ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2014, T. 43, Nr. 4

Design of Pairing-Based Proxy Signcryption System Model for Online Proxy Auctions

Chien-Lung Hsu

Department of Information Management
Chang Gung University
No. 259 Wen-Hwa 1st Road, Kwei-Shan Tao-Yuan, Taiwan, 333, R.O.C.

Yu-Hao Chuang

Department of Industrial Design
Chang Gung University
No. 259 Wen-Hwa 1st Road, Kwei-Shan Tao-Yuan, Taiwan, 333, R.O.C.
e-mail: yuhao0512@gmail.com

Pei-Ling Tsai

Chunghwa Telecom Laboratories
Ltd. No.99, Dianyan Rd., Yangmei City, Taoyuan County 32601, Taiwan

Atif Alamri

College of Computer and Information Sciences, Research Chair of Pervasive and Mobile Computing
King Saud University, Riyadh, KSA

Sk Md Mizanur Rahman

College of Computer and Information Sciences
King Saud University, Riyadh, KSA

crossref http://dx.doi.org/10.5755/j01.itc.43.4.6348

Abstract. In recent years, the online auction of consumers' goods is an increasingly popular selling channel. To copy with this tendency, the bidders' convenience and security have to be considered. Some online auctions (such as eBay or Amazon) introduced a proxy bidding strategy that the bidder can decide a maximum bid and delegate his authority to a legal agent to automatically outbid other competitors for the last winner. In this paper, we identify some practical issues in the current bidding strategies and propose a system model of a pairing-based proxy signcryption which includes two cryptographic schemes; further, we apply the proposed schemes to the online proxy auction system for comparing two authorized online proxy auction policies (agents anonymity setting and agents accountability setting) with different applications of short message and long message. Also, we discuss some critical security issues under the online proxy auctions.

Keywords: Online auction, proxy, agent, authorized, anonymity, accountability, short message, and long message.

1. Introduction

In the areas of computer communications and electronic transactions, how to transmit data in a confidential and authenticated way has become increasingly important. Traditional cryptosystems usually adopt a two-step approach to reach confidentiality, integrity, authentication, and non-repudiation. The two-step approach is: the sender first generates a signature for a message, and then encrypts both the signature and the message to get a ciphertext, which is subsequently sent to the receiver; the receiver must decrypt the ciphertext before obtaining the signature and the message, and then check the validity of the signature. This approach results in higher computation cost and communication overhead. In 1997, at CRYPTO'97 conference, Zheng [28] introduced a primitive that he called signcryption. The idea of the signcryption scheme is to combine the functionality of encryption and signature schemes, which has been proved to be more efficient than the traditional scheme.

In addition, the ability of delegation (or proxy) is an issue as it is necessary for the organization to remain normal operation [9,27]. For example, someone may need an agent when he is away during official business hours or on vocation. The proxy signcryption primitive was first proposed by Gamage, Leiwo, and Zheng [7] in 1999. Their scheme can efficiently achieve the combined functionalities of proxy signature scheme and encryption scheme. Since then, several research works on proxy signcryption have been published [11,14,15,24].

The concept of self-certified public key system was first introduced by Girault [16] at EUROCRYPT'91 conference to resolve the public key verification problems. That is, the public key for each user is generated by the certification authority (CA), while the corresponding private key is only known to the user. The user can use his private key to verify the self-certified public key issued by the CA, and thus no extra certificate is required. In addition, the verification of the public key can be accomplished with the subsequent cryptographic application in a single step to save computation effort. Because the public key verification with the self-certified public key system is more efficient in saving the communicational cost and the computational effort as compared to the identity-based and the certificate-based public key systems, more scholars invest their time to design and analyze of the self-certified public key system successively [3,8,9,22,23,29,30].

Based on the cryptographic system from pairings, a bilinear map $\hat{e}$ is defined with the properties of the bilinear, non-degenerate, and computable. Because of superior security of the pairing-based system over the elliptic curve cryptosystem (ECC)–based system, we would like to design a self-certified public key system from pairings. Making a comprehensive survey of proxy signcryption from pairings, we find that only the identity-based [1,17,18,21,25] and certificate-based [18] proxy signcryption schemes have been discussed; the self-certified based proxy signcryption schemes from pairings have not been proposed yet. According to the advantages of the self-certified public key system described above, we design two self-certified proxy signcryption schemes based on the difficulty of pairings.

In addition, two realistic cases should be concerned in a proxy signcrypted message: short message and long message. To deal with the practical situations, we design message recovery scheme to fulfill the demand of short message, such as an identity field to be signcrypted, whereas a scheme with appendix is proposed to respond the requirement of long message such as a document to be signcrypted. Moreover, the proxy signer may expect a design of signcrypted message with anonymity or non-anonymity. To meet with the requirements, we provide anonymous and non-anonymous functionalities to be chosen by the proxy signer in the proposed proxy signcryption schemes. In the next section, we introduce the system model for the proposed proxy signcryption schemes.

2. System models

Elaborating on the merits of elliptic curve cryptography and self-certified public key systems, we first propose a system model of a self-certified public key system from pairings. Realization of the self-certified public key system is proposed in Section 3. We further consider the following practical issues to propose a system model of a proxy signcryption system based on self-certified public key systems from pairings:

(i) *Length of the signed messages*: Digital signature schemes are generally divided into two categories, signature schemes with message recovery and those with appendix. The former can gain better performance in terms of communicational costs and computational complexities but the signed messages are limited to a predefined fixed length. To enhance the security for such schemes, a secure redundancy mechanism is required. The length of signed messages is unlimited in the signature scheme with appendix. Hence, such schemes are suitable for some applications with long messages. Properties of both schemes are all considered in the proposed system. That is, we can consider the application scenario to determine adopting signature with message or that with appendix.

(ii) *User anonymity or user accountability*: The proposed system considers user anonymity and user accountability properties. Users' identities are protected from being disclosed in the applications with user anonymity, while those are revealed in the applications with user accountability.

The model of the self-certified public key system mainly describes how the private/ public key pair of a user U_i is generated with the CA (see Fig. 1). We define the following algorithms in supporting the subsequent proposed proxy signcryption models.

- **Public Parameter Generation** (***Setup***): Given a security parameter 1^{κ} , output the public parameters which are shared and used by all actors in the scheme.
- **CA Key Generation** (***CAKeyGen***): Given the public parameters, output a pair of private and public keys (x_{CA}, Y_{CA}) for the CA.
- **User Key Generation Protocol** (***UKeyGenP***): Given the public parameters, output a pair of private and public keys (X_i, Y_i) for each actor U_i. All steps are given as follows:
  - $U_i \rightarrow \text{CA} : (V_i, id_i)$: U_i computes a public value V_i and delivers it with his identity id_i to the CA.
  - $\text{CA} \rightarrow U_i : (Y_i, W_i)$: the CA computes U_i 's public key Y_i and a witness W_i for Y_i , and transmits (Y_i, W_i) to U_i .
  - U_i: U_i can compute his own private key X_i and verify the public key Y_i with (Y_i, W_i) .

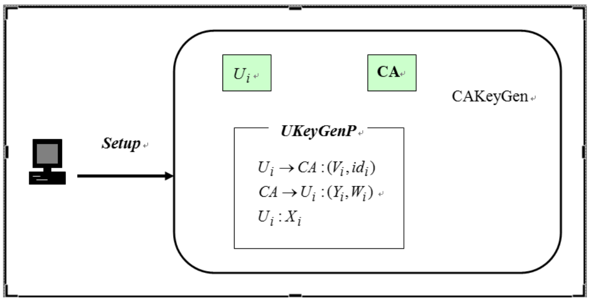
U_i
CA
CAKeyGen
Setup
UKeyGenP
$U_i \rightarrow CA:(V_i, id_i)$
$CA \rightarrow U_i:(Y_i, W_i)$
$U_i : X_i$

Figure 1. The model of the self-certified public key system

What is more, the model of the proxy signcryption scheme consists of an original signer U_A , a proxy signer U_P, and a specified verifier U_B. Through the delegation procedure of this model, it allows U_P to generate a proxy signcrypted message, which can only be unsigncrypted by U_B , on behalf of U_A (see Fig. 2). ***DeleGen*** is first performed by U_A to authorize his signing power to U_P. After performing ***DeleVerify*** and confirming the delegation is valid, U_P can execute ***PSCMR*** or ***PSCAP*** to signcrypt on behalf of U_A to U_B depending on suitable demand. Then U_B can unsigncrypt through the corresponding ***USCMR*** or ***USCAP*** . The algorithms are defined below.

- **Delegation Generation** (***DeleGen***): Given the public parameters, the original signer's private key X_A, the CA's public key Y_{CA}, and a warrant m_w, which describes the relative rights and information of the original signer and proxy signer, a valid period of time, and so on, output the authorization (σ, U, m_w) , where σ is the proxy share and U is a computed value.
- **Delegation Verification** (***DeleVerify***): Given the public parameters, the authorization (σ, U, m_w) , the original signer's public key Y_A and identity id_A, and the CA's public key Y_{CA}, output *true* if (σ, U, m_w) is correct or *false* otherwise.
- **Proxy Signcrypt with Message Recovery** (***PSCMR***): Given the public parameters, the authorization (σ, U, m_w) , the proxy signer's private key X_P , the specified verifier's public key Y_B and identity id_B, the CA's public key Y_{CA}, a message m with additional specified redundancy, and a system variable $flag$, which means anonymity when $flag = A$, and non- anonymity when $flag = NA$, output the proxy signcrypted message PSM_MR.
- **Unsigncrypt with Message Recovery** (***USCMR***): Given the public parameters, the proxy signcrypted message PSM_MR, the specified verifier's private key X_B, the CA's public key Y_{CA}, the original signer's public key Y_A and identity id_A, the proxy signer's public key Y_P and identity id_P , the delegation information (U, m_w) , and a system variable $flag$, output the message m and check the redundancy.
- **Proxy Signcrypt with Appendix** (***PSCAP***): Given the public parameters, the authorization (σ, U, m_w) , the proxy signer's private key X_P , the specified verifier's public key Y_B and identity id_B, the CA's public key Y_{CA}, a message m , and a system variable $flag$, output the proxy signcrypted message PSM_AP .
- **Unsigncrypt with Appendix** (***USCAP***): Given the public parameters, the proxy signcrypted message PSM_AP , the specified verifier's private key X_B, the CA's public key Y_{CA}, the original signer's public key Y_A and identity id_A , the proxy signer's public key Y_P and identity id_P , the delegation information (U, m_w) , and a system variable $flag$, output *true* if PSM_AP is a valid proxy signcryption for message m or *false* otherwise.

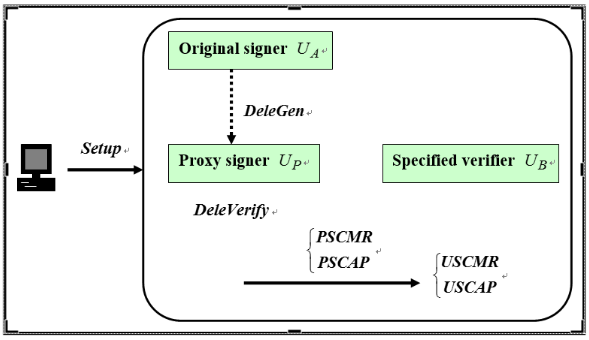

Figure 2. The model of the proxy signcryption scheme

3. The proposed self-certified public key system from pairings

Our concrete construction of the self- certified public key system from pairings is defined as follows.

- ***Setup***: Select a security parameter 1^K that defines the length of a large prime q. Generate a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$, where G_1 is a cyclic additive group generated by P, whose order is q, and G_2 is a cyclic multiplicative group of the same order q. Four cryptographic hash functions are defined: $H:\{0,1\}^* \times Z_q^* \to Z_q^*$, $H_1:\{0,1\}^* \to G_1$, $H_2: G_1 \times \{0,1\}^* \to Z_q^*$, and $H_3: G_2 \to Z_q^*$. Hereafter, the system publishes the public parameters $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$.
- ***CAKeyGen***: Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, choose $x_{CA} \in_R Z_q^*$ as the CA's private key. The corresponding public key is computed as $Y_{CA} = x_{CA}P$.
- ***UKeyGenP***: Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, perform the following operations:
  - $U_i \to \mathrm{CA}: (V_i, id_i)$: U_i chooses $v_i \in_R Z_q^*$ and computes the V_i as

    $$V_i = H(v_i \| id_i)P \qquad (1)$$

    where $\|$ denotes the string concatenation operator, and sends (id_i, V_i) to the CA.
  - $\mathrm{CA} \to U_i: (Y_i, W_i)$: the CA computes U_i's public key $Y_i = V_i + n_iP$, where $n_i \in_R Z_q^*$, and generates the witness of the public key as $W_i = n_iY_{CA} + (H_2(Y_i \| id_i) - 1)x_{CA}Y_i$. Then the CA submits (Y_i, W_i) to U_i.
  - U_i can derive his private key X_i as

    $$X_i = W_i + H(v_i \| id_i)Y_{CA} \qquad (2)$$

    and then verify the authenticity of Y_i by testing if $\hat{e}(X_i, P) = \hat{e}(Y_i, Y_{CA})^{H_2(Y_i \| id_i)}$.

4. The proposed self-certified proxy signcryption schemes

In this section, the concrete construction of our proposed self-certified proxy signcryption schemes is discussed. There are three algorithms, ***Setup***, ***CAKeyGen***, and ***UKeyGenP***, which are defined in Section 3. The approaches are illustrated in the following:

- ***DeleGen***: Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, the original signer U_A performs the following steps:
  - Pick $u \in_R Z_q^*$ and compute $U = uP$.
  - Generate a warrant m_w, which describes the relative rights and information of the original signer and proxy signer, a valid period of time, etc. Note that it doesn't contain any information about the proxy signer in the anonymous applications.
  - Compute the proxy share as follows: $\sigma = H_2(m_w \| U)X_A + uY_{CA}$.
  - Transmit (σ, U, m_w) to the proxy signer U_P through a secure channel.
- ***DeleVerify***: Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, the proxy signer U_P performs the following steps:
  - Calculate $\hat{e}(\sigma, P) =$

    $$\hat{e}(Y_A, Y_{CA})^{H_2(m_w \| U) \cdot H_2(Y_A \| id_A)} \cdot \hat{e}(U, Y_{CA})$$

    to check the validity of the delegation.
  - Perform ***PSCMR*** or ***PSCAP*** depending on different requirement if it's *true*.
- ***PSCMR***: Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, the proxy signer U_P performs the following steps:
  - Set the proxy signing key as $\sigma' = \sigma$, if $flag = A$; otherwise, set it as $\sigma' = H_2(m_w \| U)X_P + \sigma$, if $flag = NA$.
  - Choose a message m, which contains an additional specified redundancy.
  - Pick $d, w \in_R Z_q^*$.
  - Compute

    $$r = m \cdot H_3(\hat{e}(dY_{CA}, P))^{-1} (\mathrm{mod}\ q),$$

    $$S = dY_{CA} - r\sigma',$$

    $C_1 = wP$, **and**

    $$c_2 = r \oplus H_3(\hat{e}(wY_B, Y_{CA})^{H_2(Y_B \| id_B)}) \ (\mathrm{mod}\ q). \qquad (3)$$
  - Send the proxy signcrypted message $PSM\_MR = (C_1, c_2, S, U, m_w)$ to a specified verifier U_B via a public channel.

- ***USCMR*** : After receiving $PSM\_MR = (C_1, c_2, S, U, m_w)$, the specified verifier U_B executes ***USCMR*** to check the validity of PSM_MR . Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, the specified verifier U_B performs the following steps:
  - Calculate
    $$r = c_2 \oplus H_3(\hat{e}(X_B, C_1))(\text{mod } q) \qquad (4)$$
  - If $flag = A$, recover the message by the equation
    $$m = r \cdot H_3(\hat{e}(S,P) \cdot \hat{e}(U, Y_{CA})^r \cdot \hat{e}(Y_A, Y_{CA})^{r \cdot H_2(m_w \| U) \cdot H_2(Y_A \| id_A)})(\text{mod } q).$$
    If $flag = NA$, recover the message by the equation
    $$m = r \cdot H_3(\hat{e}(S,P) \cdot \hat{e}(U, Y_{CA})^r) \cdot \hat{e}(Y_A, Y_{CA})^{r \cdot H_2(m_w \| U) \cdot H_2(Y_A \| id_A)} \cdot \hat{e}(Y_P, Y_{CA})^{r \cdot H_2(m_w \| U) \cdot H_2(Y_P \| id_P)}) \\ (\text{mod } q).$$
  - Check the redundancy by analyzing whether $PSM\_MR = (C_1, c_2, S, U, m_w)$ is a valid proxy signcrypted message for *m* or not.
- ***PSCAP*** : Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, the proxy signer U_P performs the following steps:
  - Set the proxy signing key as $\sigma' = \sigma$, if $flag = A$; otherwise, set it as $\sigma' = H_2(m_w \| U) X_P + \sigma$, if $flag = NA$.
  - Choose a message m .
  - Pick $d \in_R Z_q^*$.
  - Compute $R = dP$, $S = dY_{CA} + H_2(R \| m)\sigma'$, $k = \hat{e}(dY_B, Y_{CA})^{H_2(Y_B \| id_B)}$, and $c = E_k(S \| m)$, where $E_k(\cdot)$ is an ideal symmetric key encryption algorithm.
  - Send the proxy signcrypted message $PSM\_AP = (R, c, U, m_w)$ to a specified verifier U_B via a public channel.
- ***USCAP*** : Upon receiving $PSM\_AP = (R, c, U, m_w)$ the specified verifier U_B executes ***USCAP*** . Given $(G_1, G_2, P, q, \hat{e}, H, H_1, H_2, H_3)$, the specified verifier U_B performs the following steps:
  - Calculate $k = \hat{e}(X_B, R)$ (5)
    and get the signcrypted message by the equation $S \| m = D_k(c)$, where $D_k(\cdot)$ is an ideal symmetric decryption algorithm corresponding to $E_k(\cdot)$.
  - If $flag = A$, verify the proxy signcrypted message by the following equation:
    $$\hat{e}(S,P) = \hat{e}(U, Y_{CA})^{H_2(R \| m)} \cdot \hat{e}(R, Y_{CA}) \cdot \hat{e}(Y_A, Y_{CA})^{H_2(m_w \| U) \cdot H_2(Y_A \| id_A) \cdot H_2(R \| m)}) \\ (\text{mod } q).$$

    If $flag = NA$, verify the proxy signcrypted message by the following equation:
    $$\hat{e}(S,P) = \hat{e}(U, Y_{CA})^{H_2(R \| m)} \cdot \hat{e}(R, Y_{CA}) \cdot \hat{e}(Y_A, Y_{CA})^{H_2(m_w \| U) \cdot H_2(Y_A \| id_A) \cdot H_2(R \| m)}) \cdot \hat{e}(Y_P, Y_{CA})^{H_2(m_w \| U) \cdot H_2(Y_P \| id_P) \cdot H_2(R \| m)}) \\ (\text{mod } q).$$

    If the output of the above equation is true, the proxy signcrypted message is valid; otherwise, it is invalid.

5. Analysis and discussions

We prove the correctness of our schemes in Section 5.1 and give security analysis in Section 5.2. In Section 5.3, we discuss the main contribution of the proposed schemes.

5.1. Correctness

The correctness of our schemes is verified as follows:

- For the ***UKeyGenP*** algorithm,
  $$\begin{aligned} \hat{e}(X_i, P) &= \hat{e}(W_i + H(v_i \| id_i) Y_{CA}, P) \\ &= \hat{e}(n_i Y_{CA} + (H_2(Y_i \| id_i) - 1) x_{CA} Y_i + H(v_i \| id_i) Y_{CA}, P) \\ &= \hat{e}(n_i P + H(v_i \| id_i) P + (H_2(Y_i \| id_i) - 1) Y_i, Y_{CA}) \\ &= \hat{e}(Y_i + (H_2(Y_i \| id_i) - 1) Y_i, Y_{CA}) \\ &= \hat{e}(Y_i, Y_{CA})^{H_2(Y_i \| id_i)}. \end{aligned}$$
- For the ***DeleVerify*** algorithm,
  $$\begin{aligned} \hat{e}(\sigma, P) &= \hat{e}(X_A, P)^{H_2(m_w \| U)} \cdot \hat{e}(u Y_{CA}, P) \\ &= \hat{e}(Y_A, Y_{CA})^{H_2(m_w \| U) \cdot H_2(Y_A \| id_A)} \cdot \hat{e}(U, Y_{CA}). \end{aligned}$$
- For the ***USCMR*** algorithm,
  - $H_3(\hat{e}(X_B, C_1)) = H_3(\hat{e}(X_B, P)^w) = H_3(\hat{e}(wY_B, Y_{CA})^{H_2(Y_B \| id_B)})$.
  - If $flag = A$,
    $$\begin{aligned} m &= r \cdot H_3(\hat{e}(dY_{CA}, P)) \\ &= r \cdot H_3(\hat{e}(S,P) \cdot \hat{e}(\sigma, P)^r) \\ &= r \cdot H_3(\hat{e}(S,P) \cdot \hat{e}(U, Y_{CA})^r \cdot \hat{e}(Y_A, Y_{CA})^{r \cdot H_2(m_w \| U) \cdot H_2(Y_A \| id_A)})(\text{mod } q). \end{aligned}$$
  - If $flag = NA$,

$$\begin{aligned}m &= r\cdot H_3(\hat{e}(dY_{CA},P))\\ &= r\cdot H_3(\hat{e}(S,P)\cdot\hat{e}(H_2(m_w \parallel U)X_P+\sigma,P)^r)\\ &= r\cdot H_3(\hat{e}(S,P)\cdot\hat{e}(U,Y_{CA})^r\\ &\quad\cdot\hat{e}(Y_A,Y_{CA})^{r\cdot H_2(m_w\|U)\cdot H_2(Y_A\|id_A)})\cdot\hat{e}(Y_P,\\ &\quad Y_{CA})^{r\cdot H_2(m_w\|U)\cdot H_2(Y_P\|id_P)})(\text{mod } q).\end{aligned}$$

- For the ***USCAP*** algorithm,

  - $k=\hat{e}(X_B,R)=\hat{e}(X_B,P)^d$

    $=\hat{e}(dY_B,Y_{CA})^{H_2(Y_B,id_B)}.$

  - If $flag=A$,

$$\begin{aligned}\hat{e}(S,P) &= \hat{e}(dY_{CA},P)\cdot\hat{e}(\sigma,P)^{H_2(R\|m)}\\ &= \hat{e}(U,Y_{CA})^{H_2(R\|m)}\cdot\hat{e}(R,Y_{CA})\\ &\quad\cdot\hat{e}(Y_A,Y_{CA})^{H_2(m_w\|U)\cdot H_2(Y_A\|id_A)\cdot H_2(R\|m)}).\end{aligned}$$

If $flag=NA$,

$$\begin{aligned}\hat{e}(S,P) &= \hat{e}(dY_{CA},P)\cdot\hat{e}(H_2(m_w \parallel U)X_P\\ &\quad+\sigma,P)^{H_2(R\|m)}\\ &= \hat{e}(U,Y_{CA})^{H_2(R\|m)}\cdot\hat{e}(R,Y_{CA})\\ &\quad\cdot\hat{e}(Y_A,Y_{CA})^{H_2(m_w\|U)\cdot H_2(Y_A\|id_A)\cdot H_2(R\|m)})\\ &\quad\cdot\hat{e}(Y_P,Y_{CA})^{H_2(m_w\|U)\cdot H_2(Y_P\|id_P)\cdot H_2(R\|m)}).\end{aligned}$$

5.2 Security analysis

The security analysis of the proposed self-certified proxy signcryption schemes is proved as follows:

- *Valid public key generation:*

According to the operation $CA\rightarrow U_i$: (Y_i,W_i), for any adversary, he cannot forge a legal public key Y_i and a witness W_i without knowing the CA's private key x_{CA}. Because the public key Y_i of U_i is cooperatively generated by the CA and U_i.

- *Private key confidentiality:*

According to Eqs. (1) and (2), it is s infeasible to compute $H(v_i \parallel id_i)$ from V_i. The security is based on the ECDLP (Elliptic Curve Discrete Logarithm Problem). Hence, no one except U_i can compute the private key X_i.

- *Ciphertext confidentiality:*

The proxy signcrypted message from U_P to U_B is confidential against an adversary A. If A wants to recover the signcrypted message from PSM_MR, he must compute r first. Hence, in Eq. (3), A should know the random number w, and in Eq. (4) A should know the private key X_B. If A wants to acquire the secret information from PSM_AP, he must know the private key X_B to compute the session key k in Eq. (5).

- *Unforgeability:*

If an adversary A attempts to forge a valid proxy signcryption for the specified verifier by arbitrarily choosing a message m', he should choose random numbers d' and w' to compute r', S', C_1', and c_2' in ***PSCMR***. Simultaneously, these parameters must satisfy to perform the unsigncrypted procedures in ***USCMR*.** Consider the other scenario in ***PSCAP***, the adversary A must randomly choose d' and calculate R', S', k', and c' to satisfy the unsigncrypted procedures in ***USCAP***. However, we find out that the adversary A cannot forge a valid proxy signcryption without knowing the proxy share σ and the private key X_P (in the non-anonymous way) to forge.

- *Verifiability:*

For the ***USCMR*** and ***USCAP*** algorithms, the verifier can be convinced of the original signer's agreement on the delegation authority for generating a signcryption because a valid proxy signcryption verification must contain the original signer's public key Y_A and identifier id_A.

- *Secret key's dependence:*

For the ***DeleGen***, ***PSCMR*** and ***PSCAP*** algorithms, the proxy signing key σ' is computed from the original signer's private key X_A.

- *Non-repudiation:*

Due to the trustiness assumption in the anonymous way, here we discuss the non-anonymous way. The proxy signer must generate a valid proxy signcryption with his private key X_p and the proxy share σ. Thus once the proxy signer generated a proxy signcryption, he cannot deny what he had done.

- *Forward security:*

For an adversary A in the ***USCMR***, he require the knowledge of $H_3(\hat{e}(wY_B,\ Y_{CA})^{H_2(Y_B\|id_B)})$. However, it is difficult for *A* to get *w* from *r* since it is difficult to invert the bilinear mapping. Further, it is also infeasible to derive w from C_1 for the ECDLP. In addition, for an adversary A in the ***USCAP*** he requires the knowledge of $\hat{e}(dY_B,Y_{CA})^{H_2(Y_B\|id_B)}$. Similarly, it is difficult to invert the bilinear mapping for computing *d*, and to derive *d* from *R* for the ECDLP. What is more, even if Y_{CA}, $H_2(R\parallel m)$, and σ' are given, it is infeasible to compute *d* from *S*. Thus, our schemes satisfy forward security.

5.3. Main contribution

Elaborating on the merits of signcryption schemes, proxy signature schemes, which enable self-certified public key systems and pairing-based cryptosystems, we adopt the delegation with warrants to propose two pairing-based proxy signcryption schemes with self-certified public key cryptosystems in this paper. The key features of the proposed signcryption schemes are listed below.

(i) *Efficiency*: A proxy signer can generate a signcryption to achieve the functionality of encryption and signature schemes simultaneously.

(ii) *Private key confidentiality*: All private keys cannot be compromised by the adversaries during signing or verifying the proxy message signcrypted procedures.

(iii) *Ciphertext confidentiality*: No one can acquire the information from a proxy signcrypted message except the designated verifier.

(iv) *Unforgeability*: No one can generate a valid proxy signcryption except the designated proxy signer.

(v) *Verifiability*: All verifiers can be convinced that the proxy signer has the original signer's agreement on the delegated authority for generating a signcryption.

(vi) *Secret key's dependence*: The proxy signing key is computed from the original signer's private key.

In addition, the proposed proxy signcryption schemes can grant the proxy signer to generate a signcryption with short message or long message and to choose the functionality of anonymity or non-anonymity. We also demonstrate that our proposed schemes are more secure and suitable for handling practical cases (e.g. online auction as depicted in Section 6) than previous studies [1,17,18,21,25] in the following aspects.

(i) *Higher security*: The proposed schemes are pairing-based cryptosystems that can provide superior security protections than ECC-based systems. Even without using certificates, the proposed schemes can also avoid against the active and impersonation attacks.

(ii) *Low computation and communication costs*: The proposed schemes are beneficial in computing and communicating costs since they need no extra certificates to validate public keys.

(iii) *Accountability*: The proxy signer can choose the functionality of non- anonymity to achieve the accountability since the valid proxy signcryption can only be generated by the proxy signer.

(iv) *Anonymity*: The proxy signer can choose the functionality of anonymity to protect his privacy and security, thus the proxy signer's identity is anonymous to the verifier.

(v) *Short message or long message*: The proposed schemes provide two approaches of message recovery [12] and appendix [13] to satisfy the different applications of short message and long message, respectively.

6. Implementation

In recent years, with the development of E-commerce over Internet, many enterprises sell their products to different regions and countries [4,10,27]. Internet has increased the sale of various products or services via auction mechanisms that each bidder can bid on the products or services by this way. Security and privacy problems related to online auctions have been recognized as the key of information system research and have been published in some journals since 2000 [5,6,19,20]. In this paper, we discuss a particular bidding package procedure with desirable properties that is so-called proxy auction, in which each bidder can delegate authority to their respective agents. The agent can submit package bids on behalf of the original bidder to bid the products or services over Internet. We consider that the scenario is suitable to use our proposed proxy signcryption schemes to deal. Detail procedures are described in the following (see Fig. 3).

In Fig. 3, four entities are involved in proxy auction systems: a certification authority (CA), an original bidder, an agent, and a seller. Responsibility of the CA is to setup all system parameters and perform key management tasks. An original bidder can delegate authority to an agent for his auction. The agent can submit the bidding to the seller on behalf of the original bidder. The seller will announce the auction results on the auction board. There are six phases of the proposed proxy auction system: the Setup, the Key generation, the Delegation, the Bidding, the End of bidding, and the Controversy phases. In Setup phase, all system parameters will be determined by performing ***Setup*** algorithm. In Key generation phase, keys of all entities involved in the system will be determined by performing ***CAkeyGen*** and ***UkeyGenP*** algorithms. Key management tasks will be handled by CA. In Delegation phase, the original bidder can perform ***DeleGen*** algorithm to delegate authority to a designated agent. In Bidding phase, the agent can perform ***PSCMR*** or ***PACAP*** algorithms to generate and submit a bid to the seller. In End of bidding phase, the seller can perform ***USCMR*** or ***USCAP*** algorithms to check the validity of the received bids and to announce the auction results on the board. In case of a latter dispute, the seller can prove the auction results to any third party in Controversy phase. Algorithms used in the proposed proxy auction system are described below.

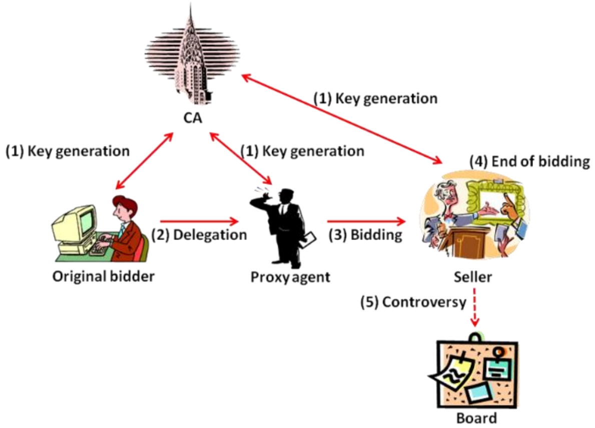

Figure 3. The procedures of proxy auction

- **Setup**: All public parameters are generated by ***Setup*** algorithm and shared with each participant.
- **Key generation:**
  - **CA key generation**: The CA can perform ***CAkeyGen*** algorithm to generate a pair of private and public keys for each participant.
  - **Actor key generation**: Bidders, agents, and sellers can perform ***UkeyGenP*** algorithm to generate their key pairs (private and public keys) with the CA.

  Note that all public keys need no certificates to check the validity. Validities of the public keys are implicitly verified in signature verification.
- **Delegation:** The bidder can perform ***DeleGen*** algorithm to delegate authority to his respective agent. Then the agent can perform ***DeleVerify*** algorithm to check the validity of a delegation.
- **Bidding**: If the delegation is valid, the agent can perform ***PSCMR*** or ***PACAP*** algorithms to generate bid packages on behalf of the original bidder to place bids for the products or services over Internet. All bid packages are encrypted, and thus any information is not revealed.
- **End of bidding**: The seller can perform ***USCMR*** or ***USCAP*** algorithms to decrypt each bid package and verify its validity. Then he decides the winning bidder.
- **Controversy**: If there is a controversy over the bidding, the seller can publish all bid packages after the bidding action. Since each bid package includes the signature of the original bidder, the original bidder cannot deny his creation of a bid package even if the bid package is generated by the agent.

7. Conclusions

We have proposed a model of a proxy signcryption system and two self-certified proxy signcryption schemes from pairings, which possess the advantages of both self-certified public key systems and the proxy signcryption schemes. One is with message recovery and the other is with appendix. From practical considerations, the former is suitable for the applications with short messages and the latter is for those with long messages. Furthermore, we consider user anonymity or user accountability to allow the proxy signer signcrypted messages anonymously or non- anonymously on behalf of the original signer for a designated verifier. The proposed schemes have higher security and gain better performance in communicational costs and computational complexities. Finally, we apply the proposed schemes to a proxy auction system.

Acknowledgement

We would like to thank anonymous referees for their valuable suggestions. We thank Healthy Aging Research Center (HARC) of Chang Gung University for excellent technical assistance. This work was supported in part by the Chang Gung University Grant CMRPD3D0041 and in part by National Science Council under the grant NARPD3D0071. The authors would also like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this research through the Research Group Project no RGP- VPP-258.

References

[1] **B. Libert, J. Quisquater**. New Identity Based Signcryption Schemes from Pairings. In: *Proceedings of IEEE Information Theory Workshop*, 2003, pp. 155-158.

[2] **C. C. Lee, T. C. Lin, S. F. Tzeng, M. S. Hwang.** Generalization of Proxy Signature Based on Factorization. *International Journal of Innovative Computing, Information and Control*, 2011, Vol. 7, No. 3, 1039-1054.

[3] **C. L. Hsu, T. S. Wu.** Efficient proxy signature schemes using self-certified public keys. *Applied Mathematics and Computation*, 2004, Vol. 152, No. 3, 807-820.

[4] **C. L. Yang, R. H. Huang**. Key success factors for online auctions: Analysis of auctions of fashion clothing. *Expert Systems with Applications*, 2011, Vol. 38, No. 6, 7774-7783.

[5] **C. Li, S. Chawla, U. Rajan, K. Sycara.** Mechanism design for coalition formation and cost sharing in group-buying markets. *Electronic Commerce Research and Applications*, 2004, Vol. 3, No. 4, 341-354.

[6] **E. J. Pinker, A. Seidmann, Y. Vakrat**. Managing online auctions: Current business and research issues. *Management Science*, 2003, Vol. 48, No. 10, 1457-1485.

[7] **G. Gamage, J. Leiwo, Y. Zheng**. An efficient scheme for secure message transmission using proxy-signcryption. In: *Proceedings of the 22nd Australasian Computer Science Conference*, 1999, pp. 420-431.

[8] **H. K. Yang, J. H. Choi, Y. H. Ann**. Self-certified identity information using the minimum knowledge. *IEEE TENCON: Digital Signature Processing Application*, 1996, pp. 641-647.

[9] **H. Petersen, P. Hoster**. Self-certified keys-concepts and applications. In: *Proceedings of the 3rd Conferen-*

ce on Communications and Multimedia Security, 1997, pp. 102-116.

[10] **H. Xiong, Z. Chen, F. Li.** Bidder-anonymous English auction protocol based on revocable ring signature. *Expert Systems with Applications*, 2012, Vol. 39, No. 8, 7062-7066.

[11] **H. Y. Lin, T. S. Wu, S. K. Huang, Y. S. Yeh.** Efficient proxy signcryption scheme with provable CCA and CMA security. *Computers and Mathematics with Applications*, 2010, Vol. 60, No. 7, 1850-1858.

[12] **ISO/IEC 9796,** Information technology – security techniques – digital signature scheme given message recovery, 1991.

[13] **ISO/IEC 14888-3,** Information technology–security techniques–digital signature with appendix–part 3: certificate-based mechanisms, International Organization for Standardization, 1998.

[14] **J. G. Li, J. Z. Li, Z. F. Cao, Y. C. Zhang**. Convertible proxy signcryption scheme. *Journal of Harbin Institute of Technology*, 2004, Vol. 11, No. 2, 209-213.

[15] **L. Chen, J. M. Lee**. Improved identity-based signcryption. Public Key Cryptography *(PKC'2005), Lecture Notes in Computer Science*, 2005, Vol. 3386, 362-379.

[16] **M. Girault**. Self-certified public key. *Advances in Cryptology (EUROCRY PT'91)*, 1991, 491-497.

[17] **M. Wang, H. Li, Z. Liu.** Efficient identity based proxy-signcryption schemes with forward security and public verifiability. In: *Proceedings of the 3rd International Conference on Networking and Mobile Computing (ICCNMC'2005)*, 2005, pp. 982-991.

[18] **Q. Wang, Z. Cao.** Two proxy signcryption schemes from bilinear parings. In: *Proceedings of the 4th International Conference on Cryptology and Network Security (CANS 2005)*, 2005, pp. 161-171.

[19] **R. Bapna, P. Goes, A. Gupta**. Analysis and design of business-to-consumer online auctions. *Management Science*, 2003, Vol. 49, No. 1, 85-101.

[20] **R. J. Kauffman, E. Walden.** Economics and electronic commerce: Survey and directions for research. *International Journal of Electronic Commerce*, 2001, Vol. 5, No. 4, 5-116.

[21] **S. S. M. Chow, S. M. Yiu, L. K. Hui, K. P. Chow.** Efficient forward and provable secure id-based signcryption scheme with public verifiability and public ciphertext authenticity. In: *International Conference on Information Security and Cryptology (ICISC'2003), Lecture Notes in Computer Science*, 2003, Vol. 2971, pp. 352-369.

[22] **S. Saeednia**. Identity-based and self-certified key-exchange protocols. In: *Proceedings of the 2nd Australasian Conference on Information Security and privacy (ACISP'97)*, 1997, pp. 303-313.

[23] **W. J. Tsaur**. Several security schemes constructed using ECC-based self-certified public key cryptosystems. *Applied Mathematics and Computation*, 2005, Vol. 168, No. 1, 447-464.

[24] **X. Boyen**. Multipurpose identity-based signcryption: a Swiss army knife for identity-based cryptography. *Advances in Cryptology (CRYPTO'2003), Lecture Notes in Computer Science*, 2003, Vol. 2729, pp. 382-398.

[25] **X. Li, K. Chen.** Identity based proxy-signcryption scheme from pairings. In: *Proceedings of 2004 IEEE International Conference on Services Computing*, 2004, pp. 494-497.

[26] **X. Wang, K. S. Chin, H. Yin.** Design of optimal double auction mechanism with multi-objectives. *Expert Systems with Applications*, 2011, Vol. 38, No. 11, 13749-13756.

[27] **Y. C. Chen, C. L. Liu, G. Horng, K. C. Chen.** A Provably Secure Certificateless Proxy Signature Scheme. *International Journal of Innovative Computing, Information and Control*, 2011, Vol. 7, No. 9, 5557-5569.

[28] **Y. Zheng.** Digital signcryption or how to achieve cost (signature & encryption) << cost (signature)+cost (encryption). *Advanced in Cryptology (CRYPTO'97), Lecture Notes in Computer Science*, 1997, Vol. 1294, pp. 165-179.

[29] **Z. Shao.** Self-certified signature scheme from pairings. *The Journal of Systems & Software*, 2007, Vol. 80, No. 3, 388-395.

[30] **Z. Shao**. Security of self-certified signatures. *Information Processing Letters*, 2009, Vol. 109, No. 20, 1147-1150.

Received February 2014.