On security of a Certificateless Aggregate Signature Scheme

Limin Shen, yinxia sun

Jiangsu Engineering Research Center of Information Security and Privacy Protection Technology
School of Computer Technology, Nanjing Normal University, Nanjing 210097, China
Email: shenlimin@njnu.edu.cn

Abstract. Aggregate signatures are useful in special areas where the signatures on many different messages generated by many different users need to be compressed. Recently, Xiong et al. proposed a certificateless aggregate signature scheme provably secure in the random oracle model under the Computational Diffie-Hellman assumption. Unfortunately, by giving concrete attacks, we indicate that Xiong et al. aggregate signature scheme does not meet the basic requirement of unforgeability.

Keywords: Aggregate signature, Certificateless aggregate signature, Unforgeability, Computational Diffie-Hellman problem

1 Introduction

In traditional public key cryptography, the authenticity of public keys is ensured by certificates signed by a certificate authority(CA). But the issues associated with certificate management are quite complex and costly. In 1984, Shamir[1] first invented a new paradigm called Identity Based public key cryptography (ID-PKC) which simplifies certificate management procedures by deriving public keys for users directly from their identity information, such as e-mail address or telephone number. However, in ID-PKC a trusted third party called Private Key Generator (PKG) must be employed to help a user to generate his private key. The user's private key fully depends on his public known identity and the master secret owned by PKG. So ID-PKC suffers from the key escrow problem.

To fill the gap between traditional cryptography and Identity Based public key cryptography, Al-Riyami and Paterson[2] proposed a new notion called certificateless public key cryptography(CL-PKC)in 2003. In contrast to traditional cryptography, CL-PKC does not require the use of any certificates to ensure the authenticity of public keys. This way, the need of certification can be avoided. Like ID-PKC, CL-PKC also uses a third party called Key Generation Center (KGC) to help a user to generate his private key. Nevertheless, CL-PKC does not suffer from the key escrow property that seems to be inherent in identity-based cryptography, because the KGC does not access to the user's full private key, it only provides the partial private key for the user. The full private key is finally generated by the user who makes use of the secret information chosen by himself and the partial private key obtained from the KGC. The public key of the user is computed from his secret information and the KGC's public parameters, and it is published by the user himself.

One of the most important primitives in public key cryptography is digital signature. Designing a high secure and efficient signature scheme is always desirable since sometimes we have to work in environments with low computability, low-bandwidth communication and low-storage.

An aggregate signature scheme was proposed by Boneh et al.[3] in which multiple signatures can be compressed into one single signature. The validity of an aggregate signature will convince a verifier that the n users did indeed sign the n original messages. CL-PKC achieves escrow free property and does not require any certificate, these advantages may enable the wide use of certificateless signatures(CLS). In fact, there may be some cases of synchronously transmitting and verifying many different signatures signed by many signers. So, it is natural to consider to extend the notion of aggregate signatures to certificateless public key settings to get certificateless aggregate signatures(CLAS), which can aggregate many different certificateless signatures into one single signature, and effectively reduce the message size and verification cost. So, it is interesting to study secure and efficient constructions of aggregate signatures in CL-PKC.

In this paper, we show that the CLAS scheme in [4] is flawed by demonstrating two kinds of attacks against it. In our first attack, we show that a Type I Adversary who replaces one user's public key of an aggregating set $\mathbb{U}$ can forge a valid aggregate signature to a receiver. Similarly, in the second attack, we show that a Type II Adversary who knows master key may impersonate any identity of an aggregating set $\mathbb{U}$ to generate a valid aggregate signature. Therefore, the scheme [4] is subject to the universal forgery of Type I and Type II adversaries. Thus, the original CLAS scheme of Xiong et al. [4] fails to achieve the security goal for an aggregate signature scheme.

The rest of this paper is organized as follows. Section 2 gives some preliminaries. Section 3 introduces the definition and the security notions for certificateless aggregate signature schemes. Section 4 reviews the certificateless aggregate signature (CLAS) scheme of Xiong et al. [4]. And Section 5 presents the attacks on Xiong et al.'s scheme. Finally, Section 6 concludes this paper.

2 Preliminaries

This section revisits some basic concepts and necessary complexity assumptions.

2.1 Bilinear pairing

Let G_1 and G_2 denote two multiplicative cyclic groups of prime order q, and g be a generator of G_1. A map $\hat{e} : G_1 \times G_1 \longrightarrow G_2$ is called a bilinear pairing if it satisfies the following properties:
− Bilinearity: $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab} for\ all\ g, h \in G_1\ and\ a, b \in Z_q^*$.
− Non-degeneracy: $\hat{e}(g, g) \neq I_{G_2}$, where I_{G_2} is the identity element of G_2.
− Computability: There exits an efficient algorithm to compute $\hat{e}(g, h)$ for all $g, h \in G_2$.

2.2 Complexity Assumptions

Discrete Logarithm (DL) Problem: Given a generator g of a cyclic group G with order q, and $h \in G^*$ to find an integer $a \in Z_q^*$ such that $h = g^a$.

The DL assumption means that there is no polynomial time algorithm to solve the DL problem in (g, h, G) with non-negligible advantage.

Computational Diffie-Hellman (CDH) problem: Given a generator g of a cyclic group G with order p, and given (g^a, g^b) for unknown $a, b \in Z_q^*$, to compute g^{ab}.

The CDH assumption means that there is no polynomial time algorithm to solve the CDH problem in (g, g^a, g^b, G) with non-negligible advantage.

3 Certificateless Aggregate Signature(CLAS)

3.1 Formal Definition of Certificateless Aggregate Signature schemes

A generic certificateless aggregate signature scheme is defined by six algorithms[4,6]: MasterKeyGen, Partial-Private-Key-Extract, UserKeyGen, Sign, Aggregate and Aggregate Verify. The description of each algorithm is as follows.

MasterKeyGen: This algorithm is performed by KGC that accepts a security parameter k to generate a master key and a list of system parameters params.

Partial-Private-Key-Extract: This algorithm is performed by KGC that accepts a user's identity ID_i, a parameter list params and a master key to produce the user's partial private key psk_{ID_i}.

UserKeyGen: An algorithm run by a user that takes as input the user's identity ID_i, and selects a random $x_{ID_i} \in Z_q^*$, outputs the user's secret/public key pair (usk_{ID_i}, upk_{ID_i}).

Sign: An algorithm run by each user U_i in an aggregating set $\mathbb{U}$. U_i's inputs are the parameter list params, his identity ID_i, a signing key (psk_{ID_i}, usk_{ID_i}) and a message $m_i \in \{0,1\}^*$. The output of U_i is a signature $\sigma_i \leftarrow Sign(psk_{ID_i}, usk_{ID_i}, m_i)$.

Aggregate: An algorithm run by the aggregate signature generator that takes as inputs an aggregating set $\mathbb{U}$ of n users $\{U_1, \cdots, U_n\}$, the identity ID_i of each user U_i, the corresponding public key upk_{ID_i} of U_i, and a signature σ_i on a message m_i under identity ID_i and public key upk_{ID_i} for each user $U_i \in \mathbb{U}$. The output of this algorithm is an aggregate signature σ on messages $\{m_1, \cdots, m_n\}$.

Aggregate Verify: This algorithm takes as input an aggregating set $\mathbb{U}$ of n users $\{U_1, \cdots, U_n\}$, the identity ID_i and a corresponding public key upk_{ID_i} of each user U_i, an aggregate signature σ on messages $\{m_1, \cdots, m_n\}$. It outputs true if the aggregate signature is valid, or false otherwise.

3.2 Security requirements of Certificateless Aggregate Signature

The basic security requirements for an aggregate signature scheme is unforgeability. Intuitively, we say that an aggregate signature scheme offers unforgeability if nobody can generate a valid aggregate signature on behalf of a generator without the possession of the full private key at least one of the users. Precise definitions of unforgeability is defined using security models. For the detail, please refer to [4,6].

As defined in [5,6,7], there are two types of adversary with different capabilities:

Type I Adversary: This type of adversary $\mathcal{A}_I$ models a malicious adversary which does not have access to the master key, but $\mathcal{A}_I$ has the ability to replace the public key of any entity with a value of his choice, because there is no certificate involved in certificateless aggregate signature schemes.

Type II Adversary: This type of adversary $\mathcal{A}_{II}$ has access to the master key but cannot perform public key replacement.

4 Revisiting the certificateless Aggregate Signature (CLAS) scheme of Xiong et al.

In this section, we will review the certificateless aggregate signature scheme of Xiong et al. [4]. Let G_1 and G_2 denote two multiplicative cyclic groups of prime order p and let $\hat{e} : G_1 \times G_1 \longrightarrow G_2$ be a bilinear pairing.

MasterKeyGen: Given a security parameter $k \in Z$, the algorithm works as follows:

1) Run the parameter generator on input k to generate a prime q, two groups G_1 and G_2 of prime order q, two different generators P and Q in G_1 and an admissible paring $\hat{e} : G_1 \times G_1 \longrightarrow G_2$.

2) Select a master-key $s \in_R Z_q^*$ and set $P_{pub} = sP$.

3) Choose cryptographic hash functions $H_0, H_0' : \{0,1\}^* \to G_1$ and $H_1, H_2, H_2' : \{0,1\}^* \to Z_q^*$. The security analysis will review H_1 and H_2 as random oracles. The system parameters are $\{q, G_1, G_2, \hat{e}, P, Q, P_{pub}, H_0, H_0', H_1, H_2, H_2'\}$. The master key is s.

PartialKeyGen: Given a user's identity $ID_i \in \{0,1\}^*$, KGC first computes $Q_{ID_i} = H_0(ID_i)$ and $Q'_{ID_i} = H_0'(ID_i)$. It then sets this user's partial key $psk_{ID_i} = (sQ_{ID_i}, sQ'_{ID_i})$ and transmits it to user ID_i secretly.

UserKeyGen: The user ID_i selects a secret value $x_{ID_i} \in_R Z_q^*$ as his secret key usk_{ID_i}, and computes his public key as $upk_{ID_i} = x_{ID_i}P$.

Sign: Given its signing key (usk_{ID_i}, psk_{ID_i}), and a message $m_i \in \{0,1\}^*$, the signer, whose identity is ID_i and the corresponding public key is upk_{ID_i} performs the following steps:

1) Computes $h_{i1} = H_1(m_i, ID_i, upk_{ID_i})$, $h_{i2} = H_2(m_i, ID_i, upk_{ID_i})$, $h'_{i2} = H_2'(m_i, ID_i, upk_{ID_i})$.

2) Computes $\sigma_i = h_{i1} \cdot x_{ID_i} \cdot Q + h_{i2} \cdot sQ_{ID_i} + h'_{i2} \cdot sQ'_{ID_i}$.

3) Outputs σ_i as the signature on m_i.

Aggregate: Anyone can act as an aggregate signature generator who can aggregate a collection of individual signatures. For an aggregating set $\mathbb{U}$ of n users $\{U_1, \cdots, U_n\}$ with identities $\{ID_1, \cdots, ID_n\}$ and the corresponding public keys $\{upk_1, \cdots, upk_n\}$, and message-signature pairs $(m_1, \sigma_1), \cdots, (m_n, \sigma_n)$ from $\{U_1, \cdots, U_n\}$ respectively, the aggregate signature generator computes $\sigma = \sum_{i=1}^{n} \sigma_i$ and output σ as an aggregate signature.

Aggregate Verify: To verify an aggregate signature σ signed by n users $\{U_1, \cdots, U_n\}$ 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) For $i = 1, \cdots, n$, compute $Q_{ID_i} = H_0(ID_i)$, $Q'_{ID_i} = H_0'(ID_i)$, $h_{i1} = H_1(m_i, ID_i, upk_{ID_i})$, $h_{i2} = H_2(m_i, ID_i, upk_{ID_i})$ and $h'_{i2} = H_2'(m_i, ID_i, upk_{ID_i})$.

2) Verify the equation

$$\hat{e}(\sigma, P) = \hat{e}(\sum_{i=1}^{n} h_{i1} upk_{ID_i}, Q)\hat{e}(\sum_{i=1}^{n}(h_{i2}Q_{ID_i} + h'_{i2}Q'_{ID_i}), P_{pub})$$

If it holds, accept the signature; else reject it.

5 Security Analysis of the CLAS Scheme by Xiong et al.

In this section, we describe our attacks on Xiong et al.'s scheme [4] to show its security vulnerabilities. Let $\{U_1, \cdots, U_n\}$ be an aggregating set of n users with identities $\{ID_1, \cdots, ID_n\}$ and the corresponding public keys $\{upk_1, \cdots, upk_n\}$ on messages $\{m_1, \cdots, m_n\}$.

5.1 An Attack on Xiong et al.'s scheme Using Type I Adversary

As defined in [6,7], a certificateless aggregate signature scheme is existentially secure iff it resists against type I and type II adversaries. Recall that a type I adversary $\mathcal{A}_I$ does not posses the knowledge of the master key, but the adversary can perform public key replacement, i.e. replacing the public key with his choice. We will show that the scheme in [4] does not resist against a type I adversary since the adversary can successfully forge a valid aggregate signature by replacing one user's public key. The concrete attack is described in four stages.

Stage 1: In this stage, $\mathcal{A}_I$ randomly chooses an identity, without losing generality, $\mathcal{A}_I$ chooses ID_n and picks $x'_{ID_n} \in_R Z_q^*$, computes $upk'_{ID_n} = x'_{ID_n}P$, and replaces the ID_n's public key upk_{ID_n} with upk'_{ID_n}.

Stage 2: $\mathcal{A}_I$ queries the Super-Sign oracle for the signature of m_i for $ID_i (i = 1, \cdots, n-1)$. Let $\{\sigma_1, \cdots, \sigma_{n-1}\}$ be the output of the Super-Sign oracle.

Stage 3: $\mathcal{A}_I$ picks $m, m' \in \{0,1\}^* (m, m' \neq m_i, i = 1, \cdots, n)$, queries the Super-Sign oracle for the signatures of (ID_n, m) and (ID_n, m'). $\{\sigma_n, \sigma'_n\}$ be the output of the Super-Sign oracle. Then $\mathcal{A}_I$ computes

$$g_{n1} = H_1(m, ID_n, upk'_{ID_n}), g_{n2} = H_2(m, ID_n, upk'_{ID_n}), g'_{n2} = H'_2(m, ID_n, upk'_{ID_n})$$

and

$$k_{n1} = H_1(m', ID_n, upk'_{ID_n}), k_{n2} = H_2(m', ID_n, upk'_{ID_n}), k'_{n2} = H'_2(m', ID_n, upk'_{ID_n})$$

So, $\mathcal{A}_I$ has the equations

$$\begin{cases} \sigma_n = g_{n1} \cdot x'_{ID_n} \cdot Q + g_{n2} \cdot sQ_{ID_n} + g'_{n2} \cdot sQ'_{ID_n} \\ \sigma'_n = k_{n1} \cdot x'_{ID_n} \cdot Q + k_{n2} \cdot sQ_{ID_n} + k'_{n2} \cdot sQ'_{ID_n} \end{cases}$$

$\mathcal{A}_I$ can easily obtain the solutions sQ_{ID_n}, sQ'_{ID_n} from the equations. Then $\mathcal{A}_I$ computes

$$h_{n1} = H_1(m_n, ID_n, upk'_{ID_n}), h_{n2} = H_2(m_n, ID_n, upk'_{ID_n}), h'_{n2} = H'_2(m_n, ID_n, upk'_{ID_n})$$

and $\sigma^*_n = h_{n1} \cdot x'_{ID_n} \cdot Q + h_{n2} \cdot sQ_{ID_n} + h'_{n2} \cdot sQ'_{ID_n}$

Stage 4: $\mathcal{A}_I$ generates an aggregate signature $\sigma^* = \sum_{i=1}^{n-1} \sigma_i + \sigma^*_n$. Here, σ^*_n is a forged signature of (ID_n, m_n) and is not the output of the Super-Sign oracle.

Since we have

$\hat{e}(\sigma^*, P)$
$= \hat{e}(\sum_{i=1}^{n-1} \sigma_i + \sigma^*_n, P)$
$= \hat{e}(\sum_{i=1}^{n-1} h_{i1} upk_{ID_i}, Q)\hat{e}(h_{n1} upk'_{ID_n}, Q)\hat{e}(\sum_{i=1}^{n-1}(h_{i2}Q_{ID_i} + h'_{i2}Q'_{ID_i}), P_{pub})\hat{e}((h_{n2}Q_{ID_n} + h'_{n2}Q'_{ID_n}), P_{pub})$
$= \hat{e}(\sum_{i=1}^{n} h_{i1} upk_{ID_i}, Q)\hat{e}(\sum_{i=1}^{n}(h_{i2}Q_{ID_i} + h'_{i2}Q'_{ID_i}), P_{pub})$

the verification equation always holds. This declares that the forged aggregate signature σ^* is valid. Therefore, the scheme is subject to universal forgery with respect to a Type I Adversary $\mathcal{A}_I$ who replaces one of the identities's public key.

5.2 An Attack on Xiong et al.'s scheme Using Type II Adversary

Recall that a type II adversary $\mathcal{A}_{II}$ is given the master key, but can not replace any public keys. In most security models, $\mathcal{A}_{II}$ is just like a "$malicious-but-passive$" KGC. We will show that the scheme in [4] does not resist against a type II adversary since $\mathcal{A}_{II}$ can successfully forge a valid aggregate signature. The concrete attack is described in four stages.

Stage 1: In this stage, $\mathcal{A}_{II}$ randomly chooses an identity, without losing generality, $\mathcal{A}_{II}$ chooses ID_n and picks $m'_n \in \{0,1\}^*(m'_n \neq m_i, i = 1, \cdots, n)$. $\mathcal{A}_{II}$ queries the Super-Sign oracle for the signature of m'_n for ID_n. Let σ'_n be the output of the Super-Sign oracle.

Stage 2: $\mathcal{A}_{II}$ computes

$$k_{n1} = H_1(m'_n, ID_n, upk_{ID_n}), k_{n2} = H_2(m'_n, ID_n, upk_{ID_n}), k'_{n2} = H'_2(m'_n, ID_n, upk_{ID_n})$$

So, $\mathcal{A}_I$ has the equation

$$\sigma'_n = k_{n1} \cdot x_{ID_n} \cdot Q + k_{n2} \cdot sQ_{ID_n} + k'_{n2} \cdot sQ'_{ID_n}$$

Since $\mathcal{A}_{II}$ kmows the master key, s, it can easily obtain the solution $x_{ID_n} \cdot Q$ from the equation. Then $\mathcal{A}_{II}$ computes

$$h_{n1} = H_1(m_n, ID_n, upk_{ID_n}), h_{n2} = H_2(m_n, ID_n, upk_{ID_n}), h'_{n2} = H'_2(m_n, ID_n, upk_{ID_n})$$

and $\sigma_n = h_{n1} \cdot x_{ID_n} \cdot Q + h_{n2} \cdot sQ_{ID_n} + h'_{n2} \cdot sQ'_{ID_n}$. Here, σ_n is a forged signature of (ID_n, m_n) and is not the output of the Super-Sign oracle.

Stage 3: $\mathcal{A}_{II}$ queries the Super-Sign oracle for the signature of m_i for $ID_i (i = 1, \cdots, n-1)$. Let $\{\sigma_1, \cdots, \sigma_{n-1}\}$ be the output of the Super-Sign oracle.

Stage 4: $\mathcal{A}_{II}$ generates an aggregate signature $\sigma^* = \Sigma_{i=1}^{n} \sigma_i$.

It is clear that σ^* is a valid aggregate signature since the verification equation always holds:

$$\hat{e}(\sigma^*, P) = \hat{e}(\sum_{i=1}^{n} h_{i1} upk_{ID_i}, Q)\hat{e}(\sum_{i=1}^{n}(h_{i2}Q_{ID_i} + h'_{i2}Q'_{ID_i}), P_{pub})$$

6 Conclusion

In this paper, we demonstrated two kinds of concrete attacks against the recently proposed CLAS scheme by Xiong et al. according to their security model. In our attacks, a Type I adversary can forge a valid aggregate signature by replacing the public key of one of the identities, and a Type II Adversary can forge a valid aggregate signature by using the master key. Thus, the CLAS scheme by Xiong et al. fails to meet the requirement of unforgeability for a secure aggregate signature scheme.

References

1. Adi Shamir. Identity-based cryptosystems and signature schemes. In CRYPTO, pp47-53, 1984.
2. S. S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. Advances in Cryptography - Asiacrypt 2003, LNCS 2894, pp452-473, Springer-Verlag, Berlin, 2003.

3. D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, in: E. Biham (Ed.), EUROCRYPT 2003, LNCS 2656, Springer-Verlag, Warsaw, Poland, 2003, pp. 416432.
4. Hu Xiong, Qianhong Wu, Zhong Chen. Strong Security Enabled Certificateless Aggregate Signatures Applicable to Mobile Computation. 2011 Third International Conference on Intelligent Networking and Collaborative Systems, 978-0-7695-4579-0/11 $26.00©2011 IEEE DOI 10.1109/INCoS.2011.151.
5. K. Y. Choi, J. H. Park, D. H. Lee. A New Provably Secure Certificateless Short Signature Scheme. Computers and Mathematics with Applications, vol. 61, no. 7, pp. 1760-1768, 2011.
6. L. Zhang, F. Zhang. A New Certificateless Aggregate Signature Scheme. Computer Communications, vol. 32, no. 6, pp. 1079-1085, 2009.
7. L. Zhang, B. Qin, Q. Wu, F. Zhang. Efficient Many-to-One Authentication with Certificateless Aggregate Signatures. Computer Networks, vol. 54, no. 14, pp. 2482-2491, 2010.