Information Sciences 268 (2014) 458–462

Contents lists available at ScienceDirect

Information Sciences

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

Insecurity of an efficient certificateless aggregate signature with constant pairing computations

Debiao He [a,*], Miaomiao Tian [b], Jianhua Chen [a]

[a] School of Mathematics and Statistics, Wuhan University, Wuhan, China
[b] School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

ARTICLE INFO

Article history:
Received 16 March 2013
Received in revised form 5 September 2013
Accepted 15 September 2013
Available online 21 September 2013

Keywords:
Certificateless cryptography
Aggregate signature
Bilinear pairing

ABSTRACT

Recently, Xiong et al. [H. Xiong, Z. Guan, Z. Chen, F. Li, An efficient certificateless aggregate signature with constant pairing computations, Information Science 219 (2013) 225–235] proposed a certificateless signature (CLS) scheme and used it to construct a certificateless aggregate signature (CLAS) scheme with constant pairing computations. They demonstrated that both of their schemes are provably secure in the random oracle model under the computational Diffie–Hellman assumption. Unfortunately, by giving concrete attack, we demonstrate that their schemes are not secure against the Type II adversary, i.e. a Type II adversary could forge a legal signature of any message.

1. Introduction

An aggregate signature scheme, which was proposed by Boneh et al. [2], is a variant of the signature scheme which allows to aggregate n signatures on n distinct messages from n distinct users into a single signature. The legality of an aggregate signature will convince a verifier confirm that the n users really sign the n original messages separately. Aggregation signature scheme is useful to reduce bandwidth and storage, and is especially attractive for mobile devices like sensors, cell phones, and PDAs where communication is more power-expensive than computation and contributes significantly to reducing battery life.

Recently, certificateless public key cryptography [1] was studied widely since it could solve both of the certificate management problem in the traditional public key cryptography and the key escrow problem in the ID-based public key cryptography [5]. To satisfy the applications in certificateless environment, certificateless aggregate signature (CLAS) scheme have attracted much attention. Several CLAS schemes [3,4,7,8] have been proposed by different researchers. Recently, Xiong et al. [6] proposed a new certificateless signature (CLS) scheme and used it to construct a simple CLAS scheme. They also demonstrated that both of their schemes are provably secure in the random oracle model under the computational Diffie–Hellman assumption. Unfortunately, we find that a Type II adversary could forge a legal signature of any message against Xiong et al.'s CLS scheme. Meanwhile, their CLAS scheme is also vulnerable to such attack since its signature is a linear combination of those generated by their CLS scheme. To save space, we just demonstrate the attack against their CLS scheme.

2. Preliminaries

2.1. Bilinear pairing

Let G_1 be a cyclic additive group of prime order q, and G_2 be a cyclic multiplicative group of the same order q. We let P be generator of G_1. A bilinear pairing is a map $e{:}G_1 \times G_1 \to G_2$ which satisfies the following properties:

* Corresponding author. Tel.: +86 15307184927.
E-mail address: hedebiao@163.com (D. He).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.09.032

(1) Bilinearity

$$e(aQ, bR) = e(Q, R)^{ab},$$

where $Q, R \in G_1$, $a, b \in Z_q^*$.

(2) Non-degeneracy

$$e(P, P) \neq 1_{G_2}.$$

(3) Computability

There is an efficient algorithm to compute $e(Q,R)$ for all $Q, R \in G_1$.

The Weil and Tate pairings associated with supersingular elliptic curves or abelian varieties can be modified to create such admissible pairings.

2.2. Formal model for CLS scheme

In this subsection, we will review the definition and security notions specified in [6], only with slight notational differences. There are two kinds of adversaries in the CLS scheme and the CLAS scheme, i.e. the Type I adversary $\mathscr{A}\mathbf{1}$ and the Type II adversary $\mathscr{A}\mathbf{2}$. The adversary $\mathscr{A}\mathbf{1}$ is not able to access the master key but he could replace public keys at his will. The adversary $\mathscr{A}\mathbf{2}$ represents a malicious KGC who generates partial private key of users. $\mathscr{A}\mathbf{2}$ could access to the master key of KGC, but he is not able to replace public keys. The following is five oracles which can be accessed by the adversaries.

- *CreateUser*: The simulator generates the user's partial private key, secret key and public key. Then it returns the public key to the adversary.
- *RevealPartialKey*: The simulator returns the user's partial private key to the adversary.
- *RevealSecertKey*: The simulator returns the user's secret key to the adversary.
- *ReplaceKey*: The simulator replaces the user's public key with the one chosen by the adversary.
- *Sign*: The simulator generates a signature of a message and returns to the adversary.

The security for a CLS scheme is defined via the game separately.

Game I: The first game is performed between a challenger $\mathscr{C}$ and an adversary $\mathscr{A} \in \{\mathscr{A}\mathbf{1}, \mathscr{A}\mathbf{2}\}$ for a CLS scheme.

- $\mathscr{C}$ executes *MasterKeyGen* to get master private/public key pair (*mpk*, *msk*).
- $\mathscr{A}$ can adaptively issue the *CreateUser*, *RevealPartialKey*, *RevealSecertKey*, *ReplaceKey* and *Sign* queries to $\mathscr{C}$.
- $\mathscr{A}$ is to output a message m_i^*, and a signature σ_i^* corresponding to a target identity ID_i^* and a public key $upk_{ID_i^*}$.

We say that $\mathscr{A}$ wins **Game I**, if the following three conditions hold.

(1) σ_i^* is a valid signature on messages m_i^* under identities ID_i^* and the corresponding public key $upk_{ID_i^*}$.
(2) If $\mathscr{A}$ is $\mathscr{A}1$, the identity ID_i^* has not submitted to *RevealPartialKey* queries to get the partial private key $psk_{ID_i^*}$. If $\mathscr{A}$ is $\mathscr{A}2, ID_i^*$ has not submitted to *RevealSecertKey* queries or *ReplaceKey* queries to get the secret key $usk_{ID_i^*}$.
(3) The oracle *Sign* has never been queried with (ID_i^*, m_i^*).

Definition 1. A CLS scheme is said to be secure if there is no probabilistic polynomial-time adversary $\mathscr{A} \in \{\mathscr{A}\mathbf{1}, \mathscr{A}2\}$ could win **Game I** with non-negligible advantage.

3. Review of Xiong et al.'s CLS scheme

In this subsection, we will briefly review Xiong et al.'s CLS scheme. Their CLS scheme consists of five algorithms: *MasterKeyGen*, *PartialKeyGen*, *UserKeyGen*, *Sign* and *Verify*. The detail of these algorithms is described as follows.

MasterKeyGen: Given a security parameter k, KGC runs the algorithm to generate the system parameters.

(1) Generate a cyclic additive group G_1 and a cyclic multiplicative group G_2 with prime order q.
(2) Generate two generators P, Q of G_1 and an admissible pairing $e: G_1 \times G_1 \to G_2$.
(3) Generate a random number $s \in Z_q^*$ and compute $P_{pub} = sP$.
(4) Choose cryptographic hash functions $H_1: \{0,1\}^* \to G_1$ and $H_2: \{0,1\}^* \to Z_q^*$.
(5) KGC publishes the system parameters $\{q, G_1, G_2, e, P, Q, P_{pub}, H_1, H_2\}$ and keeps the master key s secretly.

PartialKeyGen: Given a user's identity ID_i, KGC runs the algorithm to generate the user's partial private key.