A note on ‘An efficient certificateless aggregate signature with constant pairing computations’

Debiao He[1,*], Miaomiao Tian[2]

[1] *School of Mathematics and Statistics, Wuhan University, Wuhan, China*

[2] *School of Computer Science and Technology, University of Science and Technology of China, Hefei, China*

*Email: hedebiao@163.com

Abstract*:* Recently, Xiong et al. [H. Xiong, Z. Guan, Z. Chen, F. Li, An Efficient certificateless aggregate signature with constant pairing computations, *Information Science*, doi: 10.1016/j.ins.2012.07.004, 2012] proposed a certificateless signature (CLS) scheme and used it to construct a certificateless aggregate signature (CLAS) scheme with constant pairing computations. They also demonstrated that both of their schemes are provably secure in the random oracle model under the computational Diffie-Hellman assumption. Unfortunately, by giving concrete attacks, we point out that Xiong et al. schemes are not secure in their security model.

***Key words**: Certificateless cryptography; Aggregate signature; Bilinear pairing*

1. Introduction

An aggregate signature scheme is a signature scheme which allows to aggregate n signatures on n distinct messages from n distinct users into a single signature [1]. The validity of an aggregate signature will convince a verifier that the n users did indeed sign the n original messages. Aggregation 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. To satisfy the applications in certificateless environment, certificateless aggregate signature (CLAS) scheme have attracted much attention. Several CLAS schemes [2-5] have proposed by different researchers. However, most of these schemes [2-4] have computational complexity for pairing computations that grows linearly with the number of signers, which deviates from the main goals of aggregate signatures. Besides, both of the schemes [4, 5] of Zhang et al. require certain synchronization, i.e., all signers must share the same synchronized clocks to generate aggregate signature. It is easy to say that it is difficult to achieve synchronization in many communication scenarios. Recently, Xiong et al. [6] proposed an efficient certificateless signature (CLS) scheme and construct a simple CLAS scheme using the CLS scheme. Compared with previous CLAS schemes, Xiong et al.'s scheme is very efficient in computation, and the verification procedure needs only a very small constant number of pairing computations, independent of the number of aggregated signatures. Besides, their scheme does not require a certain synchronization for aggregating randomness. They also demonstrated that both of the two schemes are provably secure in the random oracle model under the computational Diffie-Hellman assumption. Unfortunately, we find that a Type II adversary who knows master key could forge a legal signature of any message against Xiong et al.'s schemes. The analysis shows Xiong et al.'s schemes are not secure for practical applications.

The organization of the paper is sketched as follows. Section 2 gives a brief review of Xiong et al.'s schemes. The security flaws of Xiong et al.'s schemes are shown in Section 3. Finally, we give some conclusions in Section 4.

2. Review of Xiong et al.'s schemes

2.1. 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 as follows.

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 $e: G_1 \times G_1 \rightarrow G_2$.

3) Generate a random number $s \in Z_q^*$ and compute $P_{pub} = sP$.

4) Choose cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \rightarrow Z_q^*$.

5) KGC publishes the system parameters are $\{q, G_1, G_2, e, P, Q, P_{pub}, H_1, H_2\}$ and key the master key s secretly.

PartialKeyGen: Given a user's identity ID_i, KGC computes the user's partial private key $psk_{ID_i} = sQ_{ID_i}$ and transmits it to the user secretly, where $Q_{ID_i} = H_1(ID_i)$.

UserKeyGen: The user with identity ID_i selects a random number $x_{ID_i} \in Z_q^*$ as his secret key usk_{ID_i}, and computes his public key as $upk_{ID_i} = usk_{ID_i} \cdot P$.

Sign: Given a message m_i, the partial private key psk_{ID_i}, the secret key usk_{ID_i}, the user with identity is ID_i and the corresponding public key is upk_{ID_i} performs the following steps to generate a signature.

1) Generate a random number $r_i \in Z_q^*$ and compute $U_i = r_i P$.

2) Compute $h_i = H_2(m_i, ID_i, upk_{ID_i}, U_i)$, $V_i = psk_{ID_i} + h_i \cdot r_i \cdot P_{pub} + h_i \cdot x_{ID_i} \cdot Q$.

3) Output (U_i, V_i) as the signature on m_i.

Verify: Given a signature (U_i, V_i) of message m_i on identity ID_i and corresponding public key upk_{ID_i}:

1) Compute $Q_{ID_i} = H_1(ID_i)$ and $h_i = H_2(m_i, ID_i, upk_{ID_i}, U_i)$.

2) Verify $e(V_i, P) = e(h_i \cdot U_i + Q_{ID_i}, P_{pub}) e(h_i \cdot upk_{ID_i}, Q)$ holds or not. If it holds, accept the signature.

2.2. Xiong et al.'s CLAS scheme

In this subsection, we will briefly review Xiong et al.'s CLAS scheme. Their CLAS scheme consists of six algorithms: *MasterKeyGen*, *PartialKeyGen*, *UserKeyGen*, *Sign*, *Aggregate* and *AggregateVerify*. The first four algorithms are the same as those in their CLS scheme. The detail of other two algorithms is described as follows.

Aggregate: For an aggregating set of n users $\{\mathcal{U}_1, \cdots, \mathcal{U}_n\}$ with identities $\{ID_1, \cdots, ID_n\}$ and the corresponding public keys $\{upk_1, \cdots, upk_n\}$, and message-signature pairs $\{(m_1, \sigma_1 = (U_1, V_1)), \cdots, (m_n, \sigma_n = (U_n, V_n))\}$ from $\{\mathcal{U}_1, \cdots, \mathcal{U}_n\}$

respectively, the aggregate signature generator computes $V=\sum_{i=1}^{n}V_i$ and outputs $\sigma=(U_1,\cdots,U_2,V)$ as an aggregate signature.

AggregateVerify: To verify an aggregate signature $\sigma=(U_1,\cdots,U_2,V)$ signed by n users $\{\mathcal{U}_1,\cdots,\mathcal{U}_n\}$ with identities $\{ID_1,\cdots,ID_n\}$ and the corresponding public keys $\{upk_1,\cdots,upk_n\}$ on messages $\{m_1,\cdots,m_n\}$, the verifier performs the following steps:

1) Compute $Q_{ID_i}=H_1(ID_i)$ and $h_i=H_2(m_i,ID_i,upk_{ID_i},U_i)$ for $i=1,\cdots,n$.

2) Verify $e(V,P)=e(\sum_{i=1}^{n}(h_i\cdot U_i+Q_{ID_i}),P_{pub})e(\sum_{i=1}^{n}h_i\cdot upk_{ID_i},Q)$ holds or not. If it holds, accept the signature.

3. Cryptanalysis of Xiong et al.'s scheme

Xiong et al. [6] claimed that both of their schemes are provably secure against two types of adversary in the random oracle model. However, in this section, we shall disprove their claims by giving two concrete attacks.

3.1. Attack against Xiong et al.'s CLS scheme

Xiong et al. [8] claimed their CLS scheme is semantically secure against Tpye II adversary. Unfortunately, this is not true, since there exists a polynomial time Type II adversary $\mathcal{A}2$ who can always win **Game I** as below:

1) $\mathcal{A}2$ submit a user $\mathcal{U}_i$'s identity ID_i to the *RevealPartialKey* oracle and gets $\mathcal{U}_i$'s partial private key $psk_{ID_i}=sQ_{ID_i}$, where $Q_{ID_i}=H_1(ID_i)$.

2) $\mathcal{A}2$ submit ID_i and a message m_i to the *Sign* oracle and gets a legal signature (U_i,V_i) of message m_i, where $U_i=r_iP$, $h_i=H_2(m_i,ID_i,upk_{ID_i},U_i)$, $V_i=psk_{ID_i}+h_i\cdot r_i\cdot P_{pub}+h_i\cdot x_{ID_i}\cdot Q$ and r_i is a random number generated by *Sign* oracle.

3) $\mathcal{A}2$ computes $T_i=h_i^{-1}(V_i-psk_{ID_i})$, where h_i^{-1} satisfy $h_i^{-1}\cdot h_i\equiv 1\bmod q$.

4) For any other message m_i', $\mathcal{A}2$ computes $U_i'=U_i$, $h_i'=H_2(m_i',ID_i,upk_{ID_i},U_i')$ and $V_i'=psk_{ID_i}+h_i'\cdot T_i$.

5) $\mathscr{A}2$ outputs (U_i', V_i') as the signature on m_i'.

Since $U_i = r_i P$ and $V_i = psk_{ID_i} + h_i \cdot r_i \cdot P_{pub} + h_i \cdot x_{ID_i} \cdot Q$, we could have

$$\begin{aligned} T_i &= h_i^{-1}(V_i - psk_{ID_i}) \\ &= h_i^{-1}(psk_{ID_i} + h_i \cdot r_i \cdot P_{pub} + h_i \cdot x_{ID_i} \cdot Q - psk_{ID_i}) \\ &= r_i \cdot P_{pub} + x_{ID_i} \cdot Q \end{aligned} \qquad (1)$$

$$\begin{aligned} V_i' &= psk_{ID_i} + h_i' \cdot T_i \\ &= psk_{ID_i} + h_i' \cdot (r_i \cdot P_{pub} + x_{ID_i} \cdot Q) \\ &= psk_{ID_i} + h_i' \cdot r_i \cdot P_{pub} + h_i' \cdot x_{ID_i} \cdot Q \end{aligned} \qquad (2)$$

and

$$\begin{aligned} e(V_i', P) &= e(psk_{ID_i} + h_i' \cdot r_i \cdot P_{pub} + h_i' \cdot x_{ID_i} \cdot Q, P) \\ &= e(psk_{ID_i} + h_i' \cdot r_i \cdot P_{pub}, P) e(h_i' \cdot x_{ID_i} \cdot Q, P) \\ &= e(sQ_{ID_i} + h_i' \cdot r_i \cdot sP, P) e(Q, h_i' \cdot x_{ID_i} \cdot P) \\ &= e(h_i' \cdot U_i + Q_{ID_i}, sP) e(h_i' \cdot upk_{ID_i}, Q) \\ &= e(h_i' \cdot U_i + Q_{ID_i}, sP_{pub}) e(h_i' \cdot upk_{ID_i}, Q) \end{aligned} \qquad (3)$$

Then, we know that (U_i', V_i') is a legal signature on m_i'. Besides, ID_i has not been submitted to *RevealSecertKey* queries or *ReplaceKey* queries to get the secret key $usk_{ID_i^*}$ and the oracle *Sign* has never been queried with (ID_i, m_i'). So the Tpye II adversary $\mathscr{A}2$ wins **Game II**.

Therefore, Xiong et al.'s CLS scheme is not secure against attacks of the Type II adversary.

3.2. Attack against Xiong et al.'s CLAS scheme

Xiong et al. [6] claimed their CLAS scheme is semantically secure against Tpye II adversary. Unfortunately, this is not true, since there exists a polynomial time Type II adversary $\mathscr{A}2$ who can always win **Game II** as follows:

Let $\{\mathcal{U}_1, \cdots, \mathcal{U}_n\}$ be an aggregating set of n users with identities $\{ID_1, \cdots, ID_n\}$ and the corresponding public keys $\{upk_1, \cdots, upk_n\}$.

1) For $i = 1, 2, \cdots, n$, $\mathscr{A}2$ does the following five sub-steps to generate a legal signature (U_i', V_i') on a message on m_i'.

- $\mathscr{A}2$ submit a user $\mathcal{U}_i$'s identity ID_i to the *RevealPartialKey* oracle and gets $\mathcal{U}_i$'s partial private key $psk_{ID_i} = sQ_{ID_i}$, where $Q_{ID_i} = H_1(ID_i)$.

- $\mathscr{A}2$ submit ID_i and a message m_i to the $Sign$ oracle and gets a legal signature (U_i , V_i) of message m_i , where $U_i = r_i P$, $h_i = H_2(m_i, ID_i, upk_{ID_i}, U_i)$, $V_i = psk_{ID_i} + h_i \cdot r_i \cdot P_{pub} + h_i \cdot x_{ID_i} \cdot Q$ and r_i is a random number generated by $Sign$ oracle.
- $\mathscr{A}2$ computes $T_i = h_i^{-1}(V_i - psk_{ID_i})$, where h_i^{-1} satisfy $h_i^{-1} \cdot h_i \equiv 1 \bmod q$.
- For message m_i' , $\mathscr{A}2$ computes $U_i' = U_i$, $h_i' = H_2(m_i', ID_i, upk_{ID_i}, U_i')$ and $V_i' = psk_{ID_i} + h_i' \cdot T_i$.
- $\mathscr{A}2$ outputs (U_i', V_i') as the signature on m_i' .

2) $\mathscr{A}2$ computes $V' = \sum_{i=1}^{n} V_i'$

3) $\mathscr{A}2$ outputs $\sigma' = (U_1', \cdots, U_2', V')$ as an aggregate signature.

From the analysis in the above subsection, we know that (U_i', V_i') satisfy the equation $e(V_i', P) = e(h_i' \cdot U_i + Q_{ID_i}, sP_{pub})e(h_i' \cdot upk_{ID_i}, Q)$ and $V_i' = psk_{ID_i} + h_i' \cdot r_i \cdot P_{pub} + h_i' \cdot x_{ID_i} \cdot Q$. Then we could have that

$$\begin{aligned} e(V', P) &= e(\sum_{i=1}^{n} V_i', P) = e(\sum_{i=1}^{n} (psk_{ID_i} + h_i' \cdot r_i \cdot P_{pub} + h_i' \cdot x_{ID_i} \cdot Q), P) \\ &= e(\sum_{i=1}^{n} (psk_{ID_i} + h_i' \cdot r_i \cdot P_{pub}), P) e(\sum_{i=1}^{n} h_i' \cdot x_{ID_i} \cdot Q, P) \\ &= e(\sum_{i=1}^{n} (h_i' \cdot U_i' + Q_{ID_i}), P_{pub}) e(\sum_{i=1}^{n} h_i' \cdot upk_{ID_i}, Q) \end{aligned} \tag{3}$$

Then, we know that $\sigma' = (U_1', \cdots, U_2', V')$ is a legal signature on m_i' . Besides, ID_i has not been submitted to $RevealSecertKey$ queries or $ReplaceKey$ queries to get the secret key $usk_{ID_i^*}$ and the oracle $Sign$ has never been queried with (ID_i, m_i'). So the Tpye II adversary $\mathscr{A}2$ wins **Game IV**.

Therefore, Xiong et al.'s CLAS scheme is not secure against attacks of the Type II adversary.

4. Conclusion

Recently, Xiong et al. proposed a CLS scheme and used it to construct an efficient CLAS scheme. They claimed that both of their schemes are provably secure in the random oracle model. However, after review of their scheme and analysis of its security, we demonstrate that both of the schemes cannot withstand the attack of Type II adversary.

Reference

[1]. D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, *EUROCRYPT'03*, LNCS 3027, pp. 416-432, 2003.

[2]. R. Castro, R. Dahab, Efficient Certificateless Signatures Suitable for Aggregation, Cryptology ePrint Archive, Available online: *http://eprint.iacr.org/2007/454*.

[3]. Z. Gong, Y. Long, X. Hong, K. Chen, Two certificateless aggregate signatures from bilinear maps, in: *IEEE SNPD 2007*, vol. 3, pp. 188-193, 2007.

[4]. L. Zhang, F. Zhang, A new certificateless aggregate signature scheme, *Computer Communications*, 32(6), pp. 1079-1085, 2009.

[5]. L. Zhang, B. Qin, Q. Wu, F. Zhang, Efficient many-to-one authentication with certificateless aggregate signatures, *Computer Networks*, 54(14), pp. 2482-2491, 2010.

[6]. H. Xiong, Z. Guan, Z. Chen, F. Li, An Efficient certificateless aggregate signature with constant pairing computations, *Information Science*, doi: 10.1016/j.ins.2012.07.004, 2012.