Research Article

On the Security of a Certificateless Proxy Signature Scheme with Message Recovery

Wenbo Shi,¹ Debiao He,² and Peng Gong³

¹ Department of Electronic Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

² School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China

³ National Key Laboratory of Mechatronic Engineering and Control, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Wenbo Shi; swb319@hotmail.com

Received 19 January 2013; Accepted 5 April 2013

Academic Editor: Wanquan Liu

Copyright © 2013 Wenbo Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A proxy signature scheme allows a proxy signer to sign messages on behalf of an original signer within a given context. It has lots of practical applications in distributed systems, grid computing, mobile agent applications, distributed shared object systems, global distribution networks, and mobile communications. Recently, Padhye et al. proposed a certificateless proxy signature scheme with message recovery and claimed the scheme is secure against both of the two types of adversaries. However, in this paper, we will show that Padhye et al.'s scheme is not secure against the Type I adversary. The analysis shows their scheme is not secure for practical applications.

1. Introduction

The proxy signature scheme is an important cryptographic mechanism, which was introduced first by Mambo et al. [1] in 1996. In the scheme, the original signer could delegate his signing capability to the proxy signer. After that, the proxy signer could sign a message on behalf of the original signer. The proxy signature has been widely used in distributed shared object systems, grid computing, mobile agent environment and global distribution networks, where delegation of rights is quite common [2, 3].

Recently, certificateless public key cryptography was studied widely since it could solve the certificate management problem in the traditional public key cryptography and the problem in the identity-based public key cryptography. Many certificateless key agreement schemes [4–6] and certificateless signature schemes [7–9] have been proposed for different applications. To satisfy the applications in the certificateless environment, many certificateless proxy signature (CLPS) schemes [10–17] have been proposed. In 2005, Li et al. [10] proposed the first CLPS scheme. Later, Yap et al. [11] and Lu

et al. [12] found that Li et al.'s scheme is not secure at all. Lu et al. [12] also proposed an improved CLPS scheme. In 2009, Chen et al. [13] proposed the first security model for the CLPS scheme. They also proposed a new CLPS scheme and demonstrated it was provably secure in the security model. To improve performance, several other CLPS schemes [14–16] with provably security were also proposed. All the above CLPS schemes are based on bilinear pairings. The performance of these schemes [10–16] is not satisfactory since the bilinear pairing operation is very complicated. To avoid bilinear pairing operation, Padhye and Tiwari [17] proposed a certificateless proxy signature scheme with message recovery. They also proved their scheme is secure against chosen message and identity attacks in the random oracle model. In this letter, we will show and discuss the security of Padhye et al.'s scheme and show it is not secure against the Type I adversary.

The rest of the paper is organized as follows. Section 2 gives a review of Padhye et al.'s scheme. Section 3 discusses the security problem in Padhye et al.'s scheme. Finally, we conclude the paper in Section 4.

2. Review of Padhye et al.'s Scheme

In this section, we will review Padhye et al.'s scheme. For convenience, some notations used in the paper are described in the Abbreviations section.

Padhye et al.'s CLPS scheme is composed of ten algorithms, which are *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key*, *DelGen*, *DelVerif*, *PKGen*, *PSign*, and *PSVerif*. The details of these algorithms are described as follows.

Setup. Taking a security parameter k as inputs, the KGC runs this algorithm to generate the system parameters.

- (1) KGC chooses a k -bit prime p , generates an elliptic curve E over finite field F_p , generates a group G of elliptic curve points on E with prime order n , and determines a generator P of G .
- (2) KGC chooses the master key $mk = x \in Z_n^*$ and computes the master public key $P_{pub} = x \cdot P$.
- (3) KGC chooses four cryptographic secure hash functions $H_i : \{0, 1\}^* \rightarrow Z_n^*$, where $i = 1, 2, 3, 4$.
- (4) KGC publishes $params = \{F_p, E, G, P, P_{pub}, H_1, H_2, H_3, H_4\}$ as system parameters and secretly keeps the master key x .

Partial-Private-Key-Extract. Taking a user's identity ID_U , system parameters $params$, and the master key x as inputs, KGC runs the algorithm to generate the user's partial private key.

- (1) KGC generates a random number $k_U \in Z_n^*$ and computes $K_U = k_U \cdot P$ and $h_U = H_1(ID_U, K_U)$.
- (2) KGC computes $d_U = (k_U + h_U s) \bmod n$ and sends (d_U, K_U) to the user through a secure channel.

Set-Secret-Value. Taking system parameters $params$ as inputs, the user U runs the algorithm to generate the secure value.

- (1) U generates a random number $x_U \in Z_n^*$ and computes $P_U = x_U \cdot P$.
- (2) U sets x_U as the secret value.

Set-Private-Key. Taking the secret value x_U and the partial private key d_U as inputs, the user sets $sk_U = (x_U, d_U)$ as his private key.

Set-Public-Key. Taking P_U and K_U as inputs, the user sets $pk_U = (P_U, K_U)$ as his public key.

DelGen. Taking system parameters $params$, the original signer OS's private key $sk_{OS} = (x_{OS}, d_{OS})$, the proxy signer PS's public key $pk_{OS} = (P_{OS}, K_{OS})$, and a warrant message m_ω as inputs, the original signer OS runs this algorithm to generate a delegation on the warrant message m_ω .

- (1) OS generates a random number $r_{OS} \in Z_n^*$ and computes $R_{OS} = r_{OS} \cdot P$.
- (2) OS computes $\sigma_{OS} = ((x_{OS} + d_{OS})e_{OS} + r_{OS}) \bmod n$ and sends the delegation $(m_\omega, R_{OS}, \sigma_{OS})$ to the proxy signer PS, where $e_{OS} = H_2(m_\omega, ID_{OS}, K_{OS}, P_{OS}, R_{OS})$.

DelVerif. Take the delegation $(m_\omega, R_{OS}, \sigma_{OS})$, system parameters $params$, and OS's public key $pk_{OS} = (P_{OS}, K_{OS})$ as inputs; PS runs the algorithm to verify the validity of the delegation.

- (1) PS computes $e_{OS} = H_2(m_\omega, ID_{OS}, K_{OS}, P_{OS}, R_{OS})$ and $h_{OS} = H_1(ID_{OS}, K_{OS})$.
- (2) PS checks whether the equation $\sigma_{OS} \cdot P = e_{OS}(P_{OS} + K_{OS} + h_{OS}P_{pub}) + R_{OS}$ holds. If it holds, PS accepts the delegation; otherwise, PS rejects the delegation.

PKGen. Taking system parameters $params$, the delegation $(m_\omega, R_{OS}, \sigma_{OS})$, and PS's private key $sk_{PS} = (x_{PS}, d_{PS})$ as inputs, PS runs the algorithm to generate his proxy private key.

- (1) PS computes $e_{PS} = H_3(m_\omega, ID_{PS}, K_{PS}, P_{PS})$.
- (2) PS computes $D_{PS} = (\sigma_{OS} + e_{PS}(x_{PS} + d_{PS})) \bmod n$ and sets D_{PS} as the proxy key.

PSign. Taking a message $m \in \{0, 1\}^*$, system parameters $params$, and the proxy private key D_{PS} as inputs, PS runs this algorithm to generate a proxy signature.

- (1) PS generates a random number $r_{PS} \in Z_n^*$ and computes $R_{PS} = r_{PS} \cdot P$.
- (2) PS computes $t_{PS} = (m \parallel H_4(m) + (R_{PS})_x) \bmod n$, where $(R_{PS})_x$ denotes the x -coordinates of the elliptic curve group point R_{PS} .
- (3) PS computes $e = H_5(t_{PS}, P_{OS}, K_{PS}, P_{PS})$ and $\sigma_{PS} = (r_{PS} - eD_{PS}) \bmod n$.
- (4) PS outputs $(m_\omega, \sigma_{OS}, R_{OS}, t_{PS}, \sigma_{PS})$ as the proxy signature.

PSVerif. Taking the proxy signature $(m_\omega, \sigma_{OS}, R_{OS}, t_{PS}, \sigma_{PS})$, the message m , OS's public key $pk_{OS} = (P_{OS}, K_{OS})$, PS's public key $pk_{PS} = (P_{PS}, K_{PS})$, and system parameters $params$ as inputs, the verifier V runs this algorithm to verify the validity of the proxy signature.

- (1) V computes $e_{OS} = H_2(m_\omega, ID_{OS}, K_{OS}, P_{OS}, R_{OS})$, $h_{OS} = H_1(ID_{OS}, K_{OS})$, $e_{PS} = H_3(m_\omega, ID_{PS}, K_{PS}, P_{PS})$, $h_{PS} = H_1(ID_{PS}, K_{PS})$, and $e = H_5(t_{PS}, P_{OS}, K_{PS}, P_{PS})$.
- (2) V computes $H_4(m) = t_{PS} - (\sigma_{PS} \cdot P + e(e_{OS}(P_{OS} + K_{OS} + h_{OS}P_{pub}) + R_{OS} + e_{PS}(P_{PS} + K_{PS} + h_{PS}P_{pub})))_x$.
- (3) V checks whether the hash result of the recovered m is equal to $H_4(m)$. If they are equal, V accepts the signature; otherwise, V rejects the signature.

3. Security Analysis of Padhye et al.'s Scheme

There are two types of adversaries with different capabilities in CLPS schemes. They are known as Type I adversaries and Type II adversaries. The Type I adversary A_I models an outsider adversary, who could replace the public key of any user with a value of his choice, but he does not have access to the master key. The Type II adversary A_{II} models the malicious KGC who has access to the master key, but he cannot replace the user's public key replacement. Padhye et al. claimed their scheme was secure against both of the two

types of adversaries. In this section, we will show that a Type I adversary A_I could generate a legal delegation of any warrant message and a legal proxy signature of any message.

3.1. Attack on the Delegation. Let OS be the original signer with identity ID_{OS} and the public key $pk_{OS} = (P_{OS}, K_{OS})$. Let A_I be a Type I adversary. A_I could generate a proxy signature of a message m and the warrant message m_w through the following steps.

- (1) A_I generates a random number $l_A \in Z_n^*$ and computes $h_{OS} = H_1(ID_{OS}, K_{OS})$ and $P'_{OS} = l_A \cdot P - K_{OS} - h_{OS} \cdot P_{pub}$.
- (2) A_I replaces $pk_{OS} = (P_{OS}, K_{OS})$ with $pk'_{OS} = (P'_{OS}, K_{OS})$.
- (3) A_I generates a random number $r_A \in Z_n^*$ and computes $R_A = r_A \cdot P$.
- (4) A_I computes $\sigma_A = (l_A e_A + r_A) \bmod n$ and sends the delegation (m_w, R_A, σ_A) to the proxy signer PS, where $e_A = H_2(m_w, ID_{OS}, K_{OS}, P'_{OS}, R_A)$.

Since $P'_{OS} = l_A \cdot P - K_{OS} - h_{OS} \cdot P_{pub}$ and $R_A = r_A \cdot P$, then we have

$$\begin{aligned}
 & e_A (P'_{OS} + K_{OS} + h_{OS} P_{pub}) + R_A \\
 &= e_A (l_A \cdot P - K_{OS} - h_{OS} \cdot P_{pub} + K_{OS} + h_{OS} P_{pub}) \\
 & \quad + r_A \cdot P \\
 &= e_A \cdot l_A \cdot P + r_A \cdot P = (e_A \cdot l_A + r_A) P \\
 &= \sigma_A \cdot P.
 \end{aligned} \tag{1}$$

Therefore, the generated delegation could pass the proxy signer PS's verification and A_I generates a delegation of a warrant message m_w successfully.

3.2. Attack on the Proxy Signature. Let OS be the original signer with identity ID_{OS} and the public key $pk_{OS} = (P_{OS}, K_{OS})$. Let PS be the original signer with identity ID_{PS} and the public key $pk_{PS} = (P_{PS}, K_{PS})$. Let A_I be a Type I adversary. A_I could generate a delegation of a warrant message m_w through the following steps.

- (1) A_I generates two random number $l_{OS}, l_{PS} \in Z_n^*$ and computes $h_{OS} = H_1(ID_{OS}, K_{OS})$, $h_{PS} = H_1(ID_{PS}, K_{PS})$, $P'_{OS} = l_{OS} \cdot P - K_{OS} - h_{OS} \cdot P_{pub}$, and $P'_{PS} = l_{PS} \cdot P - K_{PS} - h_{PS} \cdot P_{pub}$.
- (2) A_I replaces $pk_{OS} = (P_{OS}, K_{OS})$ and $pk_{PS} = (P_{PS}, K_{PS})$ with $pk'_{OS} = (P'_{OS}, K_{OS})$ and $pk'_{PS} = (P'_{PS}, K_{PS})$ separately.
- (3) A_I generates a random number $r_{OS} \in Z_n^*$ and computes $R_{OS} = r_{OS} \cdot P$, $e_{OS} = H_2(m_w, ID_{OS}, K_{OS}, P'_{OS}, R_A)$ and $\sigma_{OS} = (l_{OS} e_{OS} + r_{OS}) \bmod n$.
- (4) A_I generates a random number $r_{PS} \in Z_n^*$ and computes $R_{PS} = r_{PS} \cdot P$, $t_{PS} = (m \parallel H_4(m) + (R_{PS})_x) \bmod n$, $e_{PS} = H_3(m_w, ID_{PS}, K_{PS}, P'_{PS})$, $e = H_5(t_{PS}, P'_{OS}, K_{PS}, P'_{PS})$ and $\sigma_{PS} = (r_{PS} - e(\sigma_{OS} + e_{PS} l_{PS})) \bmod n$.

- (5) PS outputs $(m_w, \sigma_{OS}, R_{OS}, t_{PS}, \sigma_{PS})$ as the proxy signature.

Since $P'_{OS} = l_{OS} \cdot P - K_{OS} - h_{OS} \cdot P_{pub}$, $P'_{PS} = l_{PS} \cdot P - K_{PS} - h_{PS} \cdot P_{pub}$, $\sigma_{OS} = (l_{OS} e_{OS} + r_{OS}) \bmod n$, $R_{OS} = r_{OS} \cdot P$ and $R_{PS} = r_{PS} \cdot P$, then we have

$$\begin{aligned}
 & \sigma_{PS} \cdot P + e (e_{OS} (P'_{OS} + K_{OS} + h_{OS} \cdot P_{pub}) \\
 & \quad + R_{OS} + e_{PS} (P'_{PS} + K_{PS} + h_{PS} \cdot P_{pub})) \\
 &= (r_{PS} - e(\sigma_{OS} + e_{PS} l_{PS})) \cdot P \\
 & \quad + e (e_{OS} (l_{OS} \cdot P - K_{OS} - h_{OS} \cdot P_{pub} + K_{OS} + h_{OS} \cdot P_{pub}) \\
 & \quad + r_{OS} \cdot P + e_{PS} (l_{PS} \cdot P - K_{PS} - h_{PS} \cdot P_{pub} + K_{PS} + h_{PS} \cdot P_{pub})) \\
 &= (r_{PS} - e(\sigma_{OS} + e_{PS} l_{PS})) \cdot P \\
 & \quad + e (e_{OS} l_{OS} \cdot P + r_{OS} \cdot P + e_{PS} l_{PS} \cdot P) \\
 &= r_{PS} \cdot P - e(\sigma_{OS} + e_{PS} l_{PS}) \cdot P \\
 & \quad + e ((l_{OS} e_{OS} + r_{OS}) + e_{PS} l_{PS}) \cdot P \\
 &= r_{PS} \cdot P - e(\sigma_{OS} + e_{PS} l_{PS}) \cdot P \\
 & \quad + e (\sigma_{OS} + e_{PS} l_{PS}) \cdot P \\
 &= r_{PS} \cdot P.
 \end{aligned} \tag{2}$$

Therefore, the generated signature could pass the verification and A_I generates a signature successfully.

4. Conclusion

In this paper, we have demonstrated that Padhye et al.'s CLPS scheme with message recovery is not secure against the Type I adversary by giving concrete attacks. The analysis shows their scheme is not secure for practical applications. We will try to give a countermeasure to overcome weaknesses in their scheme in the future.

Abbreviations

p :	A large prime number
F_p :	A finite field
E :	An elliptic curve defined by the equation $y^2 = x^3 + ax + b$, where $a, b \in F_p$ and $\Delta = 4a^3 + 27b^2 \neq 0$
G :	The group consists of points on E and the infinite point O
n :	The order of G , where n is a large prime number
P :	A generator of group G
(x, P_{pub}) :	The master/public key pair of the key generation centre (KGC)

U : A user
 ID_U : The identity of U
 d_U : The partial private key of U
 x_U : The secret value of U
 sk_U : The private key of U
 pk_U : The public key of U
 OS : The original signer
 PS : The proxy signer
 D_{PS} : The proxy key.

Acknowledgments

The authors thank the editors and the anonymous reviewers for their valuable comments. This research was supported by National Natural Science Foundation of China (nos. 61202447 and 61201180), Natural Science Foundation of Hebei Province of China (no. F2013501066), Northeastern University at Qinhuangdao Science and Technology Support Program (no. xnk201307), Beijing Natural Science Foundation (no. 4132055), and Excellent Young Scholars Research Fund of Beijing Institute of Technology.

References

- [1] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, no. 9, pp. 1338–1354, 1996.
- [2] S. F. Tzeng and M. S. Hwang, "Digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem," *Computer Standards and Interfaces*, vol. 26, no. 2, pp. 61–71, 2004.
- [3] M. S. Hwang, C. C. Lee, and S. F. Tzeng, "A new proxy signature scheme for a specified group," *Information Sciences*, vol. 227, pp. 102–115, 2013.
- [4] D. He, Y. Chen, J. Chen, R. Zhang, and W. Han, "A new two-round certificateless authenticated key agreement protocol without bilinear pairings," *Mathematical and Computer Modelling*, vol. 54, no. 11–12, pp. 3143–3152, 2011.
- [5] D. He, J. Chen, and J. Hu, "A pairing-free certificateless authenticated key agreement protocol," *International Journal of Communication Systems*, vol. 25, no. 2, pp. 221–230, 2012.
- [6] D. He, S. Padhye, and J. Chen, "An efficient certificateless two-party authenticated key agreement protocol," *Computers & Mathematics with Applications*, vol. 64, no. 6, pp. 1914–1926, 2012.
- [7] D. He, J. Chen, and R. Zhang, "An efficient and provably-secure certificateless signature scheme without bilinear pairings," *International Journal of Communication Systems*, vol. 25, no. 11, pp. 1432–1442, 2012.
- [8] D. He, Y. Chen, and J. Chen, "A provably secure certificateless proxy signature scheme without pairings," *Mathematical and Computer Modelling*, vol. 57, no. 9–10, pp. 2510–2518, 2013.
- [9] D. He, B. Huang, and J. Chen, "A new certificateless short signature scheme," *IET Information Security*. In press.
- [10] X. Li, K. Chen, and L. Sun, "Certificateless signature and proxy signature schemes from bilinear pairings," *Lithuanian Mathematical Journal*, vol. 45, no. 1, pp. 76–83, 2005.
- [11] W. Yap, S. Heng, and B. Goi, "Cryptanalysis of some proxy signature schemes without certificates," in *Proceedings of the 1st Workshop on Information Security Theory and Practices (WISTP '07)*, vol. 4462 of *Lecture Notes in Computer Science*, pp. 115–126, Springer, Heraklion, Greece, May 2007.
- [12] R. Lu, D. He, and C. Wang, "Cryptanalysis and improvement of a certificateless proxy signature scheme from bilinear pairings," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07)*, pp. 285–290, Qingdao, China, July 2007.
- [13] H. Chen, F.-T. Zhang, and R.-S. Song, "Certificateless proxy signature scheme with provable security," *Journal of Software*, vol. 20, no. 3, pp. 692–701, 2009.
- [14] H. Xiong, F. Li, and Z. Qin, "A provably secure proxy signature scheme in certificateless cryptography," *Informatica*, vol. 21, no. 2, pp. 277–294, 2010.
- [15] L. Zhang, F. Zhang, and Q. Wu, "Delegation of signing rights using certificateless proxy signatures," *Information Sciences*, vol. 184, pp. 298–309, 2012.
- [16] S. Seo, K. Choi, J. Hwang, and S. Kim, "Delegation of signing rights using certificateless proxy signatures," *Information Sciences*, vol. 188, pp. 321–337, 2012.
- [17] S. Padhye and N. Tiwari, "ECDLP-based certificateless proxy signature scheme with message recovery," *Transactions on Emerging Telecommunications Technologies*, 2012.