Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 454393, 10 pages
http://dx.doi.org/10.1155/2014/454393

Research Article

A Provably Secure Proxy Signcryption Scheme Using Bilinear Pairings

Nai-Wei Lo and Jia-Lun Tsai

Department of Information Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan

Correspondence should be addressed to Nai-Wei Lo; nwlo@cs.ntust.edu.tw

Received 8 November 2013; Accepted 19 April 2014; Published 19 May 2014

Academic Editor: Ferenc Hartung

As people in modern societies are busier than any human era and computer network has profound impact on how people work and live through fast and convenient information exchange, people need more help from each other to accomplish more work via network connections in limited period of time. Therefore, privilege delegation mechanism has become a necessary service in modern enterprises and organizations. Proxy signcryption scheme provides a secure privilege delegation mechanism for a person to delegate his privilege to his proxy agent to accomplish things. In 2010, Lin et al. had proposed an efficient signcryption scheme using bilinear pairings. However, we found that the proxy signcryption scheme of Lin et al. is vulnerable to the chosen warrant attack. A provably secure proxy signcryption scheme using bilinear pairings is introduced accordingly. In terms of performance efficiency, the proposed scheme is superior to other existing schemes. In addition, a new security model is proposed to describe proxy signcryption scheme; based on the security model we show that the proposed scheme is provably secure in terms of indistinguishability under adaptive chosen ciphertext attack (IND-CCA2), unforgeability under adaptive chosen message attack (EF-CMA), and unforgeability under adaptive chosen warrant attack (EF-CWA).

1. Introduction

Since Diffie and Hellman proposed the concept of public key cryptosystem [1] in 1976, public key cryptosystems have been widely used for constructing secure network applications and communication systems. Generally, public key cryptosystems can be divided into two categories: public key encryption schemes [2–4] and signature schemes [2, 5, 6]. Public key encryption schemes are usually adopted to assure that the content of transmitted messages cannot be learned by an adversary without knowledge of the receiver's private key. Signature schemes are mainly used to assure that received messages at the destination party are not modified or falsely generated by an adversary. With rapid evolved Internet environment and more complicated business flow processes, secure privilege delegation mechanism has become a necessary function for enterprises, organizations, and even every modern citizen. New application demands such as online proxy auction, digital contract signing, and work transfer for deputy all require privilege delegation mechanism from time to time to help people delegate their authorities to someone or a group of people in order to accomplish certain work in time. Therefore, traditional public key cryptosystems [7–9] may not be able to meet the needs for these newly developed applications in terms of security robustness and operation efficiency.

The concept of proxy signature scheme was first proposed by Mambo et al. [10] in 1996. A proxy signature scheme allows the original signer to delegate his/her signing authority to a proxy signer. Once the proxy signer gains the delegated authority from the original signer, the proxy signer can generate a proxy signature on behalf of the original signer. Proxy signature schemes have been adopted in many practical applications, particularly in distributed systems and mobile agent-based systems where the delegation of user authority is commonly applied. In general, proxy delegation can be

divided into three types: full delegation, partial delegation, and delegation by warrant. In recent years, several proxy signature schemes have been proposed [8–16].

There are occasions in which applications with message transmission feature have to achieve confidentiality, integrity, authenticity, and nonrepudiation simultaneously. In 1997, Zheng first proposed a signcryption scheme in [17] to achieve these security requirements at the same time. The proposed signcryption scheme only allows the designated recipient to recover the original message from the received signcrypted ciphertext generated by the signer and then to verify the validity of this recovered message. Since then, various signcryption schemes were proposed [17–22].

In 1999, the concept of proxy signcryption scheme was first introduced by Gamage et al. [23]. Proxy signcryption scheme is subcategorized under signcryption scheme. Proxy signcryption scheme elaborates on the merits of signcryption and proxy signature. In a proxy signcryption scheme, an original signer can generate a proxy credential to delegate his/her signing authority to a proxy signer. Then, the proxy signer can generate a signcrypted message on behalf of the original signer. Only the recipient has the ability to recover the content of this signcrypted message and then to verify the validity of this recovered message content. In case a dispute occurs from the repudiation of the proxy signer or the original signer, the message recipient can announce the proxy signature to a trusted third party for public verification without extra computational cost. Proxy signcryption schemes can be used in applications such as online proxy auction and business contract signing.

Recently, bilinear pairing [24, 25] from elliptic curves is widely adopted to develop new public key cryptosystems [26–36]. Accordingly many researchers have utilized bilinear pairings to construct pairings-based proxy signcryption schemes [26–28, 30, 31]. In 2010, Lin et al. proposed an efficient proxy signcryption scheme [31] using bilinear pairings. The scheme of Lin et al. is the first one to propose a public verification mechanism for the message recipient to prove the proxy signer or the original signer is dishonest when a dispute occurs between message signers and message recipient. In addition, only four bilinear pairing operations are required in their scheme. To prove security strength of their proxy signcryption scheme, Lin et al. also give a security model for proxy signcryption scheme and then prove their scheme is secure in terms of IND-CCA2 and EF-CMA under random oracle model.

1.1. Contribution. This paper discovers that the signcryption scheme of Lin et al. [31] is vulnerable to two forgery attacks because the proxy credential generated from the original signer is not secure against the chosen warrant attack. In addition, the security model of Lin et al. did not consider unforgeability of generated proxy credential. A new proxy signcryption scheme using bilinear pairings is introduced in which the proposed scheme remedies the vulnerabilities of Lin et al.'s scheme and achieves better performance in terms of computing cost when comparing with other existing schemes. A new security model for proxy signcryption scheme is also presented and used to prove that the proposed scheme is secure in terms of indistinguishability under adaptive chosen ciphertext attack (IND-CCA2), unforgeability under adaptive chosen warrant attack (EF-CWA), and unforgeability under adaptive chosen message attack (EF-CMA) in random oracle.

2. Preliminaries

This section introduces bilinear pairings, the definition of proxy signcryption scheme, and mathematical problems used for cryptography as follows.

2.1. Bilinear Pairings. The properties of bilinear pairings are introduced as follows. Let G_1 be an additive cyclic group, let G_2 be a multiplicative cyclic group, and let P be a generator of G_1, where G_1 and G_2 have the prime order q. A bilinear pairing equation e: $G_1 \times G_1 \rightarrow G_2$ satisfies the following properties:

(1) bilinear: given $P_1, P_2, Q_1, Q_2 \in G_1$, $e(P_1 + P_2, Q_1) = e(P_1, Q_1)e(P_2, Q_1)$ and $e(P_1, Q_1 + Q_2) = e(P_1, Q_1)e(P_1, Q_2)$;

besides, given $a, b \in Z_q$, $e(aP, bQ) = e(abP, Q) = e(P, abQ) = e(P, Q)^{ab} = e(bP, aQ)$;

(2) nondegenerate: there exists $P \in G_1$ and $Q \in G_1$ such that $e(P, Q) \neq 1$, where 1 is the identity element of G_2;

(3) computable: for any $P, Q \in G_1$, the value $e(P, Q)$ is efficiently computed.

2.2. The Definition of Proxy Signcryption Scheme. The roles of a proxy signcryption scheme can be divided into three parties: an original signer U_O, a proxy signer U_p, and a designated recipient U_v. In a proxy signcryption scheme, an original signer generates a proxy credential to delegate his/her signing authority to a proxy signer. The proxy signer then generates a signcrypted message by using the proxy credential and his/her secret key. Next, the proxy signer sends the signcrypted message to a designated recipient through insecure network. Upon receiving the signcrypted message, only the designated recipient can recover the message content from the signcrypted message and then verify its validity. If a dispute occurs later, the message recipient can announce the proxy signature for public verification without extra computational cost. A proxy signcryption scheme consists of the following algorithms.

(i) *Setup*. This algorithm takes a secure parameter 1^k as input and then returns public parameters of system *params*.

(ii) *Proxy-Credential-Generation (PCG)*. This algorithm takes the private key of original signer *osk* and a warrant m_w as input and then returns a proxy credential σ_p on the warrant m_w for the proxy signer.

(iii) *Signcrypted-Message-Generation (SMG)*. This algorithm takes a message m, a proxy credential σ_p, a warrant m_w, a private key of proxy signer *psk*,

and a proxy credential as input and then outputs a signcrypted message σ_s.

(iv) *Signature-Recovery-and-Verification (SRV)*. This algorithm takes a signcrypted message σ_s, the private key of designated recipient *rsk*, a warrant m_w, and the public key pair of original signer and proxy signer (*opk*, *ppk*) and then returns a plaintext m and its converted ordinary proxy signature σ if the signcrypted message σ_s is valid. Otherwise, this algorithm returns an error symbol ¶.

2.3. Mathematical Problems for Cryptography. We introduce mathematical problems applied within our scheme for security as follows.

Discrete Logarithm Problem (DLP). Given $\{P, Q = aP\} \in G_1$, it is hard to find an integer $a \in Z_q^*$ from $Q = aP$.

Bilinear Diffie-Hellman Problem (BDHP). Given an instance $\{P, A = aP, B = bP, C = cP\} \in G_1$ for some $a, b, c \in Z_q^*$, it is hard to compute $e(P, P)^{abc}$, where P is the generator.

3. Review and Cryptanalysis of the Proxy Signcryption Scheme of Lin et al.

This section briefly reviews the proxy signcryption scheme of Lin et al. [31] and then shows that their scheme is vulnerable to two forgery attacks as follows.

3.1. Review of the Proxy Signcryption Scheme of Lin et al. We briefly review the proxy signcryption scheme of Lin et al. [31] in this subsection. Details of each algorithm are described as follows.

Setup. Let G_1 and G_2 be two groups of the same prime order q, where P is a generator of G_1. First of all, the system authority SA chooses a pairing function e: $G_1 \times G_1 \rightarrow G_2$ and three collision-resistant hash functions: $h_1 : \{0,1\}^k \times G_1 \rightarrow Z_q$, $h_2 : G_1 \rightarrow G_1$, and $h_3 : G_2 \times G_1 \rightarrow \{0,1\}^k$. Next, SA publishes ($G_1$, G_2, q, P, e, $e(P,P)$, h_1, h_2, h_3) as public parameters. Each signer also chooses a random number x_i as his/her private key and then computes the corresponding public key $Y_i = x_i P$.

Proxy-Credential-Generation. When an original signer U_O wants to delegate his/her signing privilege to a proxy signer U_p, the original signer U_O chooses a random number $d \in Z_q$ and then generates a proxy credential (σ, N, m_w) by computing the following equations:

$$\begin{aligned} N &= dP, \\ \sigma &= x_o + d\left(m_w\right) \bmod q, \end{aligned} \tag{1}$$

where m_w is the warrant including the identities of the original signer U_O and the proxy signer U_p. Next, the original signer U_O sends the proxy credential (σ, N, m_w) to a proxy signer U_p. After receiving the proxy credential (σ, N, m_w), the proxy signer U_p verifies the validity of the received proxy credential by computing the values at both sides of the equality symbol in the following equation:

$$\sigma P = Y_o + m_w N. \tag{2}$$

If (2) holds with the two computed values, the proxy credential (σ, N, m_w) is accepted; otherwise, the proxy signer U_p requests the original signer to resend the proxy credential (σ, N, m_w).

Signcrypted-Message-Generation. When the proxy signer U_p wants to generate a signcrypted message on a plaintext message $m \in_R \{0,1\}^k$, he/she computes

$$\begin{aligned} R &= rP, \\ S &= r\left(h_1\left(m, R\right) + x_p + \sigma\right)^{-1} P, \\ V &= e\left(h_2\left(\sigma Y_v\right), x_p Y_v\right), \\ X &= E_V\left(S\right), \\ Y &= h_3\left(V, R\right) \oplus m, \end{aligned} \tag{3}$$

where r is a random number and E_V is the symmetric encryption function with the secret key V. Next, the proxy signer U_p sends the signcrypted message (R, X, Y, N) and m_w to the designated recipient U_v.

Signature-Recovery-and-Verification. For a signcrypted message (R, X, Y, N), the designated recipient U_v can recover the message m and the proxy signature (S, R, N) by computing the following equations:

$$\begin{aligned} V &= e\left(h_2\left(x_v\left(Y_o + m_w N\right)\right), x_v Y_p\right), \\ m &= h_3\left(V, R\right) \oplus Y, \\ S &= D_V\left(X\right), \end{aligned} \tag{4}$$

where D_V is the symmetric decryption function with the key V. Next, the recipient U_v verifies the validity of the proxy signature by computing the values at both sides of the equality symbol in the following equation:

$$e\left(h_1\left(m, R\right) P + Y_p + Y_o + m_w N, S\right) = e\left(P, R\right). \tag{5}$$

If (5) holds with the two computed values, the proxy signature (S, R, N) is accepted by the designated recipient U_v; otherwise, the proxy signature (S, R, N) is rejected. In case a dispute occurs later, the designated recipient U_v can reveal the proxy signature as well as the message m and the warrant m_w to any trusted third party. A trusted third party can use (5) to perform an evaluation task and know whether the proxy signer U_p is dishonest or not.

3.2. Cryptanalysis of the Scheme of Lin et al. Two forgery attacks on the scheme of Lin et al. are discovered by utilizing security weakness of the proxy credential through chosen

warrant attack. Details of two forgery attacks are addressed as follows.

Forgery Attack 1. We show that a malicious proxy signer can forge any valid proxy credential on his/her chosen warrant m'_w if he/she obtains a valid proxy credential as follows.

Assume that a malicious proxy signer, who has a valid proxy credential (σ, N, m_w) on a warrant m_w, wants to forge a valid proxy credential (σ, N', m'_w) on his/her chosen warrant m'_w. The malicious proxy signer needs to generate

$$N' = \frac{m_w}{m'_w} N, \tag{6}$$

where m'_w is his/her chosen warrant. Now, the forged proxy credential (σ, N', m'_w) is created by the malicious proxy signer without knowledge of the private key of the original signer.

In the following, we show that the forged proxy credential (σ, N', m'_w) can pass the proxy credential verification equation shown in (2):

$$\begin{aligned} \sigma P &= Y_o + m'_w N' \\ &= Y_o + m'_w \frac{m_w}{m'_w} \cdot dP \\ &= Y_o + m_w N, \end{aligned} \tag{7}$$

where $N' = (m_w/m'_w)N$, $N = dP$.

Forgery Attack 2. We show that any adversary can forge a proxy signature (S', R', N') on his/her chosen message m' and his/her chosen warrant m'_w without knowledge of any valid proxy credential (σ, N, m_w), the private key of the original signer, and the private key of the proxy signer as follows.

Assume that an adversary A wants to forge a proxy signature (S', R', N') on his/her chosen message m' and warrant m'_w. The adversary A first computes

$$N' = \left(m'_w\right)^{-1} \left(-Y_P - Y_o\right), \tag{8}$$

$$R' = r'P, \tag{9}$$

$$S' = h_1\left(m', R'\right)^{-1} R', \tag{10}$$

where r' is a random number. Now, the adversary A forges a valid proxy signature (S', R', N') on his/her chosen message m' and warrant m'_w. In consequence, the proxy signcryption scheme of Lin et al. does not support nonrepudiation.

In the following, we show that the forged proxy signature (S', R', N') can pass the proxy signature verification equation shown in (5):

$$\begin{aligned} & e\left(h_1\left(m', R'\right) P + Y_p + Y_o + m'_w N', S'\right) \\ &= e\left(h_1\left(m', R'\right) P + Y_p + Y_o - Y_o - Y_p, S'\right) \quad \text{by (8)} \\ &= e\left(h_1\left(m', R'\right) P, h_1\left(m', R'\right)^{-1} R'\right) \quad \text{by (10)} \\ &= e\left(P, R'\right). \end{aligned} \tag{11}$$

4. The Proposed Scheme

This section presents our efficient proxy signcryption scheme. Details of each algorithm are described as follows.

Setup. Let G_1 and G_2 be two groups of the same prime order q and let P be a generator of G_1. In the beginning, system authority SA chooses a pairing function $e: G_1 \times G_1 \rightarrow G_2$ and four collision-resistant one-way hash functions: $h_1 : \{0,1\}^k \times G_1 \rightarrow Z_q$, $h_2 : \{0,1\}^k \times G_1 \rightarrow Z_q$, $h_3 : G_1 \rightarrow G_1$, and $h_4 : G_2 \times G_1 \rightarrow \{0,1\}^k$. Then, SA publishes $(G_1, G_2, q, P, e, e(P,P), h_1, h_2, h_3, h_4)$ as its public parameters. Each signer also chooses a random number x_i as his/her private key and then computes his/her corresponding public key $Y_i = x_i P$.

Proxy-Credential-Generation. Assume that an original signer U_O wants to delegate his/her signing authority to a proxy signer; he/she first computes

$$R_1 = r_1 P, \tag{12}$$

$$s_o = x_o + h_1\left(m_w, R_1\right) r_1 \bmod q, \tag{13}$$

where r_1 is a random number and m_w is the warrant. The original signer U_O then sends the proxy credential (R_1, s_o, m_w) to the proxy signer U_p via a secure channel. Upon receiving the proxy credential (R_1, s_o, m_w), the proxy signer U_p can verify its validity by computing the values at both sides of the equality symbol in the following equation:

$$s_o P = Y_o + h_1\left(m_w, R_1\right) R_1. \tag{14}$$

If (14) holds with the two computed values, the proxy credential is accepted; otherwise, the proxy credential signature is rejected. In the following, we show the derivation and verification process for (14):

$$\begin{aligned} s_o P &= \left(x_o + h_1\left(m_w, R_1\right) r_1\right) P \quad \text{by (13)} \\ &= x_o P + h_1\left(m_w, R_1\right) r_1 P \\ &= Y_o + h_1\left(m_w, R_1\right) R_1, \end{aligned} \tag{15}$$

where $x_o P = Y_o$ and $R_1 = r_1 P$.

Signcrypted-Message-Generation. In order to generate a signcrypted message on his/her chosen message m, the proxy signer U_p computes

$$R_2 = r_2 P, \tag{16}$$

$$s_p = \frac{1}{r_2 \cdot h_2\left(m, m_w, R_1, R_2\right) + x_p + s_o} P, \tag{17}$$

$$V = e\left(h_3\left(s_o Y_v\right), x_p Y_v\right), \tag{18}$$

$$X = E_V\left(s_p\right), \tag{19}$$

$$Y = h_4\left(V, R_2\right) \oplus m. \tag{20}$$

Then, the proxy signer U_p sends the signcrypted message (R_1, R_2, X, Y) and the warrant m_w to the designated recipient U_v.

Signature-Recovery-and-Verification. Upon receiving a signcrypted message (R_1, R_2, X, Y), the recipient U_v first recovers the message m by computing the following equations:

$$V = e\left(h_3\left(x_v\left(Y_o + h_1\left(m_w, R_1\right)R_1\right)\right), x_v Y_p\right),$$
$$m = h_4\left(V, R_2\right) \oplus Y. \tag{21}$$

Next, the recipient U_v computes

$$s_p = D_V\left(X\right) \tag{22}$$

and then verifies the validity of the proxy signature (R_1, R_2, s_p) by computing the values at both sides of the equality symbol in the following equation:

$$e\left(h_2\left(m, m_w, R_1, R_2\right)R_2 + Y_p + Y_o + h_1\left(m_w, R_1\right)R_1, s_p\right) = e\left(P, P\right). \tag{23}$$

If (23) holds with the two computed values, the recipient U_v accepts the proxy signature; otherwise, he/she rejects the proxy signature. Notice that the value of $e(P, P)$ is precomputed as one of the public parameters during system setup phase; therefore, the computational cost for the value of $e(P, P)$ can be ignored here.

If a dispute between the proxy signer and the recipient occurs, the designated recipient U_v can send the message m, the warrant m_w, and the proxy signature (R_1, R_2, s_p) to any trusted third party. A trusted third party can use (23) to perform an evaluation task and know whether the proxy signer U_p is dishonest.

In the following, we show the derivation and verification process for (23):

$$\begin{aligned} & e\left(h_2\left(m, m_w, R_1, R_2\right)R_2 + Y_p + Y_o + h_1\left(m_w, R_1\right)R_1, s_p\right) \\ & = e\left(h_2\left(m, m_w, R_1, R_2\right)R_2 + Y_p + Y_o + h_1\left(m_w, R_1\right)R_1, \right. \\ & \qquad \left. \frac{1}{h_2\left(m, m_w, R_1, R_2\right)r_2 + x_p + x_o + h_1\left(m_w, R_1\right)r_1}P\right) \\ & \qquad \text{by (12), (14), (16), and (17)} \\ & = e\left(P, P\right). \end{aligned} \tag{24}$$

5. Security Analysis

In the literature of Lin et al. [31], they had proposed a security model for proxy signcryption scheme. However, the security model of Lin et al. is incomplete as unforgeability of proxy credential was not considered.To prove security robustness of the proposed proxy signcryption scheme, we propose a new security model for proxy signcryption scheme. Consequently, this proposed security model is applied to prove that our proposed scheme is secure in terms of IND-CCA2, EF-CWA, and EF-CMA under random oracle.

5.1. Security Model. Three security requirements for proxy signcryption scheme are message confidentiality, proxy credential unforgeability, and proxy signcryption unforgeability. We give a new security model for proxy signcryption scheme as follows.

Definition 1 (confidentiality). A proxy signcryption scheme achieves confidentiality under adaptive chosen ciphertext attacks if no adversary $\mathcal{A}$ can play the following game with a challenger $\mathcal{B}$ and win this game within a probabilistic polynomial time period by possessing nonnegligible advantage.

Setup. At the beginning, $\mathcal{B}$ runs this algorithm to generate all public parameters *params* and then publishes them. Thus, $\mathcal{A}$ can obtain these public parameters *params*.

Phase 1. An adversary $\mathcal{A}$ has the ability to execute the following queries adaptively.

(i) *Proxy-Credential-Generation (PCG) Query*. When $\mathcal{A}$ calls the PCG query with his/her chosen warrant m_w, $\mathcal{B}$ returns the corresponding proxy credential to $\mathcal{A}$.

(ii) *Signcrypted-Message-Generation (SMG) Query*. When $\mathcal{A}$ calls the SMG query with his/her chosen message m, $\mathcal{B}$ first generates the proxy signature for the message m. Then, $\mathcal{B}$ generates the signcrypted message δ and then returns it to $\mathcal{A}$.

(iii) *Signature-Recovery-and-Verification (SRV) Query*. When $\mathcal{A}$ calls the SRV query, upon receiving a signcrypted message δ and its warrant m_w from $\mathcal{A}$, $\mathcal{B}$ returns a plaintext message m and its convertible proxy signature if the signcrypted message is valid. Otherwise, $\mathcal{B}$ returns an error symbol $\perp$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ sends two plaintext messages m_0 and m_1 to $\mathcal{B}$, where these two messages with the same length are chosen by the adversary $\mathcal{A}$. Next, $\mathcal{B}$ flips a coin $\lambda \rightarrow \{0, 1\}$ and then generates a signcrypted message δ^* for the message m_λ. $\mathcal{B}$ sends the signcrypted message δ^* to $\mathcal{A}$ as a challenge.

Phase 2. $\mathcal{A}$ has the ability to call several new queries defined in Phase 1. Once $\mathcal{A}$ receives the signcrypted message δ^*, $\mathcal{A}$ can call multiple queries except SRV queries to guess which message, m_0 or m_1, is signcrypted inside δ^*.

Guess. Finally, $\mathcal{A}$ outputs a bit λ' as its guess. If $\lambda' = \lambda$, $\mathcal{A}$ wins this game, where the advantage of $\mathcal{A}$ to win the game is $\mathrm{Adv}(A) = |\Pr[\lambda' = \lambda] - 1/2|$.

Definition 2 (proxy credential unforgeability). A proxy signcryption scheme achieves proxy credential unforgeability under adaptive chosen warrant attacks if no adversary $\mathcal{A}$ can play the following game with a challenger $\mathcal{B}$ and win

this game within a probabilistic polynomial time period by possessing nonnegligible advantage.

Setup. In this algorithm, $\mathscr{B}$ generates all public parameters *params* and then publishes these parameters. Thus, these parameters *params* can be learned by $\mathscr{A}$.

Phase 1. $\mathscr{A}$ can call multiple PCG queries defined in Phase 1 of Definition 1 with his/her chosen warrant m_w.

Forgery. The adversary $\mathscr{A}$ forges a valid proxy credential δ' based on his/her chosen warrant m_w' without calling any PCG query.

Definition 3 (proxy signcryption unforgeability). A proxy signcryption scheme achieves proxy signcryption unforgeability under adaptive chosen message attacks if no adversary $\mathscr{A}$ can play the following game with a challenger $\mathscr{B}$ and win this game within a probabilistic polynomial time period by possessing nonnegligible advantage.

Setup. First of all, $\mathscr{B}$ runs the setup algorithm to generate all public parameters *params* and then publishes these parameters. Therefore, $\mathscr{A}$ can obtain these parameters *params*.

Phase 1. In this phase, $\mathscr{A}$ can ask $\mathscr{B}$ to generate the proxy signature with his/her chosen message m by calling PSG queries defined in Phase 1 of Definition 1.

Forgery. The adversary $\mathscr{A}$ forges a valid proxy signature δ'' based on his/her chosen message m' without calling any PSG query.

5.2. Security Proof. This subsection shows the proposed scheme is secure against the chosen ciphertext attack (IND-CCA2), the adaptive chosen warrant attack (EF-CWA), and the adaptive chosen message attack (EF-CMA) under random oracle as follows.

Theorem 4 (confidentiality). *Let t_λ be the time for executing one bilinear pairing operation. If no adversary $\mathscr{A}$ can (t', ε')-break the bilinear Diffie-Hellman problem in probabilistic polynomial time, the proposed proxy signcryption scheme can $(t, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, q_{PCG}, q_{SMG}, q_{SRV}, \varepsilon)$-withstand the existential forgery under adaptive chosen ciphertext attack (IND-CCA2) in random oracle model, where*

$$\begin{aligned} \varepsilon' &\geq \left(2\varepsilon - q_{SRV}\left(2^{-k}\right)\right), \\ t' &\approx t + t_\lambda\left(q_{SMG} + 2q_{SRV}\right). \end{aligned} \tag{25}$$

Proof. Suppose that an algorithm $\mathscr{B}$ tries to resolve BDHP by taking (P, aP, bP, cP) as inputs. The algorithm $\mathscr{B}$ simulates itself as the challenger to serve $\mathscr{A}$ in the following game, where $\mathscr{A}$ can only ask at most q_{h_i} times of h_i oracles ($i = 1, 2, 3, 4$), q_{PCG} times of PCG query, q_{SMG} times of SMG query, and q_{SRV} times of SRV query within the period of probabilistic polynomial time t.

Setup. $\mathscr{B}$ runs the setup algorithm to generate all necessary public parameters (G_1, G_2, q, P, e, h_1, h_2, h_3, h_4) and then sends (G_1, G_2, q, P, e, h_1, h_2, h_3, h_4) and ($Y_p = aP$, $Y_v = bP$, $Y_o = wP$) to $\mathscr{A}$.

Phase 1. In this phase, $\mathscr{A}$ can call the following queries supported by $\mathscr{B}$.

(i) h_1 *Hash Query*. When $\mathscr{A}$ calls a h_1 hash query on his/her chosen warrant m_w and R_1, $\mathscr{B}$ first checks whether (m_w, R_1) exists in the h_1-list. If the pair indeed exists, $\mathscr{B}$ returns the existing v_1 to $\mathscr{A}$. Otherwise, $\mathscr{B}$ randomly selects a number $v_1 \in Z_p$, stores (m_w, R_1, v_1) into the h_1-list, and returns v_1 to $\mathscr{A}$.

(ii) h_2 *Hash Query*. If $\mathscr{A}$ sends the tuple (m, m_w, R_1, R_2) to the oracle h_2 as a query request, $\mathscr{B}$ first checks whether the tuple exists in the h_2-list. If it exists, $\mathscr{B}$ returns the existing v_2 to $\mathscr{A}$. Otherwise, $\mathscr{B}$ randomly selects a number $v_2 \in Z_p$, stores (m, m_w, R_1, R_2, v_2) into the h_2-list, and returns v_2 to $\mathscr{A}$.

(iii) h_3 *Hash Query*. If $\mathscr{A}$ calls a h_3 hash query with the value s_oY_v, $\mathscr{B}$ first checks whether this value exists in the h_3-list. If it exists, $\mathscr{B}$ returns the existing V_3 to $\mathscr{A}$. Otherwise, $\mathscr{B}$ returns $V_3 = v_3P$ to $\mathscr{A}$ and then stores the tuple (s_oY_v, v_3, V_3) into the h_3-list, where $v_3 \in Z_p$ is a random number.

(iv) h_4 *Hash Query*. When $\mathscr{A}$ calls a h_4 hash query with his own chosen value pair (V, R_2), $\mathscr{B}$ first checks whether this pair (V, R_2) exists in the h_4-list. If the pair exists, $\mathscr{B}$ returns the existing v_4 to $\mathscr{A}$. Otherwise, $\mathscr{B}$ generates and returns v_4 to $\mathscr{A}$ before storing the tuple (V, R_2, v_4) into the h_4-list, where $v_4 \in Z_p$ is a random number.

(v) *Proxy-Credential-Generation (PCG) Query*. When $\mathscr{A}$ calls this query with his own chosen warrant m_w, $\mathscr{B}$ first chooses two random numbers k_1 and v_1 and then computes $s_o = k_1P$ and $R_1 = v_1^{-1}(k_1P - wP)$, where $v_1 = h_1(m_w, R_1)$ has never been queried before. Then, $\mathscr{B}$ returns s_o and R_1 to $\mathscr{A}$.

(vi) *Signcrypted-Message-Generation (SMG) Query*. When $\mathscr{A}$ calls a SMG query with a message m, $\mathscr{B}$ first computes $s_p = k_2P$ and $R_2 = v_2^{-1}(k_2^{-1}P - aP - wP + v_1R_1)$, where k_2, v_1, and v_2 are three random numbers and $v_2 = h_2(m, m_w, R_1, R_2)$ and $v_1 = h_1(m_w, R_1)$ have never been queried before. Next, $\mathscr{B}$ calls an $h_3(s_o(bP))$ query to get (v_2,V_2). $\mathscr{B}$ then computes $V = e(v_2(aP), (bP))$ and the pair (X, Y), and then $\mathscr{B}$ returns the signcrypted message (R_1, R_2, X, Y) and the warrant m_w to $\mathscr{A}$.

(vii) *Signature-Recovery-and-Verification (SRV) Query*. When $\mathscr{A}$ calls a SRV query with a signcrypted message (R_1, R_2, X, Y) and its corresponding warrant m_w, $\mathscr{B}$ searches the h_4-list according to R_1 and R_2 and then recovers the message m. Next, $\mathscr{B}$ checks the

validity of associated proxy signature. If the validity of the proxy signature is confirmed, $\mathcal{B}$ returns the warrant m_w, the message m, and its proxy signature (R_1, R_2, s_p). Otherwise, $\mathcal{B}$ returns $\perp$ to indicate that the proxy signature is invalid.

Challenge. When $\mathcal{A}$ sends two plaintext messages m_0 and m_1 to $\mathcal{B}$, $\mathcal{B}$ first calls a PCG query to obtain $s_o^* = k_1 P$ and $R_1^* = v_1^{*-1}(k_1 P - wP)$, where k_1 and v_1 are two random numbers and $v_1^* = h_1(m_w^*, R_1^*)$ has never been queried before. Next, $\mathcal{B}$ flips a coin $\lambda \rightarrow \{0, 1\}$ to determine the value of λ and accordingly calls one PCG query and one SMG query to compute $R_2^* = v_2^{*-1}(k_2^{-1}P - aP - wP + v_1^* R_1)$, $s_p^* = k_2 P$, and $Y^* = v_3^* \oplus m_\lambda$, where $h_2(\sigma * Y_v) = z(cP)$, $V^* = e(z(cP), a(bP))$, $v_3^* = h_3(V^*, R_2^*)$, and $v_2^* = h_2(m^*, m_w^*, R_1^*, R_2^*)$. Finally, $\mathcal{B}$ returns the signcrypted message $\delta^* = \{R_1^*, R_2^*, X^*, V^*, m_w\}$ for the message m_λ.

Phase 2. $\mathcal{A}$ can call new queries defined in Phase 1, but $\mathcal{A}$ cannot call any SRV query for the signcrypted message $\delta^* = \{R_1^*, R_2^*, X^*, V^*, m_w\}$ to get the message m_λ.

Analysis of the Game. Let $\mathrm{SRV}_{\mathrm{ERR}}$ be the event that a SRV query returns the failure message $\perp$ for a valid signcrypted message $\delta = \{R_1, R_2, X, V, m_w\}$ during the entire game, let GP be the event that the entire game is perfect (i.e., no adversary can break the game.), and let QH_4^* be the event that indicates the total number of query times for h_4 oracle. The advantage of $\mathcal{A}$ is defined as $\mathrm{Adv}(A) = |\Pr[\lambda = \lambda'] - 1/2| \leq (1/2)\Pr[\neg \mathrm{GP}]$; in consequence, we have

$$\begin{aligned}\varepsilon &= \left|\Pr\left[\lambda = \lambda'\right] - \frac{1}{2}\right| \\ &\leq \left(\frac{1}{2}\right)\Pr\left[\neg \mathrm{GP}\right] \\ &= \left(\frac{1}{2}\right)\left(\Pr\left[\mathrm{QH}_4^* \vee \mathrm{SRV}_{\mathrm{ERR}}\right]\right) \\ &\leq \left(\frac{1}{2}\right)\left(\Pr\left[\mathrm{QH}_4^*\right] + \Pr\left[\mathrm{SRV}_{\mathrm{ERR}}\right]\right).\end{aligned} \tag{26}$$

In Phase 2 of our game, if $\mathcal{A}$ never calls h_4 hash query, the simulation will fail. Therefore, $\mathcal{B}$ would have nonnegligible probability to solve the bilinear Diffie-Hellman problem with probability at least

$$\varepsilon' \geq \left(2\varepsilon - q_{\mathrm{SRV}}\left(2^{-k}\right)\right). \tag{27}$$

Time complexity of the algorithm $\mathcal{B}$ is $t' \approx t + t_\lambda(q_{\mathrm{SMG}} + 2q_{\mathrm{SRV}})$, where t_λ is the time for executing one bilinear pairing operation. □

Theorem 5 (proxy credential unforgeability). *The proposed proxy signcryption scheme is secure against existential forgery under adaptive chosen warrant attacks (EF-CWA) if no adversary $\mathcal{A}$ can (t', ε')-break the DLP, where*

$$\begin{aligned}\varepsilon &\geq \frac{10\left(q_{PCG} + 1\right)\left(q_{PCG} + q_{h_1}\right)}{2^k}, \\ t' &\leq \frac{120686 q_{h_1} t}{\varepsilon}.\end{aligned} \tag{28}$$

Proof. We show that the proposed signcryption scheme can achieve security requirement for proxy credential unforgeability as follows, where $\mathcal{A}$ can only call at most q_{h_i} times of h_i oracles ($i = 1, 2, 3, 4$) and q_{PCG} times of PCG query within the period of probabilistic polynomial time t. An algorithm $\mathcal{B}$ can be constructed to break the DLP by playing the game with an adversary $\mathcal{A}$. In this game, the query algorithms and public parameters are the same as those ones defined in Theorem 4. Notice that each hash query has its own hash list to maintain corresponding tuples.

Setup. $\mathcal{B}$ runs the setup algorithm to generate all necessary public parameters ($Y_p = aP$, $Y_v = bP$, $Y_o = wP$, G_1, G_2, q, P, e, h_1, h_2, h_3, h_4) for the adversary $\mathcal{A}$.

Phase 1. In this phase, we allow $\mathcal{A}$ to call multiple PCG queries and h_1 queries as those ones defined in Phase 1 of the proof of Theorem 4.

Analysis of the Game. Suppose that $\mathcal{A}$ can only call at most q_{PCG} times of PCG query and q_{h_1} times of h_1 hash query, and the game simulation is perfect in random oracle. By applying the forking lemma, if $\varepsilon \geq 10(q_{\mathrm{PCG}} + 1)(q_{\mathrm{PCG}} + q_{h_1})/2^k$, let $\mathcal{B}$ output two proxy credentials (s_o, R_1, m_w) and (s_o^*, R_1, m_w) based on the same warrant m_w such that $h_1(m_w, R_1) \neq h_1^*(m_w, R_1)$. Then, $\mathcal{B}$ computes $(s_o - s_o^*)/(h_1(m_w, R_1) - h_1^*(m_w, R_1)) \bmod q$ as the value of w^*. According to the forking lemma, it indicates that $\mathcal{B}$ has the ability to solve one DLP instance within the period of time $t' \leq 120686 q_{h_1} t/\varepsilon$. □

Theorem 6 (proxy signcryption unforgeability). *The proxy signcryption scheme can $(t, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, q_{PCG}, q_{SMG}, q_{SRV}, \varepsilon)$-withstand adaptive chosen message attacks (EF-CMA) if no adversary $\mathcal{A}$, who plays the game with the challenger $\mathcal{B}$, can (t', ε')-break BDHP in probabilistic polynomial time t, where*

$$\begin{aligned}\varepsilon' &\geq \frac{\left(\varepsilon - \left(q_{h_3} + 1\right)/2^k\right)}{\left(q_{h_3} q_{h_4}\right)}, \\ t' &\approx t + t_\lambda q_{SMG}.\end{aligned} \tag{29}$$

Proof. Suppose that an adversary $\mathcal{A}$ can $(t, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, q_{\mathrm{PCG}}, q_{\mathrm{SMG}}, q_{\mathrm{SRV}}, \varepsilon)$-break the proposed scheme with nonnegligible advantage ε, where t indicates the maximum time consumption used to break the proposed scheme. In this game, the adversary $\mathcal{A}$ can call at most q_{h_i} times of h_i oracles ($i = 1, \ldots, 4$), q_{PCG} times of PCG query, and q_{SMG} times of SMG query. Then, an algorithm $\mathcal{B}$ can be constructed

to break the BDHP problem by playing the game with an adversary $\mathcal{A}$. The query algorithms and public parameters are the same as those ones defined in Theorem 4. Notice that each hash query has its own hash list to maintain corresponding tuples.

Setup. $\mathcal{B}$ runs this setup algorithm to generate all necessary public parameters ($Y_p = aP, Y_v = bP, Y_o = wP, G_1, G_2, q, P, e, h_1, h_2, h_3, h_4$) and then returns these public parameters to $\mathcal{A}$.

Phase 1. In this phase, $\mathcal{A}$ can call multiple PCG queries, SMG queries, and h_i ($i = 1, \ldots, 4$) queries as those ones defined in Phase 1 of the proof of Theorem 4.

Analysis of the Game. In the following, we prove that if an adversary $\mathcal{A}$ can break the proposed scheme, then there is an algorithm $\mathcal{B}$ which can break the BDHP problem. Assume that the adversary $\mathcal{A}$ can call at most q_{PCG} times of PCG query and q_{h_i} times of h_i hash queries ($i = 1, \ldots, 4$). Let SM_V be the event that the adversary $\mathcal{A}$ can forge a valid signature and let QH_3 and QH_4 be the events that indicate the total number of query times for h_3 and h_4 queries by the adversary $\mathcal{A}$, respectively. Obviously, the probability that the adversary $\mathcal{A}$ can correctly guess the hash value without querying h_3 or h_4 hash queries is less than 2^{-k}. Then, we have the following inequality:

$$\begin{aligned}
\varepsilon &= \Pr\left[\mathrm{SM}_V\right] \\
&= \Pr\left[\mathrm{SM}_V \mid \mathrm{QH}_3\right] + \Pr\left[\mathrm{SM}_V \mid \neg\mathrm{QH}_3\right] \\
&\leq \Pr\left[\mathrm{SM}_V \mid \mathrm{QH}_3\right] + 2^{-k} \\
&= \Pr\left[\mathrm{SM}_V \mathrm{QH}_3 \wedge \mathrm{QH}_4\right] \\
&\quad + \Pr\left[\mathrm{SM}_V \mid \mathrm{QH}_3 \wedge \neg\mathrm{QH}_4\right] + 2^{-k} \\
&\leq \Pr\left[\mathrm{SM}_V \mid \mathrm{QH}_3 \wedge \mathrm{QH}_4\right] + q_{h_3}\left(2^{-k}\right) + 2^{-k}.
\end{aligned} \tag{30}$$

Thus, we can rewrite the inequality to get the following inequality:

$$\Pr\left[\mathrm{SM}_V \mid \mathrm{QH}_3 \wedge \mathrm{QH}_4\right] \geq \varepsilon - \left(q_{h_3} + 1\right) 2^{-k}. \tag{31}$$

When the event SM_V occurs under the condition that both h_3 and h_4 hash queries have been called, the probability that $\mathcal{B}$ returns $V*^{z^{-1}} = e(P,P)^{abc}$ is only $(q_{h_3} q_{h_4})^{-1}$. Therefore, the probability that $\mathcal{B}$ breaks BDHP is only

$$\varepsilon' \geq \frac{\left(\varepsilon - \left(q_{h_3} + 1\right)/2^k\right)}{\left(q_{h_3} q_{h_4}\right)} \tag{32}$$

within the period of time $t' \approx t + t_\lambda q_{\mathrm{SMG}}$, where t_λ is the time for executing one bilinear pairing operation. □

6. Comparisons on Security and Performance

In this section, we compare the proposed scheme with other existing schemes including the scheme of Li and Chen (LC) [26], the scheme of Wang and Cao (WC) [27], the scheme of Duan et al. (DCZ) [28], the scheme of Elkamchouchi and Abouelseoud (EA) [30], and the scheme of Lin et al. (LWHY) [31]. The comparison on security strength among targeted proxy signcryption schemes is given in Table 1. From Table 1, one can observe that only the proposed scheme provides formal security proof on proxy credential unforgeability. In addition, only the LC scheme, the WC scheme, and the proposed scheme are secure against key-compromised attack and forgery attack. The comparison on performance efficiency among targeted schemes is shown in Table 2. As pairing operation is the most time-consuming operation in comparison with the other computing operations used among targeted schemes [37], only the total number of pairing operations is used to measure performance efficiency for all targeted schemes in Table 2. From Table 2, it is obvious that our scheme is the most efficient proxy signcryption scheme in terms of time consumption for scheme operation. In summary, our scheme provides better security strength and achieves the most efficient operation design among existing schemes.

Table 1: Security strength comparison among proxy signcryption schemes.

	LC	WC	DCZ	EA	LWHY	Ours
Resistance to key-compromised attack	Yes	Yes	No	No	Yes	Yes
Public verifiability	Yes	Yes	Yes	No	Yes	Yes
Security proof on confidentiality	No	No	No	No	Yes	Yes
Unforgeability proof on proxy credential	No	No	No	No	No	Yes
Unforgeability proof on proxy signcryption	No	No	No	No	Yes	Yes
Resistance to forgery attacks	Yes	Yes	Yes	Yes	No	Yes

Table 2: Performance efficiency comparison in terms of the total number of pairing operations required among proxy signcryption schemes.

	LC	WC	DCZ	EA	LWHY	Ours
Pairing operations for PCG	3	2	3	3	0	0
Pairing operations for SMG	2	1	2	2	1	1
Pairing operations for SRV	8	3	4	7	3	2
The total computation cost (number of pairing operations)	13	6	9	12	4	3

7. Conclusion

This paper first shows that the scheme of Lin et al. [31] is vulnerable to two forgery attacks based on chosen warrant attack. Later, a new proxy signcryption scheme is introduced. The proposed scheme only requires one pairing operation to verify the validity of a proxy signature; therefore, the proposed scheme is computationally more efficient than other existing schemes. Moreover, a new security model for proxy signcryption scheme is derived and adopted to prove

our scheme achieves the following security features: IND-CCA2, EF-CWA, and EF-CMA under random oracle model.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the support from the Taiwan Information Security Center (TWISC) and the National Science Council, Taiwan, under Grant no. NSC 102-2218-E-011-013.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, vol. 21, no. 2, pp. 120–126, 1978.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO 2001*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, Springer, Berlin, Germany, 2001.

[4] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Advances in Cryptology—ASIACRYPT 2002*, vol. 2501 of *Lecture Notes in Computer Science*, pp. 548–566, Springer, Berlin, Germany, 2002.

[5] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Advances in Cryptology—ASIACRYPT 2001*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 514–532, Springer, Berlin, Germany, 2001.

[6] F. Zhang and K. Kim, "ID-based blind signature and ring signature from pairings," in *Advances in Cryptology—ASIACRYPT 2002*, vol. 2501 of *Lecture Notes in Computer Science*, pp. 533–547, Springer, Berlin, Germany, 2002.

[7] B. C. Neuman, "Proxy-based authorization and accounting for distributed systems," in *Proceedings of the IEEE 13th International Conference on Distributed Computing Systems*, pp. 283–291, May 1993.

[8] V. Varadharajan, P. Allen, and S. Black, "An analysis of the proxy problem in distributed systems," in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 255–275, May 1991.

[9] J. L. Tsai, N. W. Lo, and T. C. Wu, "Secure delegation-based authentication protocol for wireless roaming service," *IEEE Communications Letters*, vol. 16, no. 7, pp. 1100–11102, 2012.

[10] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 48–57, March 1996.

[11] R. Lu, X. Dong, and Z. Cao, "Designing efficient proxy signature schemes for mobile communication," *Science in China, Series F: Information Sciences*, vol. 51, no. 2, pp. 183–195, 2008.

[12] F. Li, M. Shirase, and T. Takagi, "Cryptanalysis of efficient proxy signature schemes for mobile communication," *Science China. Information Sciences*, vol. 53, no. 10, pp. 2016–2021, 2010.

[13] A. Wang, J. Li, and Z. Wang, "A provably secure proxy signature scheme from bilinear pairings," *Journal of Electronics*, vol. 27, no. 3, pp. 298–304, 2010.

[14] D. Hongzhen and W. Qiaoyan, "An efficient identity-based short signature scheme from bilinear pairings," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS '07)*, pp. 725–729, Haerbin, China, December 2007.

[15] Y.-C. Lin, T.-C. Wu, and J.-L. Tsai, "ID-based aggregate proxy signature scheme realizing warrant-based delegation," *JISE. Journal of Information Science and Engineering*, vol. 29, no. 3, pp. 441–457, 2013.

[16] J. L. Tsai, N. W. Lo, and T. C. Wu, "Numerical analysis of stress on pump blade by one-way coupled fluid-structure simulation," *Information Technology and Control*, vol. 42, no. 4, pp. 315–324, 2014.

[17] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) ≪ cost(signature) + cost(encryption)," in *Advances in Cryptology—CRYPTO 1997*, pp. 165–179, Springer, 1997.

[18] Y. Zheng, "Signcryption and its applications in efficient public key solutions," in *Proceedings of the Information Security Workshop*, pp. 291–312, Springer, 1997.

[19] F. Bao and R. H. Deng, "A signcryption scheme with signature directly verifiable by public key," in *Proceedings of the Workshop on Public Key Cryptography*, pp. 55–59, Springer, 1998.

[20] H. Petersen and M. Michels, "Cryptanalysis and improvement of signcryption schemes," *IEE Proceedings Computers and Digital Techniques*, vol. 145, no. 2, pp. 149–151, 1998.

[21] W.-H. He and T.-C. Wu, "Cryptanalysis and improvement of Petersen-Michels signcryption scheme," *IEE Proceedings: Computers and Digital Techniques*, vol. 146, no. 2, pp. 123–124, 1999.

[22] J.-L. Tsai, "Convertible multi-authenticated encryption scheme with one-way hash function," *Computer Communications*, vol. 32, no. 5, pp. 783–786, 2009.

[23] C. Gamage, J. Leiwo, and Y. Zheng, "An efficient scheme for secure message transmission using proxy-signcryption," in *Proceedings of the 22nd Australasian Computer Science Conference*, pp. 420–431, Springer, 1999.

[24] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Advances in Cryptology—CRYPTO 2002*, vol. 2442 of *Lecture Notes in Computer Science*, pp. 354–368, Springer, Berlin, Germany, 2002.

[25] P. S. L. M. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," in *Selected Areas in Cryptography*, vol. 3006 of *Lecture Notes in Computer Science*, pp. 17–25, Springer, Berlin, Germany, 2004.

[26] X. Li and K. Chen, "Identity based proxy-signcryption scheme from pairings," in *Proceedings of the IEEE International Conference on Services Computing (SCC '04)*, pp. 494–497, September 2004.

[27] Q. Wang and Z. Cao, "Efficient ID-based proxy signature and proxy signcryption from bilinear pairings," in *Computational Intelligence and Security*, pp. 167–172, Springer, 2005.

[28] S. Duan, Z. Cao, and Y. Zhou, "Secure delegation-by-warrant ID-based proxy signcryption scheme," in *Proceedings of Computational Intelligence and Security Conference (CIS '05)*, vol. 3802 of *LNAI*, pp. 445–450, Springer, 2005.

[29] S. Duan and Z. Cao, "Efficient and provably secure multi-receiver identity-based signcryption," in *Information Security and Privacy*, pp. 195–206, Springer, 2006.

[30] H. Elkamchouchi and Y. Abouelseoud, A new proxy identity-based signcryption scheme for partial delegation of signing rights, Cryptology ePrint Archive, Report , 2008, http://eprint.iacr.org/ .

[31] H.-Y. Lin, T.-S. Wu, S.-K. Huang, and Y.-S. Yeh, "Efficient proxy signcryption scheme with provable CCA and CMA security," *Computers & Mathematics with Applications*, vol. 60, no. 7, pp. 1850–1858, 2010.

[32] C.-L. Hsu and H.-Y. Lin, "Pairing-based strong designated verifier proxy signature scheme with low cost," *Security and Communication Networks*, vol. 5, no. 5, pp. 517–522, 2012.

[33] H.-Y. Lin, T.-S. Wu, and S.-K. Huang, "Certificate-based secure three-party signcryption scheme with low costs," *JISE. Journal of Information Science and Engineering*, vol. 28, no. 4, pp. 739–753, 2012.

[34] J. L. Tsai, N. W. Lo, and T. C. Wu, "ID-Based authenticated group key agreement protocol from bilinear pairings for wireless mobile devices," *Adhoc & Sensor Wireless Networks*, vol. 17, no. 3-4, pp. 221–231, 2013.

[35] T. S. Wu and H. Y. Lin, "A novel probabilistic signature based on bilinear square Diffie-Hellman problem and its extension," *Security and Communication Networks*, vol. 6, no. 6, pp. 757–764, 2013.

[36] J. L. Tsai, N. W. Lo, and T. C. Wu, "Secure handover authentication protocol based on bilinear pairings," *Wireless Personal Communications*, vol. 73, no. 3, pp. 1037–1047, 2013.

[37] M. Scott, N. Costigan, and W. Abdulwahab, "Implementing cryptographic pairings on smartcards," in *Cryptographic Hardware and Embedded Systems—CHES 2006*, pp. 134–147, Springer, 2006.