Cryptanalysis of a Compact Certificateless Aggregate Signature Scheme

Chih-Cheng Chen[1], Hanmin Chien[2], and Gwoboa Horng[1]
(Corresponding author: Gwoboa Horng)
Department of Computer Science and Engineering, National Chung Hsing University[1]
250 Kuo Kuang Road, Taichung, Taiwan 402 (R.O.C.)
Department of Digital Multimedia Design, China University of Technology[2]
No. 56, Sec. 3, Xinglong Rd., Wunshan District, Taipei City 116, Taiwan (R.O.C.)
(Email: gbhorng@cs.nchu.edu.tw)
(Received July 31, 2014; revised and accepted Sept. 24 & Oct. 6, 2014)

Abstract

In this paper, we cryptanalyze a recently proposed compact certificateless aggregate signature scheme (CCLAS) and show that it is in fact insecure against a Type-I attack. We also point out that the success of the attack is due to the inappropriate security model used to prove that CCLAS is secure.

Keywords: Aggregate signature, certificateless cryptography, cryptanalysis

1 Introduction

The most important contribution of modern cryptography is the invention of a way to create digital signatures. A digital signature is an electronic analogue of a written signature to be used by the recipient or a third party to identity of the signatory or to verify the integrity of the data. To deal with specific application scenarios, digital signature schemes have evolved into many different variants. Among them, aggregate signature schemes, which allow a collection of individual signatures to be compressed into a single short signature, are most useful for reducing the size of certificate verification chains and for reducing message size in secure routing protocols [2].

Certificateless public key cryptography (CL-PKC) [1] was proposed in 2003. Since then many cryptographic schemes have been proposed based on CL-PKC. CL-PKC solves the key escrow problem of the identity-based cryptography in a way that the full private key of a user is divided into two parts. The first part, called partial private key, is controlled by a key generator center (KGC). The second part is chosen by the user himself and remains secret to the KGC. Therefore, to discuss the security issues of CL-PKC, there are two types of attacks, depending on which part of the private key is compromised.

In 2014, Zhou et al. proposed a compact certificateless aggregate signature scheme (CCLAS) [15]. They also defined security models and showed that CCLAS is existentially unforgeable under adaptive chosen-message attacks and chosen-identity attacks. In this paper, we cryptanalyze CCLAS and show that it is in fact insecure against a Type-I attack.

The organization of this paper is as follows. Section 2 consists of some preliminaries, including a generic construction of a certificateless aggregate signature scheme and security models. Review of CCLAS is given in Section 3. The cryptanalysis of CCLAS is presented in Section 4. Finally, we give conclusions in Section 5.

2 Preliminaries

2.1 Generic Construction of a Certificateless Aggregate Signature Scheme

A certificateless aggregate signature (CLAS) scheme consists of three parts, initial setup **InitSetup**, signature generation and aggregation **CL-Sign**, and signature verification **CL-Verify**:

InitSetup. This part consists of the following algorithms:

- **Setup:** This algorithm, run by the KGC, takes a security parameter as input, then outputs master-key and system parameter params.
- **Partial-Private-Key-Extract:** This algorithm, run by the KGC, takes params, master-key and a user's identity ID as inputs, then outputs a partial-private-key D_{ID} to that user.
- **Set-Secret-Value:** This algorithm, run by a user, returns a secret value x.
- **Set-Private-Key:** This algorithm, run by a user, takes the user's partial-private-key D_{ID} and his

secret value as inputs, and outputs the full private key.

Set-Public-Key: This algorithm, run by a user, takes params and the user's full private key as inputs, and outputs a public key pk_{ID} for that user.

CL-Sign. This part consists of an individual signature generation algorithm and a signature aggregation algorithm.

IndiSign: The individual signature generation algorithm, run by a signer, takes params, a message m, and the user's full private key as inputs, and outputs σ as the signature for the message m.

SignAggr: The signature aggregation algorithm, run by any user or a third party, takes n individual signatures σ_i on messages m_i generated by users of identities ID_i where $i = 1, \cdots, n$, as input and returns an aggregate signature σ.

CL-Verify. This part consists of an individual signature verification algorithm and an aggregate signature verification algorithm.

IndiVeri: The individual signature verification algorithm, run by a verifier, takes params, a public key pk_{ID}, a message m, a user's identity ID, and a signature S as inputs. The verifier accepts signature S if and only if S is the signature of the message m for the public key pk_{ID} of the user with identity ID.

SignVeri: The aggregate signature verification algorithm, run by a verifier, takes an aggregate signatures σ_i on messages m_i generated by users of identities ID_i and public key pk_{ID_i} where $i = 1, \cdots, n$, as input and accepts the aggregate signature σ if it is valid.

2.2 Security Models

Traditionally, a digital signature scheme is secure if it is existentially unforgeable against adaptive chosen message attacks. The attack methods are centered on querying signatures for adaptive chosen messages. For a CLS scheme, the situation is more complicated since the attackers can do a lot more than merely querying signatures. For example, they can query for the partial private key of any user.

Therefore, when discussing the security issues of a certificateless signature scheme, there are two types of adversaries, A_I and A_{II} corresponding to two types of attack models Type-I and Type-II respectively. A Type-I attack model is used to model the case when an adversary A_I has compromised the user secret value or replace the user public key. However, he cannot compromise the master-key nor access the user partial key. Whereas a Type-II attack model is used to model the case when an adversary A_{II} (the malicious-but-passive KGC) has gained access to the master key but cannot perform public key replacement of the user being attacked. Since our attack is of Type-I, we describe the attack model in more detail. We refer the readers to [15] for the Type-II attack model.

The type-I attack model is defined in terms of a game played between a challenger C and the Type-I adversary A_I as follows.

Initialization. C runs Setup algorithm to generate the master key and public parameters to A_I.

Queries. A_I can adaptively perform the following polynomially bounded queries.

Partial-Private-Key query: $\mathcal{A}_1$ can query for the partial private key of any user with identity ID. $\mathcal{C}$ will return the partial private key D_{ID} to $\mathcal{A}_1$.

Public-Key query: $\mathcal{A}_1$ can query for the public key of any user with identity ID. $\mathcal{C}$ will return the public key pk_{ID} of that user.

Secret-Value query: $\mathcal{A}_1$ can query for the secret value of any user with identity ID. $\mathcal{C}$ will return the secret value x_{ID} of that user to $\mathcal{A}_1$.

Public-Key-Replacement: For any user with identity ID and public key pk, $\mathcal{A}$ can set a new public key pk', and then $\mathcal{C}$ replaces pk with pk'.

IndiSign query: $\mathcal{A}_1$ can query for the signature σ_i corresponding to a message m_i, a user with identity ID_i and public key pk_i. $\mathcal{C}$ will generate σ_i, and return it to $\mathcal{A}_1$.

SignAggr query: $\mathcal{A}_1$ can query aggregate signature for multiple signatures, $\mathcal{C}$ will return an aggregate signature σ by the **SignAggr** algorithm and return it to $\mathcal{A}_1$.

Forgery. $\mathcal{A}_1$ outputs an aggregate signature $\sigma^* = (R^*, S^*)$ of n individual signatures σ_i on messages m_i generated by users of identities ID_i^* where $i = 1, \cdots, n$. $\mathcal{A}_1$ wins the game if and only if the following conditions hold.

1) The forged aggregate signature σ^* is valid.
2) For each i, $1 \leq i \leq n$, at least one of the secret value or the partial private key of ID_i^* has not been queried.
3) σ^* has never been queried by the *IndiSign* and *SignAggr* oracles.

3 CCLAS

Most certificateless signature schemes are based on bilinear pairing [10, 11, 12, 13]. A bilinear map is a mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, where $\mathbb{G}_1$ is an additive cyclic group of prime order q, and $\mathbb{G}_2$ is a multiplicative cyclic group of the same order q. We are interested in bilinear maps with the following properties:

1) Computable: given $P, Q \in \mathbb{G}_1$, there exists a polynomial time algorithm to compute $\hat{e}(P, Q) \in \mathbb{G}_2$.

2) Bilinear: for any $x, y \in \mathbb{Z}_q^*$, we have $\hat{e}(xP, yP) = \hat{e}(P, P)^{xy}$ for any $P \in \mathbb{G}_1$.

3) Non-degenerate: if P is a generator of $\mathbb{G}_1$, then $\hat{e}(P, P)$ is a generator of $\mathbb{G}_2$.

The CCLAS scheme consists of eight probabilistic-polynomial time algorithms, namely Setup, PartialKeyGen, UserKeyGen, IndiSign, IndiVeri, SignAggr, SignVeri and ExtAggr.

Setup: The KGC determines a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ where $\mathbb{G}_1$ is a cyclic additive group of prime order q with a generator P, $\mathbb{G}_2$ is a cyclic multiplicative group of the same order, and three hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*, H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$, and $H_3 : \{0,1\}^* \to \mathbb{G}_1$. Then it randomly chooses $s \in \mathbb{Z}_q^*$ as master-key, and then sets P_{pub} as the master-public-key where $P_{pub} = sP$. Finally, it publishes the system parameter params $= \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, P_{pub}, H_1, H_2, H_3 \rangle$.

Partial-Private-Key-Extract: The KGC, based on params, master-key s and user's identity ID_i, computes and returns a partial-private-key $D_i = sQ_i$ to the user with identity ID_i where $Q_i = H_1(ID_i)$.

UserKeyGen: A user with identity ID_i, sets a random value $x_i \in \mathbb{Z}_q^*$ as his secret value and public key $P_i = x_i P$. The pair (D_i, x_i) is the user's full secret key SK_i.

IndiSign: To facilitate the aggregation of individual signatures, a random string ω, called state string, is chosen by the first signer. Each subsequent signer checks that it has not used the string ω before. To sign a message m_i using the full secret key (x_i, D_i), the signer with identity ID_i should perform the following steps:

1) Compute $P_\omega = H_2(\omega)$;
2) Pick a random number from r_i $\mathbb{Z}_q^*$ and compute $R_i = r_i P$;
3) Compute $h_i = H_3(\omega)$;
4) Compute $P_\omega = H_2(m_i, ID_i, \omega)$;
5) Compute $S_i = r_i P_\omega + D_i + x_i h_i$;
6) Output $\sigma_i =< R_i, S_i >$.

IndiVeri: To verify a signature $\sigma_i =< R_i, S_i >$ on the state string ω and the message m_i, the verifier should perform the following steps:

1) Compute $P_\omega = H_2(\omega)$;
2) Compute $h_i = H_3(m_i, ID_i, \omega)$;
3) Accept the signature if and only if $\hat{e}(P, S_i) = \hat{e}(R_i, P_\omega)\hat{e}(P_{pub}, Q_i)\hat{e}(P_i, h_i)$.

SignAggr: For $i = 1, \cdots, n$, to aggregate signatures $\sigma_i =< R_i, S_i >$ on state string ω and messages m_i signed by users with identities ID_i, one should perform the following:

1) Compute $S = \Sigma_{i=1}^n S_i$ and $R = \Sigma_{i=1}^n R_i$;
2) Output the aggregate signature $\sigma =< R, S >$.

SignVeri: To verify a signature $\sigma_i =< R_i, S_i >$ on the state string ω and the message m_i, the verifier should perform the following steps:

1) Compute $P_\omega = H_2(\omega)$;
2) Compute $Q_i = H_1(ID_i)$ and $h_i = H_3(m_i, ID_i, \omega)$ for $i = 1, \cdots, n$;
3) Accept the aggregate signature if and only if

$$\hat{e}(P, S) = \hat{e}(R, P_\omega)\hat{e}(P_{pub}, \Sigma_{i=1}^n Q_i)\hat{e}(\Pi_{i=1}^n P_i, h_i).$$

The aggregate signature is compact in a sense that its length is the same as that of an individual signatures. Furthermore, CCLAS scheme introduces another algorithm called ExtAggr which can be used to extract a valid individual signature. When an individual signature is be extracted from the aggregate signature the remaining part is also a valid aggregate signature.

4 Cryptanalysis of CCLAS

4.1 A Type I Attack

In this section we will show that is in fact forgeable under Type I attack. The attack goes as follows.

Suppose an adversary, say Alice, knows the secret value x_i of a user with identity ID_i through the *Public-Key-Replacement* query or the *Secret-Value query* query.

Then Alice can issue an IndiSign query to obtain a signature σ_i on a message m_i and a state string ω such that $\sigma_i = (R_i, S_i)$ where $R_i = r_i P$, $S_i = r_i P_\omega + D_i + x_i h_i$, and $h_i = H_3(m_i, ID_i, \omega)$. Note that Alice cannot compute the partial private key D_i directly. However, from σ_i, Alice can compute $T = r_i P_\omega + D_i = S_i - x_i h_i$ since x_i is known.

Now it is very simple for Alice to forge a signature $\sigma' = (R', S')$ for any message m' under the same state string ω. She only needs to set $R' = R$ and $S' = T + x_i h'$ where $h' = H_3(m', ID_i, \omega)$. Since $\hat{e}(P, S') = \hat{e}(P, T + x_i h') = \hat{e}(P, r_i P_\omega + D_i + x_i h') = \hat{e}(R', P_\omega)\hat{e}(P_{pub}, Q_i)\hat{e}(P_i, h')$, $\sigma' = (R', S')$ is indeed a valid signature for message m'.

Hence, given an aggregate signature σ which includes σ_i, Alice can use ExtAggr algorithm to extract σ_i from σ followed by adding σ' to it to obtain a forged aggregate signature σ^*.

4.2 Discussion

The linear equation used to construct the second part of a signature in CCLAS is similar to that of the CLS short signature scheme proposed in [5] and attacked by

Shim in [7]. Therefore, the same attack can also be used to attack CCLAS. In [6], three kinds of adversaries are introduced, namely normal, strong, and super. They are distinguished by their attack power. A strong Type I adversary can make a strong-sign query which takes as input (ID, m, sv), where ID denotes the identity that has been created, m denotes the message to be signed and sv is the secret value. In the above attack, Alice is a strong Type-I adversary. Therefore, CCLAS is insecure against strong Type-I attacks.

Over the years, many provably secure certificateless signature schemes have been proposed under certain security models. However, they are shown to be insecure [3, 4, 8, 9, 14]. Therefore, the security models for certificateless signature schemes are quite subtle. Based on the security models of CCLAS, to existentially forge a signature is equivalent to derive the partial private key of a user. However, as mention in the attack, our attack cannot derive the partial key but instead forge a signature based on an existent state string. Therefore, the security model used to prove that CCLAS is secure is inappropriate.

5 Conclusions

The integration of certificateless public key cryptography and aggregate signature has many potential applications. However, for a certificateless aggregate signature scheme to be used in application environments, we must make sure that it is secure against attacks. Therefore cryptanalysis plays a vital role for a cryptographic protocol to be successfully applied in the real world. In this paper, we have analyzed CCLAS scheme and showed that it is not secure against strong Type-I attacks.

Acknowledgement

This work was partially supported by the National Science Council of the Republic of China under contract No. NSC102-2221-E-005-051.

References

[1] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," *Proceedings of Advances in Cryptography (ASIACRYPT'03)*, LNCS 2894, pp. 452–473, Springer-Verlag, 2003.

[2] D. Boneh, D. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," *Proceedings of Advances in Cryptography (EUROCRYPT'03)*, LNCS 2656, pp. 416–432, Springer-Verlag, 2003.

[3] D. He, M. Khan, and S. Wu, "On the Security of a RSA-based Certificateless Signature Scheme," *International Journal of Network Security*, vol. 16, pp. 78–80, 2014.

[4] D. He, M. Tian, and J. Chen, "Insecurity of an efficient certificateless aggregate signature with constant pairing computations." *Information Sciences*, vol. 268, pp. 458–462, 2014.

[5] X. Huang, Yi Mu, W. Susilo, D.S. Wong, and W. Wu, "Certificateless signature revisited," in *Information Security and Privacy*, LNCS 4586, pp. 308–322, Springer-Verlag, 2007.

[6] X. Huang, Y. Mu, W. Susilo, D. Wong, and W. Wu, "Certificateless signatures: new schemes and security models," *The Computer Journal*, vol. 55, pp. 457–474, 2012.

[7] K. Shim, "Breaking the short certificateless signature scheme," *Information Sciences*, vol. 179, pp. 303–306, 2009.

[8] K. Shim, "On the security of a certificateless aggregate signature scheme," *Communications Letters*, vol. 15, pp. 1136–1138, 2011.

[9] H. Tu, D. He, and B. Huang, "Reattack of a certificateless aggregate signature scheme with constant pairing computations," *The Scientific World Journal*, Article ID 343715, 10 pages, 2014.

[10] C. Wang, D. Long, and Y. Tang, "An Efficient Certificateless Signature from Pairings," *International Journal of Network Security*, vol. 8, pp. 96–100, 2009.

[11] H. Xiong, Z. Guan, Z. Chen, F. Li, "An efficient certificateless aggregate signature with constant pairing computations," *Information Science*, vol. 219, pp. 225–235, 2013.

[12] H. Xiong, Z. Qin, and F. Li, "A certificateless proxy ring signature scheme with provable security," *International Journal of Network Security*, vol. 12, pp. 92–106, 2011.

[13] L. Zhang and F. Zhang, "A new certificateless aggregate signature scheme," *Computer Communications*, vol. 32, pp. 1079–1085, 2009.

[14] M. Zhang, J. Yao, C. Wang, and T. Takagi, "Public key replacement and universal forgery of a SCLS scheme," *International Journal of Network Security*, vol. 15, pp. 133–138, 2013.

[15] M. Zhou, M. Zhang, C. Wang, and B. Yang, "CCLAS: A practical and compact certificateless aggregate signature with share extraction," *International Journal of Network Security*, vol. 16, pp. 174–181, 2014.

Chih-Cheng Chen received the M.S. degree in graduate school of computer science and information technology from National Taichung Institute of Technology, Taiwan. He is currently pursuing the Ph.D. degree in computer science and engineering from National Chung Hsing University. His research interests include data hiding, secret sharing, watermarking and image processing.

Han-min Chien was born in Taipei, Taiwan, in 1971. He received the Ph.D. degree in electronics & electrical engineering from the Queens University of Belfast, N. Ireland, U.K., in 2005. In 2006, he joined the Department of Computer Science & Information Engineering,

China University of Technology, Taiwan, as an Assistant Professor, and transferred to the Department of Digital Multimedia Design in 2011. His current research interests include image processing, virtual reality, medical instruments, and Ergonomics. He was the recipient of Young Investigator Applicants Encouragement Award from the commit of 6th Asian-Pacific Conference on Medical and Biological Engineering, Japan, 2005, the SPUR-VEC Research Funding of the Queens University of Belfast & Parliament of Northern Ireland, from 2001 to 2005.

Gwoboa Horng received the B.S. degree in Electrical Engineering from National Taiwan University in 1981 and the M.S. and Ph.D. degrees from University of Southern California in 1987 and 1992, respectively, all in Computer Science. Since 1992, he has been on the faculty of the Department of Computer Science and Engineering at National Chung-Hsing University, Taichung, Taiwan, R.O.C. His current research interests include artificial intelligence, cryptography, and information security.