text{-}PK_{\mathsf{ID}_i}, ID_i)$ into $L^{\mathcal{RO}}$. If $(\mathsf{ID}_i, CB\text{-}PK_{\mathsf{ID}_i})$ already appears in $L^{\mathcal{RO}}$, then the simulation fails and Normal-CL-$\mathcal{A}_I$ aborts. This, however, happens only with negligible probability as $|\mathcal{PK}^{\mathcal{CL}}|$ is assumed to be greater than 2^k and k is the security parameter. Otherwise, it adds $(\mathsf{ID}_i, \perp, CB\text{-}PK_{\mathsf{ID}_i})$ into the list $L1^{PK}$. Meanwhile, it sets $CB\text{-}\overline{PK}_{\mathsf{ID}_i} \leftarrow CB\text{-}PK_{\mathsf{ID}_i}$ and adds $(\mathsf{ID}_i, CB\text{-}\overline{PK}_{\mathsf{ID}_i})$ into list $L2^{PK}$. Here, the notation $\perp$ means that Normal-CL-$\mathcal{A}_I$ does not know the corresponding secret key $CB\text{-}SK_{\mathsf{ID}_i}$.

2. In addition to maintain $L1^{PK}, L2^{PK}$, Normal-CL-$\mathcal{A}_I$ will keep two additional lists $L1^{ID}$ and $L2^{ID}$ which will help it answer queries from Normal-CB-$\mathcal{A}_I$. $L1^{ID}$ consists of pairs with the form (ID_i, ID_i) where $\mathsf{ID}_i \in \Gamma$ and $ID_i \in \mathcal{ID}^{\mathcal{CL}}$. This list will help Normal-CL-$\mathcal{A}_I$ to respond Normal-CB-$\mathcal{A}_I$'s corruption queries and NormalSign queries. $L2^{ID}$ consists of pairs with the form $(\mathsf{ID}_i, \overline{ID_i})$ where $\mathsf{ID}_i \in \Gamma$ and $\overline{ID_i} \in \mathcal{ID}^{\mathcal{CL}}$. In different phases, $\overline{ID_i}$ could be the identity ID_i in the list $L1^{PK}$, or the identity $ID'_i \in \mathcal{ID}^{\mathcal{CL}}$ created at some time later.

   For a user ID_i created in this oracle, Normal-CL-$\mathcal{A}_I$ will add (ID_i, ID_i) into list $L1^{ID}$, where $ID_i \in \mathcal{ID}^{\mathcal{CL}}$ is ID_i's corresponding identity in the list CL^{PK}. Meanwhile, Normal-CL-$\mathcal{A}_I$ sets $\overline{ID_i} \leftarrow ID_i$ and adds $(\mathsf{ID}_i, \overline{ID_i})$ into list $L2^{ID}$.

- $\mathcal{O}^{\mathsf{CB-PKReplace}}$: At any time Normal-CB-$\mathcal{A}_I$ can replace a public key of a created user ID with the public key $CB\text{-}PK'_{\mathsf{ID}}$ chosen by himself. In response, Normal-CL-$\mathcal{A}_I$ first sets $CB\text{-}\overline{PK}_{\mathsf{ID}} \xleftarrow{\$} CB\text{-}PK'_{\mathsf{ID}}$, then

  1. Normal-CL-$\mathcal{A}_I$ browses the list $L2^{PK}$ and rewrites the related pair as $(\mathsf{ID},\ CB\text{-}\overline{PK}_{\mathsf{ID}})$. It then browses $L^{\mathcal{RO}}$.

  2. If $(\mathsf{ID},\ CB\text{-}\overline{PK}_{\mathsf{ID}})$ appears in $L^{\mathcal{RO}}$ in the tuple $(\mathsf{ID},\ CB\text{-}\overline{PK}_{\mathsf{ID}},\ \overline{ID})$, Normal-CL-$\mathcal{A}_I$ will make a user-create query $\overline{ID}$ to CL-Challenger if $\overline{ID}$ has not been created in the certificateless system. After that, Normal-CL-$\mathcal{A}_I$ replaces $\overline{ID}$'s certificateless public key with $CB\text{-}\overline{PK}_{\mathsf{ID}}$ and updates the corresponding pair in CL^{PK} with $(\overline{ID},\ CB\text{-}\overline{PK}_{\mathsf{ID}})$. Finally, Normal-CL-$\mathcal{A}_I$ browses the list $L2^{ID}$ and updates the related pair with $(\mathsf{ID}, \overline{ID})$.

  3. Otherwise, Normal-CL-$\mathcal{A}_I$ sets $H(\mathsf{ID},CB\text{-}\overline{PK}_{\mathsf{ID}})=\overline{ID}$, which is randomly chosen in $\mathcal{ID}^{\mathcal{CL}}$. It then adds $(\mathsf{ID},\ CB\text{-}\overline{PK}_{\mathsf{ID}},\ \overline{ID})$ into $L^{\mathcal{RO}}$. After that, Normal-CL-$\mathcal{A}_I$ asks CL-Challenger to create the user $\overline{ID}$. After creating the identity $\overline{ID}$, Normal-CL-$\mathcal{A}_I$ replaces $\overline{ID}$'s public key with $CB\text{-}\overline{PK}_{\mathsf{ID}}$. Normal-CL-$\mathcal{A}_I$ then updates CL^{PK} by adding $(\overline{ID},\ CB\text{-}\overline{PK}_{\mathsf{ID}})$. Finally, Normal-CL-$\mathcal{A}_I$ browses the list $L2^{ID}$ and updates the related pair with $(\mathsf{ID}, \overline{ID})$.

- $\mathcal{O}^{\mathsf{CB-Corruption}}$: At any time Normal-CB-$\mathcal{A}_I$ can request the secret key of a created user ID_i. In response, Normal-CL-$\mathcal{A}_I$ checks the list $L1^{ID}$ and finds (ID_i, ID_i). Then, it issues a corruption request ID_i to CL-Challenger who will return $CL\text{-}SV_{ID_i}$ to Normal-CL-$\mathcal{A}_I$, where $CL\text{-}SV_{ID_i}$ is the secret value of ID_i when it was created in the certificateless system. At last, Normal-CL-$\mathcal{A}_I$ sets $CB\text{-}SK_{\mathsf{ID}_i} \leftarrow CL\text{-}SV_{ID_i}$, returns it to Normal-CB-$\mathcal{A}_I$ and updates the information in the list $L1^{PK}$ as $(\mathsf{ID}_i,\ CB\text{-}SK_{\mathsf{ID}_i},\ CB\text{-}PK_{\mathsf{ID}_i})$.

  Correctness: This oracle should return the user ID_i's original secret key $CB\text{-}SK_{\mathsf{ID}_i}$ when the user was created. Recall that $L1^{ID}$ contains pairs (ID_i, ID_i) $i = 1, 2, \cdots, q_{uc}$, where $ID_i \in \mathcal{ID}^{\mathcal{CL}}$ is the ID_i's initial corresponding identity in certificateless system. ID_i is set as $H(\mathsf{ID}_i,CB\text{-}PK_{\mathsf{ID}_i})$ and $CL\text{-}PK_{ID_i}$=$CB\text{-}PK_{\mathsf{ID}_i}$ which is the original public key of ID_i. Thus the secret value $CL\text{-}SV_{ID_i}$ of ID_i in certificateless system is the same as the secret key $CB\text{-}SK_{\mathsf{ID}_i}$ of ID_i in certificate-based system.

- $\mathcal{O}^{\mathsf{CB-CertGen}}$: At any time Normal-CB-$\mathcal{A}_I$ can request the certificate of (ID, CB-PK) where CB-PK is chosen by the adversary itself. Normal-CL-$\mathcal{A}_I$ will try to find an identity $\overline{ID} \in \mathcal{ID}^{\mathcal{CL}}$, whose partial private key is ID's certificate under the public key CB-PK. To do that, Normal-CL-$\mathcal{A}_I$ will check $L^{\mathcal{RO}}$:

1. If (ID, CB-PK) appears in $L^{\mathcal{RO}}$ in the tuple (ID, CB-PK, $\overline{ID}$), Normal-CL-$\mathcal{A}_I$ will make a user-create query $\overline{ID}$ to CL-Challenger if $\overline{ID}$ has not been created in certificateless system.

2. Otherwise, Normal-CL-$\mathcal{A}_I$ sets H(ID,CB-PK)=$\overline{ID}$, which is randomly chosen in $\mathcal{ID}^{\mathcal{CL}}$. It then adds (ID, CB-PK, $\overline{ID}$) into $L^{\mathcal{RO}}$. After that, Normal-CL-$\mathcal{A}_I$ asks CL-Challenger to create the user $\overline{ID}$.

For either case, Normal-CL-$\mathcal{A}_I$ issues the partial private key query $\overline{ID}$ to CL-Challenger who will return the partial private key $CL\text{-}PPK_{\overline{ID}}$. At last, Normal-CL-$\mathcal{A}_I$ sets $Cert_{\mathsf{ID}} \leftarrow CL\text{-}PPK_{\overline{ID}}$ and returns it to Normal-CB-$\mathcal{A}_I$.

– $\mathcal{O}^{\mathsf{CB-NormalSign}}$: At any time, Normal-CB-$\mathcal{A}_I$ can request the signature of (m_i, ID_i). The Normal-CL-$\mathcal{A}_I$ first finds the pair (ID_i, ID_i) in the list $L1^{ID}$. Then, Normal-CL-$\mathcal{A}_I$ issues a certificateless signing query (ID_i, m_i). As defined, Normal-CL-$\mathcal{A}_I$ will obtain the signature $CL\text{-}\sigma_i$ such that $true =$ CL-Verify$(CL\text{-}mpk, CL\text{-}params, ID_i, CL\text{-}PK_{ID_i}, (m_i, CL\text{-}\sigma_i))$. Normal-CB-$\mathcal{A}_I$ will set $CB\text{-}\sigma_i \leftarrow CL\text{-}\sigma_i$, and return $CB\text{-}\sigma_i$ as the answer.

Correctness: Recall that for the pair (ID_i, ID_i) in $L1^{ID}$, ID_i is set as $H(\mathsf{ID}_i, CB\text{-}PK_{\mathsf{ID}_i})$ and $CB\text{-}PK_{\mathsf{ID}_i} = CL\text{-}PK_{ID_i}$. Here, $CB\text{-}PK_{\mathsf{ID}_i}$ is ID_i's original public key in the list $L1^{PK}$. Therefore, $true =$CB-Verify($CB\text{-}params$, $CB\text{-}mpk$, ID_i, $CB\text{-}PK_{\mathsf{ID}_i}$, $(m_i, CB\text{-}\sigma_i)$). That is, $CB\text{-}\sigma_i$ is ID_i's valid signature for m_i under the original public key returned from $\mathcal{O}^{\mathsf{CB-UserCreate}}$.

Output: After all queries, CB-$\mathcal{A}_I$ will output a forgery (m^*, $CB\text{-}\sigma^*$, ID^*). If Normal-CB-$\mathcal{A}_I$ wins game, then:

1. $true$ = CB−Verify(m^*, $CB\text{-}\sigma^*$, $CB\text{-}params, \mathsf{ID}^*$, $CB\text{-}\overline{PK}_{\mathsf{ID}^*}$, $CB\text{-}mpk$), where (ID^*, $CB\text{-}\overline{PK}_{\mathsf{ID}^*}$) is in the list $L2^{PK}$. Here, $CB\text{-}\overline{PK}_{\mathsf{ID}^*}$ is ID^*'s current public key.

   That is, if $CL\text{-}\sigma^* \leftarrow CB\text{-}\sigma^*$, $true =$ CL-Verify($CL\text{-}mpk$, $CL\text{-}params, \overline{ID^*}$, $CL\text{-}\overline{PK}_{\overline{ID^*}}$, ($m^*$, $CL\text{-}\sigma^*$)). Here, $(\mathsf{ID}^*, \overline{ID^*}) \in L2^{ID}$ which indicates that $CL\text{-}\overline{PK}_{\overline{ID^*}} = CB\text{-}\overline{PK}_{\mathsf{ID}^*}$.

2. $(\mathsf{ID}^*, m^*) \nrightarrow \mathcal{O}^{\mathsf{CB-NormalSign}}$.

   That is, $(\overline{ID^*}, m^*)$ has never been asked to $\mathcal{O}^{\mathsf{CL-NormalSign}}$ of CLS.

3. (ID^*, $CB\text{-}\overline{PK}_{ID^*}$) $\nrightarrow \mathcal{O}^{\mathsf{CB-CertGen}}$.

   That is, $\overline{ID^*}$ has never been asked to $\mathcal{O}^{\mathsf{CL-PPKExtract}}$ of CLS.

4. $\mathsf{ID}^* \nrightarrow \mathcal{O}^{\mathsf{CB-Corruption}}$.

   That is, $\overline{ID^*}$ has never been asked to $\mathcal{O}^{\mathsf{CL-Corruption}}$ of CLS.

If Normal-CL-$\mathcal{A}_I$ does not fail in the simulation, then it can output a valid forgery (m^*, CL-σ^*, $\overline{ID^*}$) of the underlying certificateless signature scheme with the same success probability as Normal-CB-$\mathcal{A}_I$. Considering that Normal-CL-$\mathcal{A}_I$ could only fail in simulating H as the random oracle, which only happens with negligible probability $q_{UC}/2^k$ (q_{UC} is the number of user-create queries). Thus, Normal-CL-$\mathcal{A}_I$ wins the game with almost the same probability as Normal-CB-$\mathcal{A}_I$. This completes the proof of Lemma 2.

Security against Strong-CB-$\mathcal{A}_I$ and Super-CB-$\mathcal{A}_I$.
One can use almost the same technique to prove that our generic construction is secure against Strong-CB-$\mathcal{A}_I$ (or, Super-CB-$\mathcal{A}_I$), if the underlying certificateless signature scheme is also secure against Strong-CL-$\mathcal{A}_I$ (or, Super-CL-$\mathcal{A}_I$) defined in [Huang et al. 2007]. The details are thus omitted here.

Lemma 3 Security against CB-$\mathcal{A}_{II}$. CLS-2-CBS *is secure (in the random oracle model) against type II adversary* ***CB-***$\mathcal{A}_{II}$ *defined in Section 3.4, if* CLS *is secure against* ***CL-***$\mathcal{A}_{II}$.

Proof. In the proof, we will regard the hash function H as the random oracle and show that if there is a CB-$\mathcal{A}_{II}$ (either Normal-CB-$\mathcal{A}_{II}$ or Super-CB-$\mathcal{A}_{II}$) who can forge a valid certificate-based signature of CLS-2-CBS with non-negligible probability, then there exists a CL-$\mathcal{A}_{II}$(correspondingly, Normal-CL-$\mathcal{A}_{II}$, or Super-CL-$\mathcal{A}_{II}$) who can use CB-$\mathcal{A}_{II}$ to forge a valid certificateless signature of CLS with almost the same probability.

In our proof, the challenger of a CB-$\mathcal{A}_{II}$ is the CL-$\mathcal{A}_{II}$ against the underlying certificateless signature scheme, who can make requests to its own challenger CL-Challenger. The description of our proof is as follows.

Initial: The CL-Challenger runs the algorithm CL-Setup of CLS and feeds the CL-$\mathcal{A}_{II}$ with CL-msk, CL-mpk and CL-$params$. CL-$\mathcal{A}_{II}$ then returns (CB-msk, CB-mpk) and CB-$params$ to CB-$\mathcal{A}_{II}$ where (CB-msk, CB-mpk) is defined to be (CL-msk, CL-mpk) and CB-$params$ is defined by extending CL-$params$ to include the description of Γ. Before CB-$\mathcal{A}_{II}$ submits any queries, CL-$\mathcal{A}_{II}$ asks CL-Challenger to create q_{uc} users in the certificateless cryptosystem. Here q_{uc} is the number of queries CB-$\mathcal{A}_{II}$ issues to $\mathcal{O}^{\mathsf{CB-UserCreate}}$. CL-$\mathcal{A}_{II}$ then records the information as (ID_i, CL-PK_{ID_i}), $i = 1, 2, \cdots, q_{uc}$ in the list CL^{PK}. Here $ID_i \in \mathcal{ID}^{\mathcal{CL}}$, CL-PK_{ID_i} is the original public key of ID_i in certificateless system.

Queries: As defined in Section 3.4, CB-$\mathcal{A}_{II}$ can issue queries to $\mathcal{O}^{\mathsf{CB-UserCreate}}$, $\mathcal{O}^{\mathsf{CB-PKReplace}}$, $\mathcal{O}^{\mathsf{CB-Corruption}}$, $\mathcal{O}^{\mathsf{CB-NormalSign}}$ (or, $\mathcal{O}^{\mathsf{CB-SuperSign}}$). These oracles are simulated by CL-$\mathcal{A}_{II}$ in the same way as described in the proof of the security against Type I adversary in Lemma 2.

Output: After all queries, CB-$\mathcal{A}_{II}$ will output a forgery (m^*, CB-σ^*, ID^*). If CB-$\mathcal{A}_{II}$ wins, then:

1. $true = \mathsf{CB{-}Verify}(m^*, CB\text{-}\sigma^*, CB\text{-}params, \mathsf{ID}^*, CB\text{-}PK_{\mathsf{ID}^*}, CB\text{-}mpk)$, where $(\mathsf{ID}^*, CB\text{-}PK_{\mathsf{ID}^*})$ is in the list $L1^{PK}$, that is, $CB\text{-}PK_{\mathsf{ID}^*}$ is ID^*'s original public key.

   That is, if $CL\text{-}\sigma^* \leftarrow CB\text{-}\sigma^*$, $true = \mathsf{CL\text{-}Verify}(CL\text{-}mpk, CL\text{-}params, ID^*, CL\text{-}PK_{ID^*}, (m^*, CL\text{-}\sigma^*))$, where $(\mathsf{ID}^*, ID^*) \in L1^{ID}$.

2. $(\mathsf{ID}^*, m^*) \nrightarrow \mathcal{O}^{\mathsf{CB-Sign}}$.

   That is, (ID^*, m^*) has never been asked to $\mathcal{O}^{\mathsf{CL-Sign}}$ of CLS.

3. $\mathsf{ID}^* \nrightarrow \mathcal{O}^{\mathsf{CB-Corruption}}$.

   That is, ID^* has never been asked to $\mathcal{O}^{\mathsf{CL-Corruption}}$ of CLS.

If CL-$\mathcal{A}_{II}$ does not fail in the simulation, then it can output a valid forgery (m^*, $CL\text{-}\sigma^*$, ID^*) of the underlying certificateless signature scheme with the same success probability as CB-$\mathcal{A}_{II}$. As in the proof of Lemma 2, CL-$\mathcal{A}_{II}$ could only fail in simulating H as the random oracle, which only happens with negligible probability $q_{UC}/2^k$ (q_{UC} is the number of user-create queries). Thus, CL-$\mathcal{A}_{II}$ wins the game with almost the same probability as CB-$\mathcal{A}_{II}$. This completes the proof of Lemma 3.

Security Against Malicious-but-Passive Type II Adversary.
One can use almost the same technique to prove that our generic construction is secure against malicious-but-passive CB-$\mathcal{A}_{II}$, if the underlying certificateless signature scheme is also secure against malicious-but-passive CL-$\mathcal{A}_{II}$ defined in [Huang et al. 2007]. The details are thus omitted here.

5 Concrete Examples of CLS-2-CBS

By applying $\mathsf{CLS\text{-}2\text{-}CBS}$ to concrete certificateless signature schemes, we can obtain several new constructions of certificate-based signatures. This section will describe two of them, which are constructed from certificateless signature schemes proposed in [Huang et al. 2007]. We start by reviewing the bilinear groups and the complexity assumption in [Huang et al. 2007].

5.1 Bilinear Groups and Security Assumptions

Let $\mathbb{G}_1$ denote an additive group of prime order p and $\mathbb{G}_T$ be a multiplicative group of the same order. Let P denote a generator in $\mathbb{G}_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ be a bilinear mapping with the following properties:

- The map e is bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1, a, b \in \mathbb{Z}_p$.
- The map e is non-degenerate: $e(P, P) \neq 1_{\mathbb{G}_T}$.

– The map e is efficiently computable.

We say that $(\mathbb{G}_1, \mathbb{G}_T)$ are bilinear groups if there exists the bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ as above, and e, and the group action in $\mathbb{G}_1$ and $\mathbb{G}_T$ can be computed efficiently.

Definition 4. Computational Diffie-Hellman (CDH) Problem in $\mathbb{G}_1$. Given (P, aP, bP), for some unknown $a, b \in \mathbb{Z}_p$, compute abP.

The success probability of any probabilistic polynomial-time algorithm $\mathcal{A}$ in solving CDH problem in $\mathbb{G}_1$ is defined to be $Succ^{CDH}_{\mathcal{A},\mathbb{G}_1} = \Pr[\mathcal{A}(P, aP, bP) = abP]$ where the probability is over the random choice of $a, b \in \mathbb{Z}_p$ and the random bits consumed by $\mathcal{A}$.

5.2 Scheme I

The scheme described in this section is based on the certificateless signature scheme in Section 4.2 of [Huang et al. 2007]. It consists of following algorithms.

– $\mathsf{CB\text{-}Setup}$: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$, for some prime number $p \geq 2^k$, where k is the system security number. e denotes the bilinear mapping $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. Let $H_0, H_1 : \{0,1\}^* \to \mathbb{G}_1^*$ and $H_2 : \Gamma \times \mathbb{G}_1^* \to \mathcal{ID}$ be three secure cryptographic hash functions, where Γ is the set of identity information in the certificate-based system and $\mathcal{ID}$ is the identity space defined in [Huang et al. 2007]. The certifier chooses a random number $s \in \mathbb{Z}_p^*$ and a generator P of $\mathbb{G}_1^*$. The certifier then calculates system's master public key $CB\text{-}mpk{=}\ sP$, where s is the master secret key $CB{-}msk$. The system's parameter $CB\text{-}params$ is $\{\mathbb{G}_1, \mathbb{G}_T, p, e, P, H_0, H_1, H_2,\ CB\text{-}mpk,\ \Gamma\}$.

– $\mathsf{CB\text{-}UserCreate}$: The user ID chooses a random number $x_{\mathsf{ID}} \in \mathbb{Z}_p^*$ and sets x_{ID} as the secret key. Here the valid secret key value space is $\mathcal{SK}^{\mathcal{CB}} = \mathbb{Z}_p^*$. User ID can also calculate the public key $CB\text{-}PK_{\mathsf{ID}}{=}\ x_{\mathsf{ID}}P$. Here the valid public key space is $\mathcal{PK}^{\mathcal{CB}} = \mathbb{G}_1^*$.

– $\mathsf{CB\text{-}CertGen}$: Given a user's identity information ID, the certifier first sets $\hat{\mathsf{ID}} = H_2(\mathsf{ID} \| CB\text{-}PK_{\mathsf{ID}})$, then computes $Cert_{\mathsf{ID}} = sH_0(\hat{\mathsf{ID}})$.

– $\mathsf{CB\text{-}Sign}$: For a message m, the user ID sets $\hat{\mathsf{ID}} = H_2(\mathsf{ID} \| CB\text{-}PK_{\mathsf{ID}})$ and computes the signature $CB\text{-}\sigma = Cert_{\mathsf{ID}} + x_{\mathsf{ID}} H_1(m \| \hat{\mathsf{ID}} \| CB\text{-}PK_{\mathsf{ID}})$.

– $\mathsf{CB\text{-}Verify}$: Given a pair (m, σ) and user ID's public key $CB\text{-}PK_{\mathsf{ID}}$, after setting $\hat{\mathsf{ID}} = H_2(\mathsf{ID} \| CB\text{-}PK_{\mathsf{ID}})$, anyone can check whether $e(CB\text{-}\sigma, P) \stackrel{?}{=} e(H_0(\hat{\mathsf{ID}}), CB\text{-}mpk) e(H_1(m \| \hat{\mathsf{ID}} \| CB\text{-}PK_{\mathsf{ID}}), CB\text{-}PK_{\mathsf{ID}})$. If the equality holds, this algorithm outputs $true$. Otherwise, this algorithm outputs $false$.

Theorem 5 Security of Concrete Scheme I. *Scheme I is secure (in the random oracle model) against Normal-CB-$\mathcal{A}_I$ and Super-CB-$\mathcal{A}_{II}$ adaptive chosen message and chosen identity attacks, assuming that CDH problem is hard in $\mathbb{G}_1$.*

Proof. The correctness of this theorem is due to Theorem 1 and the underlying certificateless signature scheme is provably secure (in the random oracle model) against Normal-CL-$\mathcal{A}_I$ and Super-CL-$\mathcal{A}_{II}$ if CDH problem is hard in $\mathbb{G}_1$ [Huang et al. 2007].

5.3 Scheme II

The scheme described in this section is based on the certificateless signature scheme in Section 4.3 of [Huang et al. 2007]. The first four algorithms are the same as those defined in Section 5.2, with the only difference that H_1 is defined as $\{0,1\}^* \rightarrow \mathbb{Z}_p$. The $\mathsf{CB\text{-}Sign}$ and $\mathsf{CB\text{-}Verify}$ algorithms are described as follows:

– $\mathsf{CB\text{-}Sign}$: For a message m, the user ID sets $\hat{\mathsf{ID}} = H_2(\mathsf{ID}\|CB\text{-}PK_{\mathsf{ID}})$ and computes the signature $\sigma = (u, v, W)$ where
  - $u = H_1(m\|\hat{\mathsf{ID}}\|CB\text{-}PK_{\mathsf{ID}}\|r_1P\|e(P,P)^{r_2})$ for random numbers $r_1, r_2 \in \mathbb{Z}_p$ chosen by user ID; and
  - $v = r_1 - ux_{\mathsf{ID}} \pmod p$, $W = r_2P - uCert_{\mathsf{ID}}$.

– $\mathsf{CB\text{-}Verify}$: Given a message/signature pair $(m, \sigma = (u, v, W))$, ID's public key $CB{-}PK_{\mathsf{ID}}$, by setting $\hat{\mathsf{ID}} = H_2(\mathsf{ID}\|CB\text{-}PK_{\mathsf{ID}})$ anyone can check whether $u \stackrel{?}{=} H_1(m\|\hat{\mathsf{ID}}\|\ CB\text{-}PK_{\mathsf{ID}}\|vP + uCB\text{-}PK_{\mathsf{ID}}\|e(W,P)e(CB{-}mpk, H_0(\hat{\mathsf{ID}}))^u)$. If the equality holds, this algorithm outputs $true$. Otherwise, this algorithm outputs $false$.

Theorem 6 Security of Concrete Scheme II. *Scheme II is secure (in the random oracle model) against Super-CB-$\mathcal{A}_I$ and Super-CB-$\mathcal{A}_{II}$ adaptive chosen message and chosen identity attacks, assuming that CDH problem is hard in $\mathbb{G}_1$.*

Proof. The correctness of this theorem is due to Theorem 1 and the underlying certificateless signature scheme is provably secure (in the random oracle model) against Super-CL-$\mathcal{A}_I$ and Super-CL-$\mathcal{A}_{II}$ if CDH problem is hard in $\mathbb{G}_1$ [Huang et al. 2007].

Remark. Scheme II is the first certificate-based signature scheme which is provably secure against Super Type I and Type II adversary.

Table 1: Efficiency Comparison

Scheme	Length	Signing Cost	Verification Cost
CBS in [Li et al. 2007]	$2\lvert\mathbb{G}_1\rvert$	$3E{+}2PA$	$4BM$
Scheme I	$\lvert\mathbb{G}_1\rvert$	$E{+}PA$	$3BM$
CBSa in [Kang et al. 2004]	$3\lvert\mathbb{G}_1\rvert$	$3E$	$2E{+}3BM{+}2PA$
Scheme II	$\lvert\mathbb{G}_1\rvert + 2\lvert\mathbb{Z}_p\rvert$	$4E{+}BM{+}PA$	$3E{+}2BM{+}PA$

Table 2: Security Level Comparison

Scheme	Security
CBS in [Li et al. 2007]	*Normal* $\mathcal{A}_I$ and $\mathcal{A}_{II}$
Scheme I	*Normal* $\mathcal{A}_I$ and *Super* $\mathcal{A}_{II}$
CBSa in [Kang et al. 2004]	*Strong* $\mathcal{A}_I$ and $\mathcal{A}_{II}$
Scheme II	*Super* $\mathcal{A}_I$ and $\mathcal{A}_{II}$

5.4 Efficiency Comparison

We now make a comparison among existing certificate-based signature schemes, which are proposed in [Liu et al. 2008, Li et al. 2007, Kang et al. 2004][4].

When compared to schemes (e.g. the first scheme in [Liu et al. 2008]) without bilinear mapping, our Scheme I and Scheme II have more computational cost but shorter signature length. When compared to schemes (e.g. the second scheme in [Liu et al. 2008]) whose security is proved without random oracles, Scheme I and Scheme II have advantages of shorter system parameter, less computational cost and shorter signature length. The comparison among other schemes with similar constructions as ours is shown in Table 1 and Table 2. The notations in Table 1 are as follows: $|\mathbb{G}_1|$ and $|\mathbb{Z}_p|$ denote the bit length of an element in $\mathbb{G}_1$ and $\mathbb{Z}_p$, respectively; E denotes the exponentiation in $\mathbb{G}_1$; BM and PA denote the bilinear mapping operation and point addition in $\mathbb{G}_1$, respectively.

As shown in Table 2, Scheme I and CBS in [Li et al. 2007] have the similar security level (To be more precisely, our Scheme I is provably secure against Normal $\mathcal{A}_I$ and Super $\mathcal{A}_{II}$, and CBS in [Li et al. 2007] is provably secure against Normal $\mathcal{A}_I$ and $\mathcal{A}_{II}$), while Scheme I has less computational operation and shorter signature length . The certificate-based signature scheme CBSa in [Kang et al. 2004] is secure against the adversary similar to the strong adversary defined in this paper, while Scheme II is secure against the super adversary with comparable computational cost and signature length. In addition, the two pairing operations ($e(P, P)$ and $e(CB{-}mpk, H_0(\hat{ID}))$) in Scheme II can be computed in an off-line manner, which can further improve the efficiency of Scheme II. The compari-

[4] As the notion of certificate-based signatures is relatively new, those are the only known certificate-based signature schemes with formal security analysis.

son shows that by applying our generic construction to efficient certificateless signature schemes, one can obtain new certificate-based signature schemes with better performance than existing ones.

6 Conclusion

The focus of this paper was on certificate-based signatures. We demonstrated the pros and the cons of certificate-based signatures, by comparing it with digital signatures in other popular public key systems. Then, we defined several new types of adversaries and gave a new security model of certificate-based signatures. Our model is more elaborated by comparison with other existing security models of certificate-based signatures. We proposed a generic construction of certificate-based signatures from certificateless signatures. Our generic construction is secure (in the random oracle model) under the security model proposed in this paper if the underlying certificateless signature scheme satisfies certain security notions. Finally, we gave two concrete instances (with different security levels) of our generic constructions.

References

[Au et al. 2007A] Au, M. H., Chen, J., Liu, J., Mu, Y., Wong, D. and Yang, G.: Malicious KGC Attacks in Certificateless Cryptography. ASIACCS 2007, ACM, (2007), 302–311. Also available at `http://eprint.iacr.org/2006/255`.

[Au et al. 2007B] Au, M. H., Liu, J., Susilo, W. and Yuen,T. H.: Certificate Based (Linkable) Ring Signature. In: Information Security Practice and Experience 2007, Lecture Notes in Computer Science Vol. 4464, Springer-Verlag, (2007) 79–92.

[Al-Riyami and Paterson 2005] Al-Riyami, S.S., Paterson, K.G.: CBE from CL-PKE: A Generic Construction and Efficient Schemes. In S. Vaudenay (ed.), PKC 2005, Lecture Notes in Computer Science Vol. 3386, Springer-Verlag, (2005) 398–415.

[Al-Riyami and Paterson 2003] Al-Riyami, S.S., Paterson, K.G.: Certificateless Public Key Cryptography. Advances in Cryptology-Asiacrypt'03. Lecture Notes in Computer Science, Vol.2894. Springer-Verlag, (2003) 452-473.

[Boneh and Franklin 2001] Boneh, D., Franklin, M.: Identity-based Encryption from the Weil Pairing. SIAM J. Comput. 32(2003) 586-615. A Preliminary Version Appeared In: Kilian, J. (ed.): Advances in Cryptology-Crypto'2001. Lecture Notes in Computer Science, Vol. 2139. Springer-Verlag, (2001) 231–229.

[Bellare et al. 2004] Bellare, M. Namprempre, C and Neven,G. Security Proofs for Identity-based Identification and Signature Schemes. In EUROCRYPT 2004, Lecture Notes in Computer Science, Vol. 3027, Springer-Verlag, (2004) 268–286.

[Gentry 2003] Gentry, C.: Certificate-based Encryption and the Certificate Revocation Problem. In: Biham, E. (ed.): Advances in Cryptology-Eurorypt'03. Lecture Notes in Computer Science, Vol. 2656. Springer-Verlag, (2003) 272–293.

[Galindo et al. 2006] Galindo, D, Herranz, J, and Kiltz, E. On the Generic Construction of Identity-Based Signatures with Additional Properties. In: ASIACRYPT 2006. Lecture Notes in Computer Science, Vol. 4284, Springer-Verlag, (2006) 178-193.

[Girault 1991] Girault, M.: Self-Certified Public Keys. In: Advances in Cryptology-EUROCRYPT'91. Lecture Notes in Computer Science, Vol. 547, Springer-Verlag, (1991) 490-497.

[Gorantla and Saxena 2005] Gorantla, M. C., Saxena, A.: An Efficient Certificateless Signature Scheme. In: Computational Intelligence and Security 2005. Lecture Notes in Computer Science, Vol. 3802. Springer-Verlag, (2005) 110–116.

[Huang et al. 2005] Huang, X., Susilo, W., Mu, Y., Zhang, F.: On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In: The 4th International Conference on Cryptology and Network Security - CANS 2005. Lecture Notes in Computer Science, Vol. 3810. Springer-Verlag, (2005) 13–25.

[Hu et al. 2006] Hu, B. C., Wong, D. S., Zhang, Z., Deng, X.: Key Replacement Attack Against a Generic Construction of Certificateless Signature. In: Information Security and Privacy - ACISP 2006. Lecture Notes in Computer Science, Vol. 4058. Springer-Verlag, (2006) 235–246.

[Huang et al. 2007] Huang, X., Mu, Y., Susilo, W., Wong, D.S., Wu, W.: Certificateless Signature Revisited. In: ACISP 2007. Lecture Notes in Computer Science, Vol. 4586. Springer-Verlag, (2007) 308–322.

[Liu et al. 2008] Liu, J., Baek, J., Susilo, W. and Zhou, J: Certificate Based Signature Schemes without Pairings or Random Oracles. 11th Information Security Conference (ISC'08) Springer Verlag, 2008 (to appear). Also available at `http://eprint.iacr.org/2008/275`.

[Kang and Park 2005] Kang, Go H and Park, Je H.: Is it possible to have CBE from CL-PKE? In: Cryptology ePrint Archive. Also available at `http://eprint.iacr.org/2005/431`.

[Kang et al. 2004] Kang, B.G., Park, J. H., Hahn, S.G.: A Certificate-based Signature Scheme. In: Okamoto, T. (ed.): CT-RSA'04. Lecture Notes in Computer Science, Vol. 2964. Springer-Verlag, (2004) 99–111.

[Li et al. 2007] Li, J., Huang, X., Mu, Y., Susilo, W., Wu. Q: Certificate-Based Signature: Security Model and Efficient Construction. In: 4th European PKI Workshop: Theory and Practice, EuroPKI 2007, Lecture Notes in Computer Science, Vol. 4582. Springer-Verlag, (2007) 110-125.

[Park 2006] Park. J. H.: An Attack on the Certificateless Signature Scheme from EUC Workshops 2006. In: Cryptology ePrint Archive. Also available at `http://eprint.iacr.org/2006/442`.

[Shamir 1985] Shamir, A.: Identity-based Cryptosystems and Signature Schemes. In: Blakley, G.R., Chaum, D. (eds.): Advances in Cryptology-Crypto'84. Lecture Notes in Computer Science, Vol. 196. Springer-Verlag, (1985) 47–53.

[Wu et al. 2008] Wu, W., Mu, Yi., Susilo, W., Huang, X: Certificate-Based Signatures: New Definitions and A Generic Construction from Certificateless Signatures: WISA 2008. Lecture Notes in Computer Science, Vol. 5379. Springer-Verlag, (2008) 99–114.

[Yum and Lee 2004] Yum, D. H., Lee, P. J.: Generic Construction of Certificateless Signature. In: Information Security and Privacy - ACISP 2004. Lecture Notes in Computer Science, Vol. 3108. Springer-Verlag, (2004) 200–211.

[Yap et al. 2006] Yap, W-S., Heng, S-H., Goi, B-M.: An Efficient Certificateless Signature Scheme. In: Emerging Directions in Embedded and Ubiquitous Computing 2006. Lecture Notes in Computer Science, Vol. 4097. Springer-Verlag, (2006) 322–331.

[Zhang et al. 2006] Zhang, Z., Wong, D. S., Xu, J., Feng, D.: Certificateless Public-Key Signature: Security Model and Efficient Construction. In: Applied Cryptography and Network Security 2006. Lecture Notes in Computer Science, Vol. 3989. Springer-Verlag, (2006) 293–308.

[Zhang and Feng 2006] Zhang, Z., Feng, D.: Key Replacement Attack on a Certificateless Signature Scheme. In: Cryptology ePrint Archive. Also available at `http://eprint.iacr.org/2006/453`.