Information Sciences 296 (2015) 315–321

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Security models for certificateless signature schemes revisited

Kyung-Ah Shim

*National Institute for Mathematical Sciences, KT Daedoek 2nd Research Center, 463-1 Jeonmin-dong, Yuseong-gu, Daejeon, Republic of Korea*

ARTICLE INFO

*Article history:*
Received 21 May 2008
Received in revised form 5 June 2014
Accepted 24 October 2014
Available online 11 November 2014

*Keywords:*
Digital signature
Identity-based system
Certificateless signature
Key replacement attack
Formal security model

ABSTRACT

Certificateless cryptography eliminates the need of certificates in the Public Key Infrastructure and solves the inherent key escrow problem in the ID-based cryptography. In this paper, we point out security pitfalls on the restrictions of an adversary's final output in security models of certificateless signature schemes by demonstrating key replacement attacks on three certificateless signature schemes in the different security models.



## 1. Introduction

Public-key cryptographys (PKCs) need authentication of users' public keys. Public Key Infrastructure (PKI) is an arrangement that binds public keys with respective user identities by means of public key certificates issued by a certificate authority (CA). The public key certificate is an electronic document which contains the CA's signature to bind a public key with an identity information. The certificate can be used to verify that a public key belongs to an individual. This PKI causes several problems of certificate management including storage, distribution and the computational cost of certificate verification. Identity (ID)-based infrastructure [12] allows a user's public key to be easily derived from its known identity information by eliminating the need for public-key certificates. Such cryptosystems alleviate certificate overhead and solve the problems of PKI technology. In ID-based infrastructure, a Private Key Generator (PKG) having a master public/secret key pair is responsible for generating private keys for users. This feature leads to an inherent key escrow problem: users' private keys are known to the PKG, therefore, it can decrypt any ciphertext and forge signatures on any message for any user. Al-Riyami and Paterson [1] introduced certificateless public key cryptography (CL-PKC) to solve the key-escrow problem. In CL-PKC, a user private key is a combination of some contribution of a KGC (called a partial private key) and some user-chosen secret, in such a way that the problem can be solved. CL-PKC is not purely ID-based, as a signature and a ciphertext are transmitted together with an additional user public key that is not required to be certified by any trusted authority. In order to verify a signed message, one must know both the user's identity and this additional public key.

Al-Riyami and Paterson [1] proposed a certificateless public-key encryption (CLE) scheme and a certificateless signature (CLS) scheme. Although the security model for CLE schemes was established in [1], while unforgeability of CLS schemes was not formally defined. Since Al-Riyami and Paterson's CLS scheme, several CLS schemes have been proposed [10,5,14]. They provided only informal analysis and were subsequently found to be vulnerable to key replacement attacks by type I adversaries [16,3,2]. Later, several formal security models have been proposed by presenting proven secure CLS schemes in the

*E-mail address:* kashim@nims.re.kr

http://dx.doi.org/10.1016/j.ins.2014.10.055
0020-0255/

model assuming random oracles [9,17,4]. Liu et al. [11] proposed the first provably secure CLS scheme in the standard model, based on Water's scheme [13]. In addition to these direct constructions, there exist generic constructions that convert existing signature schemes in different infrastructures into CLS schemes. Yum and Lee [15] proposed a generic construction for CLS schemes by combining any standard signature (SS) scheme with any ID-based signature (IBS) scheme. Subsequently, Hu et al. [6] showed that this construction is insecure against key replacement attacks and then proposed its improved version by modifying the input of signing algorithm. In particular, Hu et al. [7] established a simplified definition and formal security models for CLS schemes which are shown to be more versatile than previous ones [9,17]. Au et al. [2] suggested a malicious-but-passive-KGC attack in which a KGC may not generate master public/secret key pair honestly to mount the attack, and they modified Hu et al.'s model for capturing the attack. They also showed that Al-Riyami and Paterson's scheme and its variants [1,9,10] are insecure against the malicious-but-passive-KGC attacks and the security of the CLS scheme converted from the modified Yum–Lee's construction is preserved in their new model. In summary, there are two types of adversaries In CLS schemes: a type I adversary represents a malicious third party who can replace a user public key, called a key replacement attack, and a type II adversary is a malicious KGC who knows the master secret, but cannot replace user public keys. Existence of type I adversary is due to the uncertified feature of a user public key and considering type II adversary is for solving the key escrow problem, i.e., disclosure of KGC's master secret does not compromise the secret of each user. In malicious-but-passive-KGC attacks, a type II adversary, KGC, is passive, in the sense that the KGC would not actively replace user public keys, and can generate a mater secret/public key pair to attack a target user. In this paper, we discuss differences between formal security models of CLS schemes in [17,2,7,8] and then point out security flaws on the restrictions of an adversary's final output in the security models by demonstrating key replacement attacks on three CLS schemes [10,4,11].

The rest of the paper is organized as follows. In Section 2, we describe a definition of CLS schemes and their formal security models proposed in [17]. We discuss differences in the security models in [17,2,8,7] in Section 3. In Section 4, we point out vulnerabilities of Liu et al.'s scheme, Choi et al.'s scheme and Li–Chen's scheme against key replacement attacks in Zhang et al.'s security model [17]. We then suggest improvements. Concluding remarks are given in Section 5.

## 2. Formal security models for CLS schemes

We briefly describe a definition for CLS schemes and their formal security models in [17].

COMPONENTS OF CERTIFICATELESS SIGNATURE SCHEMES. A CLS scheme $\mathcal{CLS} = (\textbf{Setup}, \textbf{Partial} - \textbf{Private} - \textbf{Key} - \textbf{Extract}, \textbf{Set} - \textbf{Secret} - \textbf{Value}, \textbf{Set} - \textbf{Private} - \textbf{Key}, \textbf{Set} - \textbf{Public} - \textbf{Key}, \textbf{CL} - \textbf{Sign}, \textbf{CL} - \textbf{Verify})$ is specified by six polynomial time algorithms with the following functionality:

- **Setup.** It takes as input a security parameter $k$, and returns a list `params` of system parameters and a master public/secret key pair (*mpk,msk*). The algorithm is assumed to be run by a KGC for the initial setup of a certificateless system.
- **Partial-Private-Key-Extract.** It takes as inputs `params`, a master secret key *msk* and a user identity $ID \in \{0,1\}^*$, and outputs a partial private key $D_{ID}$. This algorithm is run by the KGC once for each user, and the partial private key generated is assumed to be distributed securely to the corresponding user.
- **Set-Secret-Value.** Taking as inputs params and a user's identity *ID*, this algorithm generates a secret value $S_{ID}$. This algorithm is supposed to be run by each user in the system.
- **Set-Private-Key.** This algorithm takes params, a user's partial private key $D_{ID}$ and his secret value $S_{ID}$, and outputs a full private key $SK_{ID}$. This algorithm is run by each user.
- **Set-Public-Key.** It takes as inputs `params` and a user's secret value $S_{ID}$, and generates a public key $PK_{ID}$ for that user. This algorithm is run by the user, and the resulting public key is assumed to be publicly known.
- **CL-Sign.** This algorithm takes as inputs `params`, a message $m \in \{0,1\}^*$, a user's identity *ID*, and the user's full private key $SK_{ID}$, and outputs a signature $\sigma$.
- **CL-Verify.** This algorithm takes as inputs `params`, a public key $PK_{ID}$, a message $m$, a user's identity *ID*, and a signature $\sigma$, and returns a bit $b$. $b = 1$ means that the signature is accepted, whereas $b = 0$ means rejected.

There are two types of adversaries, $\mathcal{A}^I$ and $\mathcal{A}^{II}$ in CLS schemes. The type I adversary $\mathcal{A}^I$ is a malicious third party who compromises a user secret key or replaces a user public key, while $\mathcal{A}^I$ is given neither a master secret key *msk* nor a partial private key. The type II adversary $\mathcal{A}^{II}$ is a malicious KGC, who knows a master secret *msk* and hence can derive the value of any user's partial private key, while it can neither access to a user public key nor replace a user secret key.

UNFORGEABILITY OF CLS SCHEMES. Let $\mathcal{CLS}$ be a CLS scheme. We consider two games Game I and Game II where $\mathcal{A}^I$ and $\mathcal{A}^{II}$ interact with their challenger in these two games, respectively.

**[Game I]**. This is the game in which $\mathcal{A}^I$ interacts with the challenger.

- **Phase I-1:** The challenger runs $\texttt{Setup}(1^k)$ for generating (*mpk,msk*) and `params`. The challenger then gives `params` and *mpk* to $\mathcal{A}^I$ while keeping *msk*.
- **Phase I-2:** $\mathcal{A}^I$ performs the following oracle-query operations:
  - `Extract Partial Private Key Queries`: On receiving such a query, the challenger computes $D_{ID}$=`Partial-Private-Key-Extract(params,`*msk*, *ID*) and returns it to $\mathcal{A}^I$.