INFORMATICA, 2017, Vol. 28, No. 2, 215–235

DOI: http://dx.doi.org/10.15388/Informatica.2017.127

Generic Construction of Certificate-Based Signature from Certificateless Signature with Provable Security

Wei GAO[1,2*], Guilin WANG[3], Kefei CHEN[4], Xueli WANG[5]
[1]*School of Mathematics and Statistics, Ludong University, Yantai 264025, China*
[2]*Nanjing University of Information Science & Technology, Nanjing 210044, China*
[3]*Huawei Technologies Co. Ltd., Singapore*
[4]*Department of Mathematics, Hangzhou Normal University, Hangzhou 311121, China*
[5]*School of Mathematics, South China Normal University, Guangzhou 510631, China*
e-mail: mygaowei@163.com, wang.guilin@huawei.com, kfchen@hznu.edu.cn, wangxuyuyan@gmail.com

Received: July 2015; accepted: November 2016

Abstract. This paper studies the generic construction of certificate-based signature (CBS) from certificateless signature (CLS). This paper proposes a new generic conversion from CLS to CBS which is more intuitive, simpler, and provably secure without random oracles than the current one. To develop the security proof, we put forth one novel CLS security model which features a previously neglected but nontrivial attack and hence captures the CLS security notion more comprehensively. We show that many existing CLS schemes can be proved secure in the current model by slightly modifying its original security proof. Following this conversion, many provably secure CBS schemes can be constructed from the corresponding existing CLS schemes.

Key words: certificateless signature, certificate-based signature, identity based signature, provable security.

1. Introduction

Identity-based Cryptography (IBC). In 1984, Shamir (1984) proposed the concept of identity-based PKC (IBC) for reducing the requirements on the public key infrastructure. In IBC, the public key for one entity is its well-known identity information, such as the email address and the full name. The private key for this entity is generated by one trusted party called Private Key Generater (PKG), usually being the signature on the identity information. With this approach, the certification of public keys becomes implicit. However, IBC suffers from the main drawback of being inherently key escrowed, since PKG can generate the private key of any entity.

Certificate-based Cryptography (CBC). Motivated by avoiding the problem of third party queries in PKC, Gentry proposed the new concept of certificate-based encryption (CBE) by combining public key encryption and identity based encryption. Following

[*] Corresponding author.

the idea of CBE, Kang *et al.* (2004) proposed the concept of certificate based-signature (CBS). This certificate has all of the functionality of a conventional PKI certificate – e.g. it can be used explicitly as proof of current certification – but it can also be used as part of the user decryption/signing key, which is composed of the user-generated private key and the certificate. As a result, CBC achieves two main advantages. On the one hand, unlike IBC, there is no key escrow for CBC, since CA does not know the user-generated private key. On the other hand, the encrypter (or signature verifier) does not need to verify the certificate, because the ciphertext (or signature) can not be decrypted (or generated) unless the certificate exists. This allows CBC to eliminate third party queries on certificate status (Gentry, 2003). By refining basic CBE/CBS through the use of subset covers, an exceptionally efficient PKI can be constructed (Gentry, 2003).

Certificateless cryptography (CLC). Similarly motivated by combining merits of the traditional PKC and IBC, independently from the work of CBC, the concept of certificateless encryption (CLE) and certificateless signature (CLS) were introduced by Al-Riyami and Paterson (2003). In CLC, each entity has two secrets: a secret value SV chosen by the entity and a partial private key PPK generated by a third party called Private Key Generater (PKG). The full private key is the output of a function by taking SV and PPK as the input, and hence can be only known by the user. On the one hand, unlike IBC, CLC does not suffer from key escrow, since PKG does not have access to the user's secret value SV. CLC does not require the use of certificates to guarantee the authenticity of public key. The two concepts of CBC and CLC are similar. The main difference is the generation of the partial secret key in CLC (it is called certificate in CBC). In CLC the PKG does not require the user's public key for the generation of the partial secret key, while in CBC the CA does require the user's public key for the generation of the certificate. Additionally, the trust level of CBC is 3, while the trust level of CLC is 2 (Liu *et al.*, 2007). All the above three kinds of public key cryptographic primitives have the considerable advantages in key managements which make them potentially suitable for many applications such as wireless networks (Guo *et al.*, 2014; Xie and Wang, 2014; Shen *et al.*, 2015) where key management will be additional burden for restricted resources.

1.1. *Related Works*

Although CBC and CLC were developed independently, both of them can be conceptually seen as intermediates between traditional PKC and IBC, seeking to simplify the management of certificates while avoiding the key escrow problem in IBC. So a natural question to establish the connection of two concepts arose in the theoretical cryptography field. In 2005, Al-Riyami and Paterson (2005) presented one generic conversion from CLE to CBE and claimed the provable security of this generic construction of CBE. Shortly later, Kang and Park (2005) pointed out that this generic conversion from CLE to CBE has a critical flaw in the security proof. Recently, Wu *et al.* (2009) proposed a new generic conversion from CLS to CBS, and a generic conversion from CLE to CBE (Wu *et al.*, 2012). These two generic conversions have to depend on the cryptographic hash function which is taken as a random oracle (Canetti *et al.*, 2004) in the security proof. In addition to the generic results of CBS, there are also some concrete CBS schemes proposed in Kang *et al.* (2004), Liu *et al.* (2008, 2011), Wu *et al.* (2009, 2012), Li *et al.* (2010, 2012). In current research of cryptography, there are much more results for certificateless cryptography than certificate-based cryptography.

1.2. *Motivations*

From the perspective of theoretical cryptography, it is very interesting to investigate the general relationship between certificate-based signatures and certificateless signatures. As mentioned in Wu *et al.* (2012), those two cryptographic primitives are quite closely related. In fact, four similarities in certificateless signatures and certificate-based signatures are presented in Wu *et al.* (2012). However, the state-of-the-art result in Wu *et al.* (2012) is not satisfactory enough in the viewpoint of theory, since the undesirable primitive "random oracle" (Bellare and Rogaway, 1993) is needed in this framework:

"CLS + Random Oracle → CBS".

In the field of provable secure cryptography, researchers usually try to construct cryptographic schemes in the standard model instead of the random oracle model. In contrast, it is a natural and intuitive task in theory to wipe out "Random Oracle", or to get the most concise framework:

"CLS → CBS".

From the perspective of cryptographic practice, it is also worthwhile to study this most concise framework. In fact, there are much more research works on CLS schemes than those in CBS schemes. Hence, the study on generic construction of CBS from CLS can help us to obtain many concrete CBS schemes from existing CLS schemes with almost no price. Among those constructed in this way, some CBS schemes may have better properties than existing ones.

1.3. *Contributions*

A new security model for CLS is developed. It captures some new attacks which have never been mentioned in literature. For more details, please refer to Section 2 where two remarks aim to explain its originality. From the viewpoint of basic theory, this further development of security model is an essential refinement for certificateless cryptography theory. From the viewpoint of applications, it is the key technique helping us to prove secure our generic conversion of CBS schemes from CLS schemes.

The most intuitive generic conversion of CLS schemes from CBS schemes is "revived". Compared with the Wu *et al.*'s (2009) generic conversion from CLS to CBS, our generic conversion is as intuitive as possible, and as concise as possible. Although it is so intuitive, concise and reasonable, this conversion framework can not be adopted by previous works in cryptography, until the formal security proof is completed in this paper. In terms of cryptographic theory, we provide the formal security proof for the intuitive observation on close relation between CLS signatures and CBS signatures.

Many concrete CBS schemes can be constructed from existing CLS schemes by applying the generic framework. One existing CLS scheme is proved secure in our new security model. With this proof as an example, we also listed other CLS schemes which also can be proved secure in the new security model. Hence, many CBS signature schemes can be constructed.

1.4. *Organization*

This paper is organized as follows. Section 2 reviews the syntax definition of CLS and proposes the new security model for CLS. Section 3 briefly reviews the definition and security model for CBS in one more concise framework. Section 4 presents our new generic construction of CBS from CLS and its security proof. Section 5 compares our result with the state-of-the-art one proposed in Wu *et al.* (2012). Section 6 presents a concrete CBS signature scheme as an application example of our generic conversion. Some other CLS schemes which can be similarly proved secure are listed. At last, Section 7 draws the conclusion.

2. Certificateless Signature

We use the notation ID and ID to denote the identity information in the certificateless system and certificate-based system respectively. We put the prefix *CL.* to specify that this is in the certificateless system and *CB* to specify that this is in the certificate-based system. We use the notation

$$query \nrightarrow O$$

to denote that the query *query* has never been submitted to the oracle O, and the notation

$$answer \leftarrow O$$

to denote that the answer *answer* has been returned by the oracle O.

Definition 1. A certificateless signature scheme consists of the following six algorithms.

(1) $\mathsf{CL.Setup}(1^k) \rightarrow (CL.msk, CL.param)$. It takes 1^k as input where k is the security parameter, and returns the master private key $CL.msk$ and the parameter $CL.param$ which is shared in the system.

(2) $\mathsf{CL.ExtractPPK}(CL.msk, CL.param, \mathsf{ID}) \rightarrow D_{\mathsf{ID}}$. It takes the master private key $CL.msk$, the system parameter $CL.param$ and the identity ID as input, and returns the partial private key D_{ID}.

(3) $\mathsf{CL.SetSV}(CL.param) \to X_{\mathsf{ID}}$. It takes as input the system parameter $CL.param$, and outputs a secret value X_{ID}.
(4) $\mathsf{CL.SetPK}(CL.param, X_{\mathsf{ID}}) \to (CL.PK_{\mathsf{ID}})$. It takes as input the system parameter $CL.param$, and this identity's secret value X_{ID}, and outputs the public key $CL.PK_{\mathsf{ID}}$.
(5) $\mathsf{CL.Sign}(CL.param, D_{\mathsf{ID}}, X_{\mathsf{ID}}, m) \to CL.\sigma$. It takes as input the system parameter $CL.param$, the partial private key D_{ID}, the secret value X_{ID} and the message m, and outputs the signature $CL.\sigma$.
(6) $\mathsf{CL.Verify}(CL.param, \mathsf{ID}, CL.PK_{\mathsf{ID}}, m, CL.\sigma) \to b \in \{0, 1\}$. It takes as input the system parameter $CL.param$, an identity ID, this identity's public key $CL.PK_{\mathsf{ID}}$ and a message/signature pair $(m, CL.\sigma)$, and outputs 1 if the signature is correct, or 0 otherwise.

The following definition has some essentially different points from other ones including that in Huang *et al.* (2011). Two remarks on these basic differences will be presented during the definition. To get more succinct security definition than others, we put four kinds of adversary models together in one unified framework. Additionally, in Section 6, we will show that many existing CLS signature schemes can be proved secure in our new security model and present 6 examples.

Definition 2. A CLS scheme is CL-EUF-CMCI secure against a certain kind of adversary

$$\mathcal{A} \in \{\text{CL.Normal-}\mathcal{A}^{I}, \text{CL.Super-}\mathcal{A}^{I}, \text{CL.Normal-}\mathcal{A}^{II}, \text{CL.Super-}\mathcal{A}^{II}\},$$

if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible success probability in the following CLS game. In the following,

$$\text{CL.Normal-}\mathcal{A}^{I},\ \text{CL.Super-}\mathcal{A}^{I},\ \text{CL.Normal-}\mathcal{A}^{II},\ \text{CL.Super-}\mathcal{A}^{II}$$

will be called normal Type I adversary, super Type I adversary, normal Type II adversary, and super Type II adversary respectively.

(1) Initial: the challenger runs the algorithm $\mathsf{CL.Setup}$, returns $CL.Params$ and the auxiliary information $aux \in \{nil, CL.msk\}$ to the attacker $\mathcal{A}$ (nil means nothing), where

$$\begin{aligned} aux &= nil, && \text{for } \mathcal{A} \in \{\text{CL.Normal-}\mathcal{A}^{I}, \text{CL.Super-}\mathcal{A}^{I}\}, \\ aux &= CL.msk, && \text{for } \mathcal{A} \in \{\text{CL.Normal-}\mathcal{A}^{II}, \text{CL.Super-}\mathcal{A}^{II}\}. \end{aligned}$$

(2) Queries: in this phase, $\mathcal{A}$ can adaptively make requests to a few oracles among the following ones.

(i) $O^{CL.CreateU}(\mathsf{ID}) \to CL.PK_{\mathsf{ID}}$. This oracle receives an input ID and outputs this original public key of the identity ID.

(ii) $O^{CL.ReplacePK}(\mathsf{ID}, CL.PK) \to \emptyset$. For a public key replacement query $(\mathsf{ID}, CL.PK)$, it sets $CL.PK$ as the current public key.

(iii) $O^{CL.SecretV}(CL.PK) \to X$. If $CL.PK$ is the original public key of a ceratin identity ID (i.e. $CL.PK$ has been returned from the oracle $O^{CL.CreateU}$), this oracle returns the secret value X corresponding to $CL.PK$. Otherwise, it refuses this query.

(iv) $O^{CL.PartialPK}(\mathsf{ID}, CL.PK) \to D_{\mathsf{ID}}$. For a partial private key query ID, this oracle runs the algorithm $\mathsf{CL.ExtractPPK}$ and outputs the result D_{ID}.

(v) $O^{CL.NSign}(\mathsf{ID}, m) \to CL.\sigma$. If

$$CL.\overline{PK}_{\mathsf{ID}} \leftarrow O^{CL.CreateU},$$

which means that the current public key $CL.\overline{PK}_{\mathsf{ID}}$ has been provided by the oracle $O^{CL.CreateU}$, it outputs a valid signature $CL.\sigma$ of m under the current public key $CL.\overline{PK}_{\mathsf{ID}}$ of the identity ID. Otherwise, it refuses the query.

Remark 1. Originally, "normal" is used to mean that one signing query is normal (or reasonable) for the challenger which knows both the current secret value and partial private key for this query. To make one signing oracle query "normal", all previous definition models require that the current public key be the original one of the current identity. However, this requirement is only sufficient for that the challenger knows the current secret value, but not necessary. In fact, the sufficient and necessary condition for "normal" signing queries should be that the current public key has been generated by the challenger, or formally the oracle $O^{CL.CreateU}$. For example, the adversary gets Alice's public key pk_A and Bob's public key pk_B from the oracle $O^{CL.CreateU}$, then requires Alice's public key to take the value of pk_B through the oracle $O^{CL.ReplacePK}$, and finally queries the signature corresponding to Alice's identity and the replaced (not original) public key. In this example, although Alice's public key has been replaced, the challenger still knows the current secret value and hence this signing query should be called "normal". In fact, our definition for "normal" captures this example, while previous security models leave it out.

(vi) $O^{CL.SSign}(\mathsf{ID}, m) \to CL.\sigma$. For a super signing query (ID, m), it outputs a valid signature $CL.\sigma$ of m under the current public key $CL.\overline{PK}_{\mathsf{ID}}$ of the identity ID.

Every attack model $\mathcal{A}$ has its own set $\mathcal{O}_{\mathcal{A}}$ of allowed oracles. In particular, with $\mathcal{O}' = \{O^{CL.CreateUser}, O^{CL.ReplacePK}, O^{CL.SecretV}\}$ being commonly allowed,

$$\begin{aligned}
&\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \{O^{CL.NSign}, O^{CL.PartialPK}\}, && \text{for } \mathcal{A} = \text{CL.Normal-}\mathcal{A}^{I},\\
&\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \{O^{CL.SSign}, O^{CL.PartialPK}\}, && \text{for } \mathcal{A} = \text{CL.Super-}\mathcal{A}^{I},\\
&\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \{O^{CL.NSign}\}, && \text{for } \mathcal{A} = \text{CL.Normal-}\mathcal{A}^{II},\\
&\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \{O^{CL.SSign}\}, && \text{for } \mathcal{A} = \text{CL.Super-}\mathcal{A}^{II}.
\end{aligned}$$

(3) Output: After all queries, $\mathcal{A}$ outputs a forgery $(\mathsf{ID}^{*}, m^{*}, CL.\sigma^{*})$. Let $CL.\overline{PK}_{\mathsf{ID}^{*}}$ be the current public key of ID^{*}. $\mathcal{A}$ is said to win the game if the forgery satisfies the following restrictions to ensure that the successful forgery is nontrivial. It is well-known that the

basic security requirement for CLS is that both the secret value and the partial private key are the two indispensable factors for generating a CLS signature. In other words, any forgery generated by the attacker who knows at most one of these two indispensable factors should be accepted as successful.

(i) For $\mathcal{A} \in \{\text{CL.Normal-}\mathcal{A}^{I}, \text{CL.Super-}\mathcal{A}^{I}, \text{CL.Normal-}\mathcal{A}^{II}, \text{CL.Super-}\mathcal{A}^{II}\}$, it is commonly required that

$$1 = \mathsf{CL.Verify}\big(CL.mpk, \text{CL.param}, \mathsf{ID}^*, CL.\overline{PK}_{\mathsf{ID}^*}, m^*, CL.\sigma^*\big),$$

and

$$\big(\mathsf{ID}^*, m^*\big) \nrightarrow O^{CL.Sign},$$

where

$$\begin{aligned} O^{CL.Sign} &= O^{CL.NSign}, \quad \text{for } \mathcal{A} \in \big\{\text{CL.Normal-}\mathcal{A}^{I}, \text{CL.Normal-}\mathcal{A}^{II}\big\}, \\ O^{CL.Sign} &= O^{CL.SSign}, \quad \text{for } \mathcal{A} \in \big\{\text{CL.Super-}\mathcal{A}^{I}, \text{CL.Super-}\mathcal{A}^{II}\big\}. \end{aligned}$$

Here, we use the notation $\nrightarrow$ to denote that the query (ID^*, m^*) has never been provided to the oracle O^{Sign}. This restriction ensures that the signature is valid and not trivially obtained from the signing oracle.

(ii) For $\mathcal{A} \in \{\text{CL.Normal-}\mathcal{A}^{I}, \text{CL.Super-}\mathcal{A}^{I}\}$, additionally, it is required that

$$\mathsf{ID}^* \nrightarrow O^{CL.PartialPK}.$$

This restriction ensures that the target partial private key is not known by $\mathcal{A}$.

(iii) For $\mathcal{A} \in \{\text{CL.Normal-}\mathcal{A}^{II}, \text{CL.Super-}\mathcal{A}^{II}\}$, additionally, it is required that

$$CL.\overline{PK}_{\mathsf{ID}^*} \nrightarrow O^{CL.SecretV},$$

and

$$CL.\overline{PK}_{\mathsf{ID}^*} \leftarrow O^{CL.CreateU},$$

where $CL.\overline{PK}_{\mathsf{ID}^*} \leftarrow O^{CL.CreateU}$ means that $CL.\overline{PK}_{\mathsf{ID}^*}$ is provided by the oracle $O^{CL.CreateU}$.

Remark 2. Fundamentally speaking, the original purpose for this restriction is to ensure that the target secret value is not trivially known by Type II adversary. For this purpose, the type II adversary is prohibited from generating the target public key by himself. In fact, previous security definitions require that the target public key must be the original one for the target identity, while our definition only requires that the current public key must come from the challenger, or formally the oracle $O^{CL.CreateU}$. For example, the adversary gets Alice's public key pk_A and Bob's public key pk_B from the oracle $O^{CL.CreateU}$, then requires Alice's public key to take the value of pk_B through the oracle $O^{CL.ReplacePK}$, and finally forges the signature corresponding to Alice's identity and her replaced (not original) public key. In this case, although Alice's public key has been replaced, the adversary still does not know the current secret value and hence this signing forgery should be seen as successful. In fact, our definition formally captures this example, while previous security models leave it out. In fact, this remark for "successful" forgery is somewhat like that for "normal" signing query.

3. Certificate-Based Signature

Definition 3. A Certificate-Based Signature Scheme (CBS) consists of five algorithms as follows.

(i) $\mathsf{CB.Setup}(1^k) \rightarrow (CB.msk, CB.param)$. It takes as input the security parameter 1^k and returns the certifier's master secret key $CB.msk$ and the system parameter $CB.param$ that includes the description of a string space Γ, which can be any subset of $\{0, 1\}^*$.

(ii) $\mathsf{CB.GenUK}(CB.param) \rightarrow (CB.PK_{ID}, CB.SK_{ID})$. It takes input the system parameter $CB.param$, and outputs the secet/public key pair (SK_{ID}, PK_{ID}) for a certain entity ID.

(iii) $\mathsf{CB.Cert}(CB.msk, CB.param, ID, CB.PK_{ID}) \rightarrow cert_{ID}$. It takes as input the master secret key $CB.msk$, the system parameter $CB.param$, the identity ID and its public key $CB.PK_{ID}$, and outputs the certificate $cert_{ID}$.

(iv) $\mathsf{CB.Sign}(CB.param, ID, CB.PK_{ID}, cert_{ID}, CB.SK_{ID}, m) \rightarrow CB.\sigma$. It takes as input the system parameter $CB.param$, the identity ID, the public key $CB.PK$, the certificate $cert_{ID}$, the secret key $CB.SK_{ID}$ and the message m, and outputs the signature $CB.\sigma$.

(v) $\mathsf{CB.Verify}(CB.param, ID, CB.PK_{ID}, m, CB.\sigma) \rightarrow b \in \{0, 1\}$. It takes as input the system parameter $CB.param$, an identity ID, this identity's public key $CB.PK_{ID}$ and a message/signature pair $(m, CB.\sigma)$, and outputs 1 if the signature is correct, or 0 otherwise.

Definition 4. A CBS scheme is CB-EUF-CMCI secure against a certain kind of adversary

$$\mathcal{A} \in \left\{\text{CB.Normal-}\mathcal{A}^{I}, \text{CB.Super-}\mathcal{A}^{I}, \text{CB.Normal-}\mathcal{A}^{II}, CB.Super-\mathcal{A}^{II}\right\},$$

if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible success probability in the following CBS game. In this definition,

$$\text{CB.Normal-}\mathcal{A}^{I},\ \text{CB.Super-}\mathcal{A}^{I},\ \text{CB.Normal-}\mathcal{A}^{II},\ \text{CB.Super-}\mathcal{A}^{II}$$

will be called normal Type I adversary, super Type I adversary, normal Type II adversary, and super Type II adversary respectively.

(1) Initial: the challenger runs the algorithm CB.Setup, returns $CB.Params$ and the auxiliary information $aux \in \{nil, CB.msk\}$ to the attack $\mathcal{A}$ (nil means nothing), where

$$\begin{aligned} &aux = nil, && \text{for } \mathcal{A} \in \left\{\text{CB.Normal-}\mathcal{A}^{I}, \text{CB.Super-}\mathcal{A}^{I}\right\}, \\ &aux = CB.msk, && \text{for } \mathcal{A} \in \left\{\text{CB.Normal-}\mathcal{A}^{II}, \text{CB.Super-}\mathcal{A}^{II}\right\}. \end{aligned}$$

(2) Queries: in this phase, $\mathcal{A}$ can adaptively make requests to a few oracles among the following ones.

(i) $O^{CB.CreateU}(ID) \rightarrow CB.PK_{ID}$. This oracle receives an input ID and outputs this original public key of the identity ID.
(ii) $O^{CB.ReplacePK}(ID, CB.PK) \rightarrow \emptyset$. For a public key replacement query $(ID, CB.PK)$, it sets $CB.PK$ as the current public key of ID.
(iii) $O^{CB.Corrupt}(ID) \rightarrow CB.SK_{ID}$. If ID has been submitted to the oracle $O^{CB.CreateU}$, this oracle returns the secret key $CB.SK_{ID}$ corresponding to ID's original public key. Otherwise, it first makes the oracle query $O^{CB.CreateU}(ID)$ and then the oracle query $O^{CB.Corrupt}(ID)$.
(iv) $O^{CB.Cert}(ID, CB.PK) \rightarrow cert_{ID}$. For this certification query, this oracle gets the result $cert_{ID}$ by running the algorithm CB.Cert and outputs it.
(v) $O^{CB.NSign}(ID, m) \rightarrow CB.\sigma$. If ID has been submitted to $O^{CB.CreateU}$, it outputs a valid signature $CB.\sigma$ of m under the original public key $CB.PK_{ID}$ of the identity ID. Otherwise, it refuses the query.
(vi) $O^{CB.SSign}(ID, m) \rightarrow CB.\sigma$. For a super signing query (ID, m), it outputs a valid signature $CB.\sigma$ of m under the current public key $CB.\overline{PK}_{ID}$ of the identity ID.

Every attack model $\mathcal{A}$ has its own set $\mathcal{O}_{\mathcal{A}}$ of allowed oracles. In particular, with $\mathcal{O}' = \{O^{CB.CreateUser}, O^{CB.ReplacePK}, O^{CB.Corrupt}\}$ being commonly allowed,

$$\begin{aligned} &\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \left\{O^{CB.NSign}, O^{CB.Cert}\right\}, && \text{for } \mathcal{A} = \text{CB.Normal-}\mathcal{A}^{I}, \\ &\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \left\{O^{CB.SSign}, O^{CB.Cert}\right\}, && \text{for } \mathcal{A} = \text{CB.Super-}\mathcal{A}^{I}, \\ &\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \left\{O^{CB.NSign}\right\}, && \text{for } \mathcal{A} = \text{CB.Normal-}\mathcal{A}^{II}, \\ &\mathcal{O}_{\mathcal{A}} = \mathcal{O}' \cup \left\{O^{CB.SSign}\right\}, && \text{for } \mathcal{A} = \text{CB.Super-}\mathcal{A}^{II}. \end{aligned}$$

(3) Output: after all queries, $\mathcal{A}$ outputs a forgery $(ID^*, m^*, CB.\sigma^*)$. Let $CB.\overline{PK}_{ID^*}$ be the current public key of ID^*. $\mathcal{A}$ is said to win the game if this forgery satisfies the following requirements.

(i) For $\mathcal{A} \in \{\text{CB.Normal-}\mathcal{A}^{I}, \text{CB.Super-}\mathcal{A}^{I}, \text{CB.Normal-}\mathcal{A}^{II}, \text{CB.Super-}\mathcal{A}^{II}\}$, it is commonly required that

$$1 = \mathsf{CB.Verify}\left(CB.mpk, CB.param, ID^*, CB.\overline{PK}_{ID^*}, m^*, CB.\sigma^*\right),$$

and

$$\left(ID^*, m^*\right) \nrightarrow O^{Sign},$$

where

$$\mathcal{O}^{C\text{B.Sign}} = \mathcal{O}^{CB.NSign}, \quad \text{for } \mathcal{A} \in \{\text{CB.Normal-}\mathcal{A}^{I}, \text{CB.Normal-}\mathcal{A}^{II}\},$$
$$\mathcal{O}^{C\text{B.Sign}} = \mathcal{O}^{CB.SSign}, \quad \text{for } \mathcal{A} \in \{\text{CB.Super-}\mathcal{A}^{I}, \text{CB.Super-}\mathcal{A}^{II}\}.$$

(ii) For $\mathcal{A} \in \{\text{CB.Normal-}\mathcal{A}^{I}, \text{CB.Super-}\mathcal{A}^{I}\}$, additionally, it is required that

$$ID^* \nrightarrow \mathcal{O}^{CB.Cert}.$$

(iii) For $\mathcal{A} \in \{\text{CB.Normal-}\mathcal{A}^{II}, \text{CB.Super-}\mathcal{A}^{II}\}$, additionally, it is required that

$$ID^* \nrightarrow \mathcal{O}^{CB.Corrupt},$$

and

$$CB.\overline{PK}_{ID^*} = \mathcal{O}^{CB.CreateU}(ID^*).$$

4. Generic Construction CLS-2-CBS and Security Proof

Let Π^{CL} be a CLS scheme

$$\Pi^{CL} = (\mathsf{CL.Setup}, \mathsf{CL.SetSV}, \mathsf{CL.SetPK}, \mathsf{CL.ExtractPPK}, \mathsf{CL.Sign}, \mathsf{CL.Verify}),$$

with algorithms as specified in Definition 1. Then a CBS scheme

$$\Pi^{CL} = (\mathsf{CB.Setup}, \mathsf{CB.GenUK}, \mathsf{CB.Cert}, \mathsf{CB.Sign}, \mathsf{CB.Verify})$$

is defined as follows. Let Γ be the identity information space for Π^{CB}, $\mathcal{PKCB}$ be the public key space for Π^{CB} and $\mathcal{IDCL}$ denotes the space of identities for Π^{CL}. Without loss of generality, we assume that $\mathcal{IDCL} = \Gamma \times \mathcal{PKCB}$.

(1) $\mathsf{CB.Setup}$. On input a security parameter 1^k, first run

$$(CL.msk, CL.param) \leftarrow \mathsf{CL.Setup}(1^k).$$

Then set $CB.msk = CL.msk$. Define $CB.param$ by extending $CL.param$ to include some other relative information. The output is $(CB.msk, CB.param)$.

(2) $\mathsf{CB.GenUK}$. On input $CB.param$, first extract $CL.param$ from $CB.param$. Run

$$X \leftarrow \mathsf{CL.SetSV}(CL.param),$$
$$CL.PK \leftarrow \mathsf{CL.SetPK}(CL.param, X).$$

The output is $(CB.PK, CB.SK) = (CL.PK, X)$.

(3) $\mathsf{CB.Cert}$. On input $CB.msk, CB.param, ID, CB.PK_{ID}$, first extract $CL.param$ from $CB.param$. Set the $\mathsf{ID} = ID||CB.PK_{ID}$ and $CL.msk = CB.msk$. The output is

$$cert_{ID} \leftarrow \mathsf{CL.ExtractPPK}(CL.param, CL.msk, \mathsf{ID}).$$

(4) $\mathsf{CB.Sign}$. On input $CB.param$, $cert_{ID}$, $CB.SK_{ID}$, m, first extract $CL.param$ from $CB.param$. Then set $\mathsf{ID} = ID||CB.PK_{ID}$, $CL.PK_{\mathsf{ID}} = CB.PK_{ID}$, $D_{\mathsf{ID}} = cert_{ID}$, $X_{\mathsf{ID}} = CB.SK_{ID}$. The output is

$$CB.\sigma \leftarrow \mathsf{CL.Sign}(CL.param, \mathsf{ID}, CL.PK_{\mathsf{ID}}, D_{\mathsf{ID}}, X_{\mathsf{ID}}, m).$$

(5) $\mathsf{CB.Verify}$. On input $CB.param, ID, CB.PK_{ID}, m, CB.\sigma$, extract $CL.param$ from $CB.param$. Set $\mathsf{ID} = ID||CB.PK_{ID}$, $CL.PK_{ID} = CB.PK_{ID}$ and $CL.\sigma = CB.\sigma$. The output is

$$b \leftarrow \mathsf{CL.Verify}(CL.param, \mathsf{ID}, CL.PK_{\mathsf{ID}}, m, CL.\sigma).$$

The following four theorems deal with the four kinds of adversaries respectively. As will be seen, all these security reductions are perfectly tight and depend on no random oracles. These two reduction features show that both concepts of CBS and CLS are very closely related to each other.

Theorem 1. *Suppose that $\mathcal{A}^I$ is a super Type I adversary against Π^{CB} with success probability ϵ and running time t. Then there is a super Type I adversary $\mathcal{B}^I$ against Π^{CL} with success probability ϵ and running time $O(t)$.*

Proof. Let $\mathcal{C}$ denote the Π^{CL} challenger against $\mathcal{B}^I$. $\mathcal{B}^I$ mounts a Type I attack on Π^{CL} by simulating the challenger for $\mathcal{A}^I$ and using help from $\mathcal{A}^I$ as follows.

$\mathsf{Initial}$ phase for CBS game. $\mathcal{B}^I$ obtains from $\mathcal{C}$ the system parameter of Π^{CL} and extends it into the system parameter $CB.param$ of Π^{CB} as done in $\mathsf{CB.Setup}$ of Π^{CB}. $\mathcal{B}^I$ supplies $CB.param$ to $\mathcal{A}^I$.

$\mathsf{Queries}$ Phase for CBS game. When $\mathcal{A}^I$ enters the $\mathsf{Queries}$ phase for the CBS game, $\mathcal{B}^I$ accordingly enters the $\mathsf{Queries}$ phase of the CLS game. For the oracle queries from $\mathcal{A}^I$, $\mathcal{B}^I$ handles these queries as follows.

(1) $O^{CB.CreateU}(ID) \rightarrow CB.PK_{ID}$. If the query ID has been submitted to the oracle $O^{CB.CreateU}$, it will directly return the previous answer which is recorded in the list L. Otherwise, it does as follows. $\mathcal{B}^I$ chooses a random identity ID' and obtains its original public key $CL.\widetilde{PK}_{\mathsf{ID}'}$ through the oracle $O^{CL.CreateU}$. It sets $\mathsf{ID} = ID||CB.PK_{\mathsf{ID}'}$ and requires the oracle $O^{CL.ReplacePK}$ to change the public key of ID into $CL.\widetilde{PK}_{\mathsf{ID}'}$. In this case, the CBS original public key of ID is the CLS original public key of ID'. In other words, every original public key for Π^{CB} is related with a certain original public key for Π^{CL}.

Additionally, to record the above operations for future use, $\mathcal{B}^I$ sets the original public key

$$CB.\widetilde{PK}_{ID} = CL.\widetilde{PK}_{\mathsf{ID}'}$$

and the current public key

$$CB.\overline{PK}_{ID} = CL.\widetilde{PK}_{\mathsf{ID}'},$$

and adds the tuple

$$\big(ID, \mathsf{ID}', CB.\widetilde{PK}_{ID}, CB.\overline{PK}_{ID}\big)$$

to the initially empty list L.

(2) $O^{CB.ReplacePK}(ID, CB.PK) \rightarrow \emptyset$. If ID has not been submitted to the oracle $O^{CB.CreateU}$, $\mathcal{B}^I$ first makes the query $O^{CB.CreateU}(ID)$ by himself before the following operations. Otherwise, it directly does the following. $\mathcal{B}^I$ sets $\mathsf{ID} = ID||CB.PK$, and sequentially makes two oracle queries $O^{CL.CreateU}(\mathsf{ID})$ and $O^{CL.ReplacePK}$ $(\mathsf{ID}, CB.PK)$ to change the public key value of ID into $CB.PK$.

Additionally, to record the above operation, $\mathcal{B}^I$ searches the relative tuple

$$\big(ID, \mathsf{ID}', CB.\widetilde{PK}_{ID}, CB.\overline{PK}_{ID}\big),$$

and then changes the value of $CB.\overline{PK}_{ID}$ into $CB.PK$.

(3) $O^{CB.Corrupt}(ID) \rightarrow CB.SK_{ID}$. Without loss of generality, we assume that ID has been submitted to the oracle $O^{CB.CreateU}$. $\mathcal{B}^I$ searches the corresponding tuple

$$\big(ID, \mathsf{ID}', CB.\widetilde{PK}_{ID}, CB.\overline{PK}_{ID}\big)$$

in the list L. Then it makes the oracle query $O^{CL.SecretV}(CB.\widetilde{PK}_{ID})$ and relays this answer to the attacker $\mathcal{A}^I$. Here note $CB.\widetilde{PK}_{ID} = CL.\widetilde{PK}_{\mathsf{ID}'}$.

(4) $O^{CB.Cert}(ID, CB.PK) \rightarrow cert_{ID}$. Without loss of generality, we assume that the current public key $CB.\overline{PK}_{ID} = CB.PK$ in the corresponding tuple

$$\big(ID, \mathsf{ID}', CB.\widetilde{PK}_{ID}, CB.\overline{PK}_{ID}\big)$$

of the list L. $\mathcal{B}^I$ sets $\mathsf{ID} = ID||CB.PK$, makes the oracle query $O^{CL.PartialPK}(\mathsf{ID})$, and then relays the returned partial private key D_{ID} as the certificate for $\mathcal{A}^I$.

(5) $O^{CB.SSign}(ID, m) \rightarrow CB.\sigma$. For a query (ID, m), this oracle browses the list L for the corresponding tuple

$$\big(ID, \mathsf{ID}', CB.\widetilde{PK}_{ID}, CB.\overline{PK}_{ID}\big).$$

Then it sets $\mathsf{ID} = ID||CB.\overline{PK}_{ID}$, makes the signing oracle query $O^{CL.SSign}(\mathsf{ID}, m)$ and relays this answer to $\mathcal{A}^I$.

Output for CBS game. Now the attacker $\mathcal{A}^I$ returns its forgery $(ID^*, m^*, CB.\sigma^*)$. Without loss of generality, we assume that $\mathcal{A}^I$ has made the oracle query $O^{CB.CreateU}$ or the replacing public key oracle query for ID^*. $\mathcal{B}^I$ browses the list L for the the corresponding tuple

$$\big(ID^*, \mathsf{ID}'^*, CB.\widetilde{PK}_{ID^*}, CB.\overline{PK}_{ID^*}\big),$$

and returns $(\mathsf{ID}^*, m^*, CL.\sigma^*)$ to its challenger $\mathcal{C}$, where

$$\mathsf{ID}^* = ID^*||CB.\overline{PK}_{ID^*}, CL.\sigma^* = CB.\sigma^*.$$

Analysis. First, from the relations of Π^{CB} and Π^{CL}, it can be easily or trivially seen that $\mathcal{B}^I$ perfectly simulates the game settings for $\mathcal{A}^I$ in the two phases of Initial and Queries. Second, if the forgery $(ID^*, m^*, CB.\sigma^*)$ is successful, i.e. this forgery satisfies the three additions:

$$\begin{aligned} \mathsf{CB.Verify}\big(CB.param, ID^*, CB.\overline{PK}_{ID^*}, m*, CB.\sigma^*\big) &= 1, \\ \big(ID^*, m^*\big) &\not\twoheadrightarrow \mathcal{O}^{CB.SSign}, \\ \big(ID^*, CB.\overline{PK}_{ID^*}\big) &\not\twoheadrightarrow \mathcal{O}^{CB.Cert}, \end{aligned}$$

then, by checking these two groups of three restrictions one by one (the above and the below), it easily follows that $(\mathsf{ID}^*, m^*, CL.\sigma^*)$ is also successful, i.e. this forgery satisfies that

$$\begin{aligned} \mathsf{CL.Verify}\big(CL.param, \mathsf{ID}^*, CL.\overline{PK}_{\mathsf{ID}^*}, m^*, CL.\sigma^*\big) &= 1, \\ \big(\mathsf{ID}^*, m^*\big) &\not\twoheadrightarrow \mathcal{O}^{CL.SSign}, \\ \mathsf{ID}^* &\not\twoheadrightarrow \mathcal{O}^{CL.PartialPK}. \end{aligned}$$

Hence, the success probability of $\mathcal{B}^I$ is same to that of $\mathcal{A}^I$. Additionally, since what $\mathcal{B}^I$ mainly does in reduction is just issuing some relative queries to $\mathcal{C}$, it is obvious that the time of $\mathcal{B}^I$ is almost equal to the time t of $\mathcal{A}^I$. Hence we say that the running time of $\mathcal{B}^I$ is $O(t)$. □

Theorem 2. *Suppose that $\mathcal{A}^{II}$ is a super Type II adversary against Π^{CB} with success probability ϵ and running time t. Then there is a super Type II adversary $\mathcal{B}^{II}$ against Π^{CL} with success probability ϵ and $O(t)$.*

Proof. Let $\mathcal{C}$ denote a Π^{CL} challenger against Type II adversary $\mathcal{B}^{II}$. $\mathcal{B}^{II}$ mounts a Type II attack on Π^{CL} using help from $\mathcal{A}^{II}$ as follows.

Initial for CBS game. How $\mathcal{B}^{II}$ communicates with $\mathcal{A}^{II}$ is the same to how $\mathcal{B}^I$ communicates with $\mathcal{A}^I$ in the above proof for Theorem 1, except that $\mathcal{B}^{II}$ additionally gets the master private key $CL.msk$ and relays it to $\mathcal{A}^{II}$ as the master key $CB.msk$ for Π^{CB}.

Queries for CBS game. How $\mathcal{B}^{II}$ answers these queries from $\mathcal{A}^{II}$ is the same to how $\mathcal{B}^I$ simulates the oracles for $\mathcal{A}^I$ in the above proof for Theorem 1, except that $\mathcal{A}^{II}$ does not query the oracle $\mathcal{O}^{CB.Cert}$.

Output for CBS game. How $\mathcal{B}^{II}$ generates the forged signature is the same to how $\mathcal{B}^I$ generates the forged signature in the above proof for Theorem 1. Of course, the forged signatures from $\mathcal{B}^I$ and $\mathcal{B}^{II}$ satisfy different conditions.

Analysis. This analysis can immediately follow from the analysis in the above proof of Theorem 1. Here, we only deal with the somewhat difficult part of the analysis. In particular, we will show how to get

$$CL.\overline{PK}_{\mathsf{ID}^*} \nrightarrow O^{CL.SecretV}, \quad \text{and} \quad CL.\overline{PK}_{\mathsf{ID}^*} \leftarrow O^{CL.CreateU},$$

from

$$ID^* \nrightarrow O^{CB.Corrupt}, \quad \text{and} \quad CB.\overline{PK}_{ID^*} = O^{CB.CreateU}\big(ID^*\big),$$

where

$$\mathsf{ID}^* = ID^*||CB.\overline{PK}_{ID^*}, \qquad CL.\sigma^* = CB.\sigma^*.$$

By checking the simulation of the oracle $O^{CB.CreateU}$ by $\mathcal{B}^{II}$ for $\mathcal{A}^{II}$ (same to those in the proof of Theorem 1), the equation $CB.\overline{PK}_{ID^*} = O^{CB.CreateU}(ID^*)$ means that the original public key $CB.\widetilde{PK}_{ID^*}$ is equal to the current public key $CB.\overline{PK}_{ID^*}$, i.e. that

$$CB.\widetilde{PK}_{ID^*} = CB.\overline{PK}_{ID^*}.$$

By checking the simulation of oracles $O^{CB.CreateU}$ and $O^{CB.ReplacePK}$, it can be seen that the CLS current public key $CL.\overline{PK}_{\mathsf{ID}}$ of $\mathsf{ID} = ID||CB.\overline{PK}_{ID}$ is always equal to the CBS current public key $CB.\overline{PK}_{ID}$ of ID. In particular, for $\mathsf{ID}^* = ID^*||CB.\overline{PK}_{ID^*}$, we have

$$CB.\overline{PK}_{ID^*} = CL.\overline{PK}_{\mathsf{ID}^*}.$$

Let the tuple $(ID^*, \mathsf{ID}'^*, CB.\widetilde{PK}_{ID^*}, CB.\overline{PK}_{ID^*})$ be the corresponding record in L. Then by checking the simulation of the oracle $O^{CB.CreateU}$ again, it can be seen that the CBS original public key $CB.\widetilde{PK}_{ID^*}$ is equal to the CLS original public key $CL.\widetilde{PK}_{\mathsf{ID}'^*}$ of ID'^*, i.e. that

$$CB.\widetilde{PK}_{ID^*} = CL.\widetilde{PK}_{\mathsf{ID}'^*}.$$

By the above three equations, we can get

$$CL.\overline{PK}_{\mathsf{ID}^*} = CL.\widetilde{PK}_{\mathsf{ID}'^*}.$$

Hence, from

$$CB.\overline{PK}_{ID^*} = O^{CB.CreateU}\big(ID^*\big),$$

we can get that

$$CL.\overline{PK}_{\mathsf{ID}^*} \leftarrow O^{CL.CreateU}.$$

By checking the oracle simulation process provided by $\mathcal{B}^{II}$ to $\mathcal{A}^{II}$ (same to those in the proof of Theorem 1), it can be seen that $O^{CL.SecretV}(CL.\widetilde{PK}_{\mathsf{ID}'^*})$ is queried by $\mathcal{B}^{II}$ only when $O^{CB.Corrupt}(ID^*)$ is queried by $\mathcal{A}^{II}$. Then by the just proved equation $CL.\overline{PK}_{\mathsf{ID}^*} = CL.\widetilde{PK}_{\mathsf{ID}'^*}$, we prove that $O^{CL.SecretV}$ $(CL.\overline{PK}_{\mathsf{ID}^*})$ is queried by $\mathcal{B}^{II}$ only when $O^{CB.Corrupt}(ID^*)$ is queried by $\mathcal{A}^{II}$. In other words, from

$$ID^* \nrightarrow O^{CB.Corrupt}$$

we can get

$$CL.\overline{PK}_{\mathsf{ID}^*} \nrightarrow O^{CL.SecretV}.$$ □

Theorem 3. *Suppose that $\mathcal{A}^I$ is a normal Type I adversary against Π^{CB} with success probability ϵ and running time t. Then there is a normal Type I adversary $\mathcal{B}^I$ against Π^{CL} with success probability ϵ and running time $O(t)$.*

Proof. The proof for Theorem 3 is the same to that for Theorem 1, except the difference that the CBS normal signing oracle is simulated depending on the CLS normal signing oracle in this proof for Theorem 3, while the CBS super signing oracle is simulated depending on the CLS super signing oracle in this proof for Theorem 1. Now show how to simulate the CBS normal signing oracle using the CLS normal signing oracle.

For a normal signing query $O^{CB.NSign}(ID, m)$, browse the list L for the corresponding tuple

$$\big(ID, \mathsf{ID}', CB.\widetilde{PK}_{ID}, CB.\overline{PK}_{ID}\big).$$

Set $\mathsf{ID} = ID||CB.\overline{PK}_{ID}$ and make the signing oracle query $O^{CL.NSign}(\mathsf{ID}, m)$ and relay this answer to $\mathcal{A}^I$. By checking the simulation of the oracle $O^{CB.CreateU}$, it can be seen that the original CBS public key $CB.\widetilde{PK}_{ID}$ of ID is the original CLS public key $CL.\widetilde{PK}_{\mathsf{ID}'}$ of ID', and the original CLS public key $CL.\widetilde{PK}_{\mathsf{ID}'}$ of ID' is the current CLS public key $CL.\overline{PK}_{\mathsf{ID}}$ of ID. Hence, the query (ID, m) will not be rejected by the oracle $O^{CL.NSign}$. Hence, this proof can immediately follow Theorem 1. □

Theorem 4. *Suppose that $\mathcal{A}^I$ is a normal Type II adversary against Π^{CB} with success probability ϵ and running time t. Then there is a normal Type II adversary $\mathcal{B}^I$ against Π^{CL} with success probability ϵ and running time $O(t)$.*

Proof. The proof for Theorem 4 is the same to that for Theorem 2, except for the difference that the CBS normal signing oracle is simulated depending on the CLS normal signing oracle in this proof for Theorem 4, while the CBS super signing oracle is simulated depending on the CLS super signing oracle in this proof for Theorem 2. Additionally, in the proof of Theorem 3, we have showed how to simulate the CBS normal signing oracle by using the CLS normal signing oracle. Hence, this proof can immediately follow Theorem 2 and Theorem 3. □

5. Comparison Between CLS-2-CBS with Wu *et al.*'s Result

(1) Our generic construction is very similar to that of Wu *et al.* (2009). In particular, if we replace

$$\mathsf{ID} = ID||CB.PK_{ID}$$

with

$$\mathsf{ID} = \mathrm{H}(ID||\mathrm{CB.PK}_{ID}), \quad \text{where } H \text{ is a hash function,}$$

then our construction will become identical with Wu *et al.*'s (2009) construction.

(2) Our security proof is in the standard model. Contrastingly, the security proof in Wu *et al.* (2009) relies the assumption that the hash function in $\mathsf{ID} = \mathrm{H}(ID||CB.PK_{ID})$ is a random oracle. As a result, our method can convert a CLS scheme secure in the standard model into a CBS scheme secure in the standard model. However, through Wu *et al.*'s method, the resulting CBS scheme is only secure in the random oracle model. Of course, the standard model is preferable over the random oracle model in security proof Canetti *et al.* (2004).

(3) We emphasize the theoretical significance of our generic construction. Roughly speaking, by our result, the cryptographic primitive of CBS can be obtained from the single primitive of CLS. However, by Wu *et al.*'s result, the primitive of CBS can not be obtained from the single primitive of CLS, since the random oracle is additionally needed.

(4) To support our generic conversion, the new CLS security model is developed by introducing one nontrivial attack never mentioned before. This new definition succeeds in more elaborately capturing the security notion of CLS. Notably, this new security is inherently preserved in many existing CLS schemes. In other words, our contribution in this respect is that we point out, formalize, and apply this subtle property to refine the close relation between CLS and CBS.

6. Application Example – One Concrete CBS scheme

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q and let P be a generator of $\mathbb{G}_1$, where $\mathbb{G}_1$ is additively represented and $\mathbb{G}_2$ is multiplicatively represented. A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is said to be a bilinear pairing, if the following three conditions hold: (1) e is bilinear, i.e. $e(aP, bP) = e(P, P)^{ab}$ for all $a, b \in \mathbb{Z}_q^*$; (2) e is non-degenerate, i.e. $e(P, P) \neq 1$, where 1 is the identity of $\mathbb{G}_2$; (3) e is efficiently computable.

The CDH problem: given P, aP, bP with uniformly random choices of $a, b \in \mathbb{Z}_q$, output abP. An algorithm $\mathcal{A}$ has success probability ϵ in solving the CDH problem, if

$$Pr\big[\mathcal{A}(P, aP, bP) = abP\big] = \epsilon].$$

The CDH problem is said to be (t, ϵ)-intractable if there is no algorithm to solve this problem with time less than t and success probability greater than ϵ.

The following scheme is constructed from the second CLS scheme in Huang *et al.* (2011) through the generic conversion *CLS*-2-*CBS*.

(i) $\mathsf{Setup}(1^k) \rightarrow (msk, params)$. This algorithm is run by the authority CA. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q, P be a generator of $\mathbb{G}_1$, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear pairing. It specifies two hash functions $H_0, H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$. Let Γ be the set of identity information. It chooses the master key $msk = s$ uniformly at random from $\mathbb{Z}_q$ and computes the public key $mpk = sP$. The system parameter is

$$params = (q, e, \mathbb{G}_1, \mathbb{G}_2, P, mpk, H_0, H_1, \Gamma).$$

(ii) $\mathsf{GenUK}(params) \rightarrow (PK_{ID}, SK_{ID})$. It selects a random $SK_{ID} \in Z_q$, computes $PK_{ID} = SK_{ID}P$ and outputs SK_{ID}, PK_{ID} as *ID*'s secret/public key pair.

(iii) $\mathsf{Cert}(CB.msk, params, ID, PK_{ID}) \rightarrow cert$. It sets $cert_{ID} = sH_0(ID||PK_{ID})$.

(iv) $\mathsf{Sign}(params, ID, cert_{ID}, PK_{ID}, SK_{ID}, m) \rightarrow \sigma$. For a message m, the user *ID* computes the signature $\sigma = (u, v, W)$, where

- $u = H_1(m||ID||PK_{ID}||r_1P||e(P, P)^{r_2})$ for random $r_1, r_2 \in Z_q$, which are chosen by *ID*;
- $v = r_1 - u \cdot SK_{ID} \bmod q$, $W = r_2P - u \cdot cert_{ID}$.

(v) $\mathsf{Verify}(params, ID, PK_{ID}, m, \sigma) \rightarrow b$. Given a message/signature pair $(m, \sigma = (u, v, W))$, user *ID*'s public key PK_{ID}, anyone can check whether

$$u = H_1\big(m||ID||PK_{ID}||vP + u \cdot PK_{ID}||e(W, P)e\big(H_0(ID||PK_{ID}), sP\big)^u\big).$$

If the equality holds, output 1; otherwise, output 0.

Lemma 1. *The second certificateless scheme in Huang* et al. (2011) *is CL-EUF-CMCI secure against a super Type I adversary.*

Proof. The proof directly follows the observation that (1) the security definition against a super Type I adversary in Huang *et al.* (2011) is essentially the same as that of ours, except some notation differences. □

Lemma 2. *The second certificateless scheme in Huang* et al. (2011) *is CL-EUF-CMCI secure against a super Type II adversary.*

Proof. First, compare the security definition against a super Type II adversary in Huang *et al.* (2011) and that of ours.

(1) Observe the difference: Huang's definition in Huang *et al.* (2011) requires that the attacked identity's public key be the original one, while our definition allows

that the attacked identity's public key can be the replaced one under the constraint condition that this public key must be the other identity's original public key. In other words, for our definition, the attacker can replace Alice's public key with Bob's original public key, and then attack Alice with this replaced public key.

(2) Observe the common purpose of these restrictions: in both definitions, the super type II adversary does not know the attacked identity's current secret value. All in all, facing the same purpose of keeping the target secret value secret from the adversary, our definition tries to make less restrictions while the restriction in Huang's definition goes too far.

Next revisit the security proof (for Theorem 4.4) in Huang *et al.* (2011):

(3) Observe the key point of probability analysis (for Theorem 4.4 in Huang *et al.* (2011)): only if the target public key comes from the challenger and the corresponding secret value is never queried, this security proof always conducts well. In other words, for the proof of Huang *et al.* (2011), what matters is not that the target public key is "original", but that the corresponding secret key is not known by this adversary.

At last, following the above observations, we can easily obtain the security proof for Lemma 2, by slightly adapting the security proof for Theorem 4.4 in Huang *et al.* (2011): in addition to some trivial modifications, change (1) the condition equation "$ID^* = ID_\pi$" into "ID^*'s current public key $= ID_\pi$'s original public key" and (2) "$\mathcal{A}_{II}$ is not allowed to replace this user's public key" into "$\mathcal{A}_{II}$ is not allowed to query the secret value corresponding to this user's current public key". Hence, we can omit the detailed proof. □

Theorem 5. *The above CBS scheme is secure (in the random oracle model) against CB.Super-$\mathcal{A}^I$ and CB.Super-$\mathcal{A}^{II}$, assuming that CDH problem is hard in $\mathbb{G}_1$.*

Proof. We can obtain the proof immediately from Theorems 1, 2 for the security of the generic construction and Lemmas 1, 2 for the security of the underlying CLS scheme. □

Remark 3. Just as the proof of Lemmas 1 and 2, we can similarly prove that many existing certificateless signature schemes are secure in our new security model, by slightly modifying their original proof. In fact, by revisiting the two remarks during Definition 2 in Section 2 and the three observations in Lemma 2, we can conclude that how to get these proofs is almost trivial. In other words, for a normal signing query, the common basic reason why both the original proof and the modified proof conduct well is not that the public key has been replaced, but that the current secret value is not known by the challenger; for a signature forgery, the common basic reason why both the original proof and the modified proof conducts is not that the public key has been replaced, but that the current secret value is not known by the adversary. Hence, these almost trivial modifications and the proof in our new security model for existing CLS schemes can be omitted here. Some examples among them are as follows.

(1) The first certificateless scheme in Huang *et al.* (2011) is CL-EUF-CMCI secure against a normal Type I adversary and a super Type II adversary;
(2) The certificateless scheme in Huang *et al.* (2005) is CL-EUF-CMCI secure against a normal Type I adversary and a normal Type II adversary;
(3) The certificateless scheme in Choi *et al.* (2011) is CL-EUF-CMCI secure against a super Type I adversary and a super Type II adversary;
(4) The certificateless scheme in Zhang *et al.* (2007) is CL-EUF-CMCI secure against a normal Type I adversary and a normal Type II adversary.
(5) The certificateless scheme based on RSA in Zhang and Mao (2012) is CL-EUF-CMCI secure against a normal Type I adversary and a normal Type II adversary.
(6) The certificateless scheme in Pang *et al.* (2015) is CL-EUF-CMCI secure against a normal Type I adversary and a normal Type II adversary.

All the above certificateless signature schemes can be used to construct certificate-based signature schemes through the generic framework. For example, based on the state-of-the-art CLS signature scheme secure in the standard model (Pang *et al.*, 2015), we can construct one CBS scheme in the standard model which is almost the same to the the state-of-the-art CBS scheme recently proposed by Lu and Li (2015). Hence, we omit the trivial scheme description. In particular, both of them can be seen as the extension of the famous Waters signature scheme (Waters, 2005).

Now we simply present the efficiency comparisons between the Lu-Li CBS scheme and our CBS scheme generically constructed from Pang *et al.*'s CLS scheme. First, because both of them are constructed based on the Waters signature scheme, there is almost no difference in the first three algorithms for setup, user key generation and the certification generation of both CBS schemes, except some notational differences. Second, for the signing algorithms, both CBS schemes have the same signature size, i.e. 3 group elements. To generate a signature, our CBS scheme needs 7 exponentiations in the bilinear group, while the Lu-Li CBS scheme needs 6 exponentiations. Third, to verify a signature, both CBS schemes need 3 pairing computations. Here note that group exponentiations and parings are the main computation for generating and verifying a signature respectively. Hence, we can see that the CBS scheme constructed using our generic framework in the standard model is almost as efficient as the Lu-Li CBS scheme in terms of signature size and computation complexity.

7. Conclusion

In this paper, we proposed a new provably secure generic conversion from CLS to CBS. To analyse its security, we redefined the security model for CLS by formalizing some important but previously ignored properties. This new security definition is the main reason why our new conversion can be provably secure in the standard model. As an example, we constructed a new provably secure certificate-based signature scheme by applying this new generic method. These results formally showed that the two conceptions of CBS and CLS (CBC and CLC) are very closely related to each other.

Acknowledgements. This work is partially supported by National Natural Science Foundation of China (61202475, 61472114 and 61133014), Shandong Province Statistics Key Project (KT16022), the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD, Nanjing University of Information Science & Technology) and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET, Nanjing University of Information Science & Technology).

References

Al-Riyami, S.S., Paterson, K.G. (2003). Certificateless public key cryptography. In: *Proceedings of ASIACRYPT 2003*, *Lecture Notes in Computer Science*, Vol. 2894. Springer, Berlin, pp. 452–473.

Al-Riyami, S.S., Paterson, K.G. (2005). CBE from CLE: a generic construction and effiient schemes. In: *Proceedings of PKC 2005*, *Lecture Notes in Computer Science*, Vol. 3386. Springer, Berlin, pp. 398–415.

Bellare, M., Rogaway, P. (1993). Random oracles are practical: a paradigm for designing efficient protocols. In: *Proceedings of 1st ACM Conference on Computer and Communications Security*. ACM Press, pp. 62–73.

Canetti, R., Goldreich, O., Halevi, S. (2004). The random oracle methodology, revisited. *Journal of ACM*, 51(4), 557–594.

Choi, K.Y., Park, J.H., Lee, D.H. (2011). A new provably secure certificateless short signature scheme. *Computers & Mathematics with Applications*, 61(7), 1760–1768.

Gentry, C. (2003). Certificate-based signature and the certificate revocation problem. In: *Proeedings of EUROCRYPT 2003*, *Lecture Notes in Computer Science*, Vol. 2656. Springer, Berlin, 272–293.

Guo, P., Wang, J., Li, B., Lee, S. (2014). A variable threshold-value authentication architecture for wireless mesh networks. *Journal of Internet Technology*, 15(16), 929–936.

Huang, X., Susilo, W., Mu, Y., Zhang, F. (2005). On the security of certificateless signature schemes from asiacrypt 2003. In: *Proceedings of Cryptology and Network Security 2005*, *Lecture Notes in Computer Science*, Vol. 3810. Springer, Berlin, pp. 13–25.

Huang, X., Mu, Y., Susilo, W., Wong, D., Wu, W. (2011). Certificateless signatures: new schemes and security models. *The Computer Journal*, 55(4), 457–474.

Kang, B.G., Park, J.H. (2005). Is it possible to have CBE from CLE? Iacr cryptology print archive, 2005, available at eprint.iacr.org/2005/431.ps.

Kang, B.G., Park, J.H., Hahn, S.G. (2004). A certificate-based signature scheme. In: *Proceedings of CT-RSA 2004*, *Lecture Notes in Computer Science*, Vol. 2964. Springer, Berlin, pp. 99–111.

Li, J., Huang, X., Mu, Y., Susilo, W., Wu, Q. (2010). Constructions of certificate-based signature secure against key replacement attacks. *Journal of Computer Securit*y, 18(3), 421–449.

Li, J., Huang, X., Zhang, Y., Xu, L. (2012). An efficient short certificate-based signature scheme. *Journal of Systems and Software*, 85(2), 314–322.

Liu, J.K., Au, M.H., Susilo, W. (2007). Self-generated certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In: *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*. ACM Press, 273–283.

Liu, J.K., Baek, J., Susilo, W., Zhou, J. (2008). Certificate-based signature schemes without pairings or random oracles. In: *Proceedings of Information Security Conference 2008*, *Lecture Notes in Computer Science*, Vol. 5222. Springer, Berlin, pp. 285–297.

Liu, J.K., Bao, F., Zhou, J. (2011). Short and efficient certificate-based signature. In: *Proceedings of Networking Workshops 2011*, *Lecture Notes in Computer Science*, Vol. 2867. Springer, Berlin, pp. 167–178.

Lu, Y., Li, J. (2015). Improved certificate-based signature scheme without random oracles. *IET Information Security*, 10(2), 80–86.

Pang, L., Hu, Y., Liu, Y., Xu, K., Li, H. (2015). Efficient and secure certificateless signature scheme in the standard model. *International Journal of Communication Systems*. Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/dac.3041.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In: *Proceedings of Crypto 1984*, *Lecture Notes in Computer Science*, Vol. 196. Springer, Berlin, pp. 47–53.

Shen, J., Tan, H., Wang, J., Wang, J.W., Lee, S. (2015). A novel routing protocol providing good transmission reliability in underwater sensor networks. *Journal of Internet Technology*, 16(1), 171–178.

Waters, B. (2005). Efficient identity-based encryption without random oracles. In: *Proceedings of EUROCRYPT 2005*, *Lecture Notes in Computer Science*, Vol. 3494. Springer, Berlin, pp. 114–127.

Wu, W., Mu, Y., Susilo, W., Huang, X. (2009). Certificate-based signatures revisited. *Journal of Universal Computer Science*, 15 (8), 1659–1684.

Wu, W., Mu, Y., Susilo, W., Huang, X. (2012). Provably secure construction of certificate-based encryption from certificateless encryption. *The Computer Journal*, 55(10), 1157–1168.

Xie, S., Wang, Y. (2014). Construction of tree network with limited delivery latency in homogeneous wireless wensor networks. *Wireless Personal Communications*, 78(1), 231–246.

Zhang, J., Mao, J. (2012). An efficient RSA-based certificateless signature scheme. *The Journal of Systems and Software*, 85(3), 638–642.

Zhang, L., Zhang, F., Zhang, F. (2007). New efficient certificateless signature scheme. In: *Proceedings of EUC 2007*, *Lecture Notes in Computer Science*, Vol. 4809. Springer, Berlin, pp. 692–703.

W. Gao received his PhD and MS degrees in applied mathematics from Hunan University in 2006, Guangzhou University 2003, respectively. He is an associate professor in Ludong University from 2012.

G. Wang received his PhD degree in computer science, from Institute of Software, Chinese Academy of Sciences, PR China, in 2001. He was a senior lecturer in University of Wollongong, Australia. Now he works in Huawei Technologies Co. Ltd., Singapore. His research interests include cryptography and information security.

K. Chen is a professor of cryptography and information security school of science, Hangzhou Normal University since 2013. From 1996 to 2013, he was a professor of cryptography and information security in the School of Science, Shanghai Jiaotong University. His interest fields are public key cryptography, cryptographic protocol analysis, applied cryptographic techniques and computer security.

X. Wang received his PhD degree in mathematics from the Academy of China in 1991, his MS degree in mathematics from Shannxi Normal University in 1987. He is currently a professor of Computer Science at South China Normal University. His current research interests include cryptography, number theory.