An Efficient Secure Anonymous Proxy Signature Scheme

*Jue-Sam Chou [1], Shih-Che Hung [2], Yalin Chen [3]

[1] Department of Information Management, Nanhua University Chiayi 622, Taiwan

*: corresponding author

jschou@mail.nhu.edu.tw

Tel: 886+ (0)5+272-1001 ext.56226

[2] Department of Information Management, Nanhua University Chiayi 622, Taiwan

g7151008@mail1.nhu.edu.tw

Tel: 886+ (0)6+215-5319

[3] Institute of information systems and applications, National Tsing Hua University, Taiwan

d949702@oz.nthu.edu.tw

Abstract

Proxy signature schemes can be used in many business applications such as when the original signer is not present to sign important documents. Any proxy signature scheme has to meet the identifiability, undeniability, verifiability and unforgeability security requirements. In some conditions, it may be necessary to protect the proxy signer's privacy from outsiders or third parties. Recently, several studies about proxy signature schemes have been conducted but only Yu et al.' anonymous proxy signature scheme proposed in 2009 attempting to protect the proxy signer's privacy from outsiders. They claimed their scheme can make the proxy signer anonymous. However, based on our research, we determined that this was not the case and the proxy signer's privacy was not anonymous. Hence, in this paper, we propose a new anonymous proxy signature scheme that truly makes the proxy signer anonymous while making it more secure and efficient when compared with Yu et al.'s scheme in 2009. Our proxy signature scheme consists of two constructions. First, we mainly use random numbers and bilinear pairings to attain the anonymous property in our proxy. Secondly, we increase the security, integrity, and efficiency of our proxy through modifications.

Keywords: *Proxy signature, Anonymous, Bilinear pairings, undeniability, unforgeability,*

1. Introduction

In 1996, Mambo et al. [1] first proposed the concept of proxy signature. In their proposal, there are three parties: a user also called original signer, a proxy signer whom is delegated to sign a message on behalf of the original signer, and a verifier who verifies whether a signed message is legal or not. Proxy signature schemes can be used in many business applications such as when the original signer is not present to sign important documents. For example, an important document needs to be signed by the CEO, but the CEO is out of the office or not immediately available. At this time, the CEO can use the proxy signature scheme to designate the general manager or business executive to sign the document on his or her behalf. The signed document will be valid, and can be verified by everyone without the CEO actually signing it.

Since Mambo et al.'s 1996 scheme, many proxy signature schemes have been proposed [2-31]. Overall, generally speaking, there are two main categories of proxy signature schemes, the first category is one-to-one and the other is one-to-many. The one-to-one schemes are [8, 12, 15, 17, 18, 20, 23] and the proxy blind signature [5], which is a special digital signature scheme first introduced by Chaum [25] in 1983. In the one-to-many, there are there two subsets, one is the proxy multi-signature and the other is the (t, n) threshold proxy signature. In the proxy multi-signature [10, 11 14, 16, 26, 27, 28, 29, 30, 31], the original signer has an authorize proxy signer group, each proxy signer has to generate a partials proxy signature. If all partials of signatures are correct, the proxy signature will be generated by summation or multiplication operation of the partial proxy signatures. In the (t, n) threshold proxy signature [3, 6, 16, 24], the original signer can choose the threshold and a proxy signing key is shared by n proxy signers. Any t of proxy signers can cooperatively derive the proxy signing key to sign the message. In any proxy signature, the following security properties are required:

- **Unforgeability** [1, 13, 14, 15, 19, 21, 22, 24, 28]: Only a designated proxy signer can create a valid proxy signature for the original signer. In other words, nobody can forge a valid proxy signature without the delegation of the original signer.
- **Verifiability** [1, 3, 4, 11, 14, 15, 19, 21, 24]: After checking and verifying the proxy signature, a verifier can be convinced that the received message is signed by the proxy signer authorized by the original signer.
- **Undeniability** [1, 3, 4, 15, 19, 21, 24]: The proxy signer cannot repudiate the signature he produced.
- **Identifiability** [1, 3, 4, 14, 15, 24]: Anyone including the original signer can determine the corresponding proxy signer's identity from the proxy signature.

- **Anonymity** [10, 13, 15, 21]: The relating studies about anonymous property in proxy signature scheme aims to protect the identity of the proxy signer, keeping the secrecy of the proxy signer to outsider.

Although proxy signatures incorporate the above mentioned security functions, they still face many threats such as frame attack and public-key substitute attack. The detailed about these two attacks can be referred to studies [30] and [16, 31] respectively. In 2009, Yu et al. [13] further proposed an anonymous proxy signature (APS) scheme which provides anonymity property for proxy multi-signature. In their scheme, there is a group of proxy signers, but only one proxy signer can anonymously signs the message. By using a group of signers, Yu et al. wanted to provide privacy and anonymous protection for the proxy signer such that any other proxy signer cannot know who the real signer is. However, based on our research using transmitted data along with public information, we were able to isolate and identify the proxy signer. More detail of the analysis is described in Section 3.2.

The rest of the paper is organized as follows. In Section 2, we present the basic concepts of bilinear pairings and some related mathematical problems. In Section 3, we review and show the weakness of Yu et al.'s scheme. Section 4 shows the proposed scheme and Section 5 makes comparison in computation efficiency between Yu et al.'s scheme and ours. Finally, a conclusion is given in Section 6.

2. Background

In this section, we describe the concept of bilinear pairings which is used as the mathematical basis of this design.

- **Bilinear Pairings**

Let G_1 be a cyclic additive group of order q generated by a base point P on Elliptic curve and G_2 be a cyclic multiplicative group with the same order. It is considered that solving the Elliptic curve discrete logarithm problem (ECDLP) in G_1 and discrete logarithm problem (DLP) problem in G_2 are difficult. A bilinear map e is defined as $e: G_1 \times G_1 \to G_2$ which has the following properties:

(1) Bilinear: $e(aP,\ bQ) = e(P,\ Q)^{ab}$,where $P,\ Q \in G_1$ and all $a, b \in Z_q^*$.

(2) Non-degeneracy: There exists $P,\ Q \in G_1$ such that $e(P,\ Q) \neq 1$; in other words, the map does not send all pairs in $G_1 \times G_1$ to the identity in G_2.

(3) Computable: There is an efficient algorithm to compute $e(P,\ Q)$ for all $P,\ Q \in G_1$.

3. Review of Yu et al.'s scheme

In this section, we review Yu et al.'s APS scheme [13] and demonstrate that the original APS cannot satisfy the anonymous property in Section 3.2.

3.1 Yu et al.'s APS scheme

There are six phases in Yu et al.'s APS scheme: (1) the parameter generation phase, (2) the key generation phase, (3) the delegation signing phase, (4) the delegation verification phase, (5) the APS generation phase, and (6) the APS verification phase. We describe them as follows, and also depict phases (2), (3), and (4) in figure 1 and phases (5), (6) in figure 2.:

(1) In the parameter generation phase, on input of security parameter k, a system parameter generation algorithm outputs (G_1, G_2, q, e, P), including a cyclic additive group G_1 of order q, a multiplicative group G_2 of the same order, a bilinear map $e: G_1 \times G_1 \rightarrow G_2$, and a generator P of G_1. This algorithm also outputs two cryptographic hash functions: $H_0: \{0, 1\}^* \times G_1 \rightarrow Z_q^*$ and $H_1: \{0, 1\}^* \rightarrow G_1$.

(2) In the key generation phase as shown in Fig. 1, the original signer Alice selects $x_o \in Z_q^*$ as her private key and computes her public key as $Y_o = x_o P$. Each proxy signer $u_i \in U$ randomly selects $x_i \in Z_q^*$ as his/her private key and sets the corresponding public key as $Y_i = x_i P$.

(3) In the delegation signing phase, Alice firstly generates a warrant m_w which contains some explicit descriptions about the delegation relation such as the identities of both the Alice and the proxy signers, the expiration time of the delegation, and the signing power in the warrant. Then, Alice randomly picks a number $r \in Z_q^*$, and computes $R = rP$ and $s = r + x_o H_0(m_w, R) \bmod q$. Finally, Alice sends (m_w, R, s) to the proxy signers in set $U = \{u_1, ..., u_n\}$.

(4) Upon receiving (m_w, R, s), each proxy signer u_i checks if the equation $sP = R + H_0(m_w, R) Y_o$ holds. If it does not, the delegation will be rejected. Otherwise, it will be accepted and each proxy signer u_i computes his/her proxy secret key as $psk_i = s + x_i H_0(m_w, R) \bmod q$.

	Original signer Alice		Proxy signer u_i
Key generation	$x_o \in Z_q^*$ *private key*		$x_i \in Z_q^*$ *private key*
	$Y_o = x_o P$ *public key*		$Y_i = x_i P$ *public key*
Delegation signing	$m_w(warrant)$		
	$r \in Z_q^*$		
	$R = rP$		
	$s = r + x_o H_0(m_w, R) \bmod q$		
		$\xrightarrow{(m_w, R, s)}$	
Delegation verification			$checks\ sP = R + H_0(m_w, R) Y_o$
			$psk_i = s + x_i H_0(m_w, R) \bmod q$

Fig. 1: Key generation, delegation signing and delegation verification phase of Yu et al.'s scheme

(5) In the APS generation phase as shown in Fig. 2, proxy signer $u_s \in U$ with his proxy secret key psk_s signs on a message m on behalf of the original signer, Alice, in an anonymous way. u_s first chooses random numbers $r_i \in Z_q^*$, where $i \in \{1, 2, ..., n\}$ and $i \neq s$, computes both $\sigma_i = r_i P$ and $\sigma_s = \frac{1}{psk_s}\left(H_1(m \| m_w) - \sum_{i \neq s} r_i \left(R + H_0(m_w, R)(Y_o + Y_i)\right)\right)$, and sends $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n, m, m_w, R)$ to the verifier.

(6) In the APS verification phase, given public keys $Y_o, Y_1, ..., Y_n$ and a received anonymous proxy signature σ, the verifier can examine the validity of the signature σ by checking whether the following expression holds.

$$\begin{aligned}
&\prod_{i=1}^{n} e(R + H_0(m_w, R)(Y_o + Y_i), \sigma_i) \\
&= \prod_{i=1, i \neq s}^{n} e\left(R + H_0(m_w, R)(Y_o + Y_i), \sigma_i\right) \bullet e\left(R + H_0(m_w, R)(Y_o + Y_s), \sigma_s\right) \\
&= \prod_{i=1, i \neq s}^{n} e\left(r_i \left(R + H_0(m_w, R)(Y_o + Y_i)\right), P\right) \bullet \\
&\quad e\left(R + H_0(m_w, R)(Y_o + Y_s), \frac{1}{psk_s}\left(H_1(m \| m_w) - \sum_{i \neq s} r_i \left(R + H_0(m_w, R)(Y_o + Y_i)\right)\right)\right)
\end{aligned}$$

$$= \prod_{i=1, i\neq s}^{n} e\left(r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ P\right) \bullet$$

$$e\left(P,\ H_1\left(m \,\|\, m_w\right) - \sum_{i\neq s} r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right)\right)$$

$$= e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)$$

	Proxy signer u_s	**Verifier**
Proxy signature generation	$r_i \in Z_q^*$	
	$\sigma_i = r_i P$	
	$\sigma_s = \frac{1}{psk_s}\left(\left(H_1\left(m \,\|\, m_w\right)\right) - \sum_{i\neq s} r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right)\right)$	
	$\sigma = \left(\sigma_1,\ \sigma_2, \ldots,\ \sigma_n,\ m,\ m_w,\ R\right)$	
	$\xrightarrow{\ \sigma\ }$	
		checks
		$\prod_{i=1}^{n} e(R + H_0(m_w,\ R)(Y_o + Y_i),\ \sigma_i) = e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)$

Fig. 2: APS generation phase and the APS verification phase of Yu et al.'s scheme

3.2 Weakness of Yu et al.'s scheme

After reviewing Yu et al.'s scheme above, we now examine the scheme's anonymous property which they emphasized as follows:

Since R , $H_0\left(m_w,\ R\right)$ and $\left(Y_o + Y_s\right)$ are public, we can obtain $psk_s P$ by

deducing $psk_s P = R + H_0\left(m_w,\ R\right)\left(Y_o + Y_s\right)$, because

$$\begin{aligned} psk_s P &= (s + x_i H_0\left(m_w,\ R\right))P \\ &= (r + x_o H_0\left(m_w,\ R\right) + x_i H_0\left(m_w,\ R\right))P \\ &= (r + (x_o + x_i)H_0\left(m_w, R\right))P \\ &= (rP + ((x_o + x_i)H_0\left(m_w, R\right)P)) \\ &= R + H_0\left(m_w,\ R\right)\left(Y_o + Y_s\right) \end{aligned}$$

Next, we define an inspector X to be $e(psk_xP, \sigma_j)$, where psk_x is u_x's secret proxy signing key, σ_j is a specific sub-signature in σ, and x, $j \in \{1, \dots n\}$. In addition, we define Y to be $\prod_{i=1, i\neq x}^{n} e\left(\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_i\right)\right), \sigma_i\right)$. Then, if there exist some x and j satisfying $X \cdot Y = e\left(P, H_1\left(m \| m_w\right)\right)$, we can determine that x should be equal to j, and u_j is then the right proxy signer. This is because if u_j is the right proxy, then the corresponding sub-signature σ_j must have the factor $\frac{1}{psk_j}$, and therefore only applying the right psk_xP, i.e., $x = j$, can cancel the factor result in the holing of the end. Otherwise, we continue to examine next possible x or j. By doing this way, we can deduce the right proxy signer at most n^2 times which is not computationally infeasible.

For more clarity, we take three proxy signers, u_1, u_2, u_3, as an example. Suppose u_2 is the real proxy signer, then $\sigma_1 = r_1P$, $\sigma_2 = (psk_2)^{-1}(H_1(m \| m_w) - \sum_{i=1, i\neq 1}^{3} r_i(R + H_0(m_w, R)(Y_o + Y_i)))$ and $\sigma_3 = r_3P$.

If we first try σ_1 with different x = 1, 2, 3, then we have three tries as the following.

(1.1) When $x = 1$ and thus $X = e(psk_1P, \sigma_1)$, the value $X \cdot Y$ should be

$$
\begin{aligned}
& e(psk_1P, \sigma_1) \bullet \prod_{i=1, i\neq 1}^{3} e\left(r_i\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_i\right)\right), P\right) \\
= \ & e(P, psk_1\sigma_1) \bullet \prod_{i=1, i\neq 1}^{3} e\left(\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_i\right)\right), r_iP\right) \\
= \ & e(P, psk_1 \bullet r_1P) \bullet e\left(\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_1\right)\right), \sigma_2\right) \bullet e\left(\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_3\right)\right), \sigma_3\right) \\
\neq \ & e\left(P, H_1\left(m \| m_w\right)\right)
\end{aligned}
$$

(1.2) When $x = 2$ and thus $X = e(psk_2P, \sigma_1)$, the value $X \cdot Y$ should be

$$
\begin{aligned}
& e(psk_2P, \sigma_1) \bullet \prod_{i=1, i\neq 2}^{3} e\left(r_i\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_i\right)\right), P\right) \\
= \ & e(P, psk_2\sigma_1) \bullet \prod_{i=1, i\neq 2}^{3} e\left(\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_i\right)\right), r_iP\right) \\
= \ & e(P, psk_2 \bullet r_1P) \bullet e\left(\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_1\right)\right), \sigma_1\right) \bullet e\left(\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_3\right)\right), \sigma_3\right) \\
\neq \ & e\left(P, H_1\left(m \| m_w\right)\right)
\end{aligned}
$$

(1.3) When $x = 3$ and thus $X = e(psk_3P, \sigma_1)$, the value $X \cdot Y$ should be

$$
e(psk_3P, \sigma_1) \bullet \prod_{i=1, i\neq 3}^{3} e\left(r_i\left(R + H_0\left(m_w, R\right)\left(Y_o + Y_i\right)\right), P\right)
$$

$$= e(P,\ psk_3\sigma_1) \bullet \prod_{i=1, i\neq 3}^{3} e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ r_i P\right)$$
$$= e(P,\ psk_3 \bullet r_1 P) \bullet e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_2\right)\right),\ \sigma_1\right) \bullet e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_1\right)\right),\ \sigma_2\right)$$
$$\neq e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)$$

Secondly, if we try σ_2 with different x = 1, 2, 3, then we have three tries as the following.

(2.1) When x = 1 and thus $X = e(psk_1 P,\ \sigma_2)$, the value $X \cdot Y$ should be

$$e(psk_1 P,\ \sigma_2) \bullet \prod_{i=1, i\neq 1}^{3} e\left(r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right), P\right)$$
$$= e(P,\ psk_1\sigma_2) \bullet \prod_{i=1, i\neq 1}^{3} e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ r_i P\right)$$
$$= e(P,\ psk_1 \bullet r_2 P) \bullet e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_1\right)\right),\ \sigma_2\right) \bullet e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_3\right)\right),\ \sigma_3\right)$$
$$\neq e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)$$

(2.2) When x = 2 and thus $X = e(psk_2 P,\ \sigma_2)$, the value $X \cdot Y$ should be

$$e(psk_2 P,\ \sigma_2) \bullet \prod_{i=1, i\neq 2}^{3} e\left(r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ P\right)$$
$$= e(P,\ psk_2\sigma_2) \bullet \prod_{i=1, i\neq 2}^{3} e\left(r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ P\right)$$
$$= e(P,\ psk_2 \bullet \frac{1}{psk_2}\left(H_1\left(m \,\|\, m_w\right) - \sum_{i\neq 2} r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right)\right)) \bullet$$
$$\prod_{i=1, i\neq 2}^{3} e\left(r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ P\right)$$
$$= e(P,\ H_1\left(m \,\|\, m_w\right) - \sum_{i\neq 2} r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right)) \bullet$$
$$\prod_{i=1, i\neq 2}^{3} e\left(r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ P\right)$$

$$= \frac{e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)}{e\left(P,\ r_1\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_1\right)\right)\right) \bullet e\left(P,\ r_3\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_3\right)\right)\right)} \bullet$$
$$e\left(P,\ r_1\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_1\right)\right)\right)\ e\left(P,\ r_3\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_3\right)\right)\right)$$
$$= \frac{e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)}{e\left(\sigma_1,\ \left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_1\right)\right)\right) \bullet e\left(\sigma_3,\ \left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_3\right)\right)\right)} \bullet$$
$$e\left(\sigma_1,\ \left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_1\right)\right)\right)\ e\left(\sigma_3,\ \left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_3\right)\right)\right)$$
$$= e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)$$

(2.3) When x = 3 and thus $X = e(psk_3 P,\ \sigma_2)$, the value $X \cdot Y$ should be

$$e(psk_3 P,\ \sigma_2) \bullet \prod_{i=1, i\neq 3}^{3} e\left(r_i\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right), P\right)$$

$$= e(P,\ psk_3\sigma_2) \bullet \prod_{i=1, i\neq 3}^{3} e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_i\right)\right),\ r_iP\right)$$
$$= e(P,\ psk_3 \bullet r_2P) \bullet e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_1\right)\right),\ \sigma_1\right) \bullet e\left(\left(R + H_0\left(m_w,\ R\right)\left(Y_o + Y_3\right)\right),\ \sigma_2\right)$$
$$\neq e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)$$

From above demonstration, for inspector $X = e(psk_xP,\ \sigma_j)$, only when the subscript $x = j = 2$, the result of $X \cdot Y$ is $e\left(P,\ H_1\left(m \,\|\, m_w\right)\right)$. Therefore, we determined that u_2 is the right proxy signer and the anonymous property that they emphasized is broken.

4. Proposed scheme

In this section, we propose a new APS to Yu et al.'s 2009 APS scheme to correct the anonymous flaw as discovered in Section 3. Our scheme is the same as theirs in the first two phases. The differences are in the last four phases, the delegation signing, delegation verification, APS generation, and APS verification phase. More detail of our APS is shown in Section 4.1. Its correctness is demonstrated in Section 4.2 and the APS requirements are analyzed in Section 4.3.

4.1 The new proposed APSs scheme

In our APS scheme, there also exist an original signer Alice and a proxy signer group $P_i \in \{P_1,\ P_2,\ ...,\ P_n\}$ where $i = 1,\ ...,\ n$ and only one proxy signer of proxy signers group can sign the message. For more clarity, we show our improvement in detail as follows. The proposed scheme consists of six phases: (1) the parameter generation phase, (2) key generation phase, (3) delegation signing phase, (4) delegation verification phase, (5) APS generation phase, and (6) APS verification phase. Phase (1) and (2) are the same as in Yu et al.'s scheme which has been delineated on Section 3.1. We omit these phases in the following but show phase (3) and (4) in figure 3 and phase (5) and (6) in figure 4.

(3) In the delegation signing phase, as shown in Fig. 3, the original signer randomly selects a number $r \in Z_q^*$, and uses r to computes $R = rP$, and $r + x_oH_0(m_w, R) = v$. Then the original signer sends $(m_w,\ R,\ v)$ to the proxy signer group $P_i \in \{P_1,\ P_2,\ ...,\ P_n\}$ with warrant m_w, where warrant contains the records of the original signer and proxy signer's identities, delegation, authorization period, valid period, etc.

	Original signer		**Proxy signer** P_i
Key generation	$x_o \in Z_q^*$ *private key* $Y_o = x_o P$ *public key*		$x_i \in Z_q^*$ *private key* $Y_i = x_i P$ *public key*
Delegation signing	$m_w(warrant)$ $r \in Z_q^*$ $R = rP$ $v = r + x_o H_0(m_w, R)$	$(m_w,\ R,\ v)$ $\longrightarrow$	
Delegation verification			*checks* $vP \,?\!= R + H_0\left(m_w,\ R\right)Y_o$ If it holds, computes $V = vP$ $r_i \in Z_q^*,\ i = 1 \ to \ n$ $c = H_0(r_1,\ ...,\ r_n)$ $U = cP$ $psk_i = r_i^{-1} *\ x_i^{-1} *\ H_0\left(m_w,\ m,\ V,\ U\right)$

Fig. 3: The delegation signing and delegation verification phases of our scheme

(4) In the delegation verification phase, after receiving $(m_w,\ R,\ v)$, each member P_i in the proxy signers group first checks whether the equation $vP = R + H_0\left(m_w,\ R\right)Y_o$ holds. If it doesn't, stop the protocol, otherwise, the message will be accepted. Second, they compute $V = vP$ and each chooses n random numbers $r_i \in Z_q^*,\ i = 1 \ to \ n$, and computes $c = H_0(r_1,\ ...,\ r_n)$, $U = cP$, and $psk_i = r_i^{-1} *\ x_i^{-1} *\ H_0\left(m_w,\ m,\ V,\ U\right)$.

(5) In the APS generation phase, as shown in Fig. 4, let P_s be the proxy signer. He computes $\sigma_i = r_i V$, where $i \in \{1,\ 2,\ ...,\ n\}$ and $i \neq s$ and computes $L = c * x_s^{-1} * V$, then sets $Y,\ \sigma_s,\ p\sigma sum,\ A,\ B,\ C$ and D, as $Y = \sum_{i=1}^{n} Y_i$, $\sigma_s = psk_s * Y = r_s^{-1} *\ x_s^{-1} *\ H_0\left(m_w,\ m,\ V,\ U\right) * Y$, $p\sigma sum = \sum_{i=1}^{n} \sigma_i$, $A = r_s * c * psk_s P$, $B = r_s \sigma_s$, $C = r_s * p\sigma sum$, and $D = r_s * c * V$. Finally, the proxy signer outputs $\sigma = \left(\sigma_1,\ \sigma_2, ...,\ \sigma_n,\ m,\ m_w,\ c,\ A,\ B,\ C,\ D,\ L,\ U,\ V\right)$ as the anonymous proxy signature and sends σ to the verifier.

	Proxy signer P_s	**Verifier**
Anonymous proxy signature generation	$\sigma_i = r_i V,\ where\ i = 1\ to\ n,\ i \neq s$ $\sigma_s = psk_s * Y = r_s^{-1} * x_s^{-1} * H_0(m_w,\ m,\ V,\ U) * Y$ $p\sigma sum = \sum_{i=1}^{n} \sigma_i$ $A = r_s * c * psk_s P$ $B = r_s \sigma_s$ $C = r_s * p\sigma sum$ $D = r_s * c * V$ $L = c * x_s^{-1} * V$ $\sigma = (\sigma_1,\ \sigma_2, \ldots,\ \sigma_n,\ m,\ m_w,\ c,\ A,\ B,\ C,\ D,\ L,\ U,\ V)$	$\xrightarrow{\ \sigma\ }$
Anonymous proxy signature verification		*checks* $(e(D,\ \sum_{i=1}^{n} \sigma_i)) \bullet e(A,\ Y)?$ $= e(cV,\ C-B) \bullet e(L,\ H_0(m_w,\ m,\ V,\ U)Y) \bullet e(U,\ B)$

Fig. 4 Anonymous proxy signature generation phase and the verification phase of our scheme

(6) In APS verification phase, upon receiving the proxy signature the verifier computes $\sum_{i=1}^{n} Y_i = Y$ and checks whether the equation $e(D,\ \sum_{i=1}^{n} \sigma_i) \bullet e(A,\ Y) ?= e(cV,\ C-B) \bullet e(L,\ H_0(m_w,\ m,\ V,\ U)Y)$ $\bullet\ e(U,\ B)$ holds.

If it holds, the verifier accepts the signature, otherwise rejects it.

4.2 Correctness

In the delegation verification phase, the proxy signers can check whether the equation holds $vP ?= R + H_0(m_w,\ R)Y_o$ holds as follows:

Proof 1.

$vP ?= R + H_0(m_w,\ R)Y_o$

$$vP = (r + x_o H_0(m_w, R))P$$
$$= rP + x_o H_0(m_w, R)P$$
$$= R + H_0(m_w, R)Y_o$$

If it holds, the proxy signers can know that the message is sent from the original signer. Because in the verification equation, he uses the original signer's public key Y_o to examine it. If any adversary intercepts the message and modify it, it cannot pass the verify equation.

In the proxy signature verification phase, the following equation gives the correctness of the verification:

Proof 2.

$$(\,e(D, \sum_{i=1}^{n} \sigma_i)) \bullet e(A, Y) = (\prod_{i=1}^{n} e(D, \sigma_i)) \bullet e(A, Y)$$
$$=?\, e(cV, C - B) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$$
$$= (\prod_{i=1, i\neq s}^{n} e(cr_sV, \sigma_i) \bullet e(cr_sV, \sigma_s)) \bullet e(r_s * c * psk_sP, Y)$$
$$= \prod_{i=1, i\neq s}^{n} e(cr_sV, \sigma_i) \bullet e(cr_sV, r_s^{-1} * x_s^{-1} * H_0(m_w, m, V, U) * Y) \bullet e(cP, r_s psk_sY)$$
$$= \prod_{i=1, i\neq s}^{n} e(cr_sV, \sigma_i) \bullet e(cr_sV, r_s^{-1} * x_s^{-1} * H_0(m_w, m, V, U) * Y) \bullet e(cP, r_s\sigma_s)$$
$$= \prod_{i=1, i\neq s}^{n} e(cr_sV, \sigma_i) \bullet e(x_s^{-1} * cV, H_0(m_w, m, V, U) * Y) \bullet e(U, B)$$
$$= \prod_{i=1, i\neq s}^{n} e(cr_sV, \sigma_i) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$$
$$= e(cr_sV, \sum_{i=1, i\neq s}^{n} \sigma_i) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$$
$$= e(cr_sV, p\sigma sum - \sigma_s) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$$
$$= e(cV, r_s(p\sigma sum - \sigma_s)) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$$
$$= e(cV, r_s(p\sigma sum - \sigma_s)) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$$
$$= e(cV, C - B) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$$

4.3 Security analyses

In this section, we demonstrate that our APS scheme can satisfy the security properties as discussed in Section 1 for (1) verifiability, (2) unforgeability, (3) undeniability, (4) anonymity, and (5) identifiability. Among the security properties, we only explore properties (1) – (4). No discussion of property (5) is required since our scheme is anonymous, thus identifiability is not required. Our scheme satisfies these four security properties as follows:

(1) **Verifiability.** In APS verification phase, after checking and verifying the proxy signature σ where $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n, m, m_w, c, A, B, C, D, L, U, V)$, the verifier can calculate to check whether the verification equation $(e(D, \sum_{i=1}^{n} \sigma_i)) \bullet e(A, Y) = ?\ e(cV, C-B) \bullet e(L, H_0(m_w, m, V, U)Y) \bullet e(U, B)$ holds. If it does, the verifier can be convinced that the received message is signed by one of the proxy signer members authorized by the original signer because $Y(=\sum_{i=1}^{n} Y_i)$ and $V(= vP = R + H_0(m_w, R)Y_o)$ are used in the verification equation.

(2) **Unforgeability.** It means that any entity, including the original signer, other than the proxy signer himself cannot generate a valid proxy signature. Only an authorized proxy signer P_s can create a valid proxy signature σ. If any attacker wants to forge a proxy signature, he must be authorized by the original signer signing on a warrant m_w and use the proxy signer's proxy secret key psk_s to compute σ_s. However, this is impossible since the identity of the attacker wasn't in m_w signed by the original signer. Not to mention, he doesn't know psk_s. Under this situation (with a valid σ in hand and without the knowledge of psk_s), even if he wants to (1) fake the proxy signer key as psk_s', (2) change value c to c', or (3) randomly select $r_s' \in Z_q^*$, trying to counterfeit the proxy signature, we demonstrate that his attempts deem to fail. We demonstrate the reasons for the failures of these three cases in the following.

Case 1. If an attacker does not know the proxy secret key psk_s, he cannot generate valid $\sigma_s(= psk_s * Y)$, $p\sigma sum(=\sum_{i=1}^{n} \sigma_i)$, $A(= r_s * c * psk_s P)$, $B(= r_s \sigma_s)$, and $C(= r_s * p\sigma sum)$. Even if he uses a random psk_s' to sign the message, since $psk_s = r_s^{-1} * x_s^{-1} * H_0(m_w, m, V, U)$, he cannot evaluate the right value x_s^{-1} to compute L to be successfully verified in the verification equation.

Case 2. Because c is changed to c', at least one of the random numbers r_i should also be modified. Without loss of generality, we let $r_i = r_1 \neq r_s$. Accordingly, all the parameters $U(= cP)$,

$psk_s(=r_s^{-1} * x_s^{-1} * H_0(m_w,\ m,\ V,\ U))$, $\sigma_s(=psk_s * Y)$, $p\sigma sum(=\sum_{i=1}^{n} \sigma_i)$, $A(=r_s * c * psk_s P)$, $B(=r_s\sigma_s)$, $C(=r_s * p\sigma sum)$, $D(=r_s * c * V)$, and $L(=c * x_s^{-1} * V)$ are all changed as well. That is, $\sigma' = (\sigma_1',\ \sigma_2, ...,\ \sigma_s',\ \sigma_{s+1}, ...,\ \sigma_n,\ m,\ m_w,\ c',\ A',\ B',\ C',\ D',\ L',\ U',\ V)$. Apparently, the verification equation ($e(D,\ \sum_{i=1}^{n} \sigma_i)) \bullet e(A,\ Y) = ?\, e(cV,\ C-B) \bullet e(L,\ H_0(m_w,\ m,\ V,\ U)Y) \bullet e(U,\ B)$ cannot hold. Below, we only show the inequality of portion of the verification equation $e(A',\ Y) = e(U',\ B')$.

$$\begin{aligned}
&e(A',\ Y) = e(r_s' * c' * psk_s' P,\ Y) \\
&= e(c'P,\ r_s' psk_s' Y) \\
&= e(c'P,\ r_s' \sigma_s) \\
&\neq e(U,\ B)
\end{aligned}$$

Case 3. In this case, if any attacker randomly selects $r_s' \in Z_q^*$ and tries to generate the valid proxy signature σ'. Accordingly, the parameters $U(=cP)$, $psk_s(=r_s^{-1} * x_s^{-1} * H_0(m_w,\ m,\ V,\ U))$, $\sigma_s(=r_s^{-1} * x_s^{-1} * H_0(m_w,\ m,\ V,\ U) * Y)$, $p\sigma sum(=\sum_{i=1}^{n} \sigma_i)$, $A(=r_s * c * psk_s P)$, $B(=r_s\sigma_s)$, $C(=r_s * p\sigma sum)$, $D(=r_s * c * V)$, and $L(=c * x_s^{-1} * V)$ are all changed as well, similar to **Case 2**. Finally the signature becomes $\sigma' = (\sigma_1,\ \sigma_2, ..., \sigma_s',\ \sigma_{s+1}, ...,\ \sigma_n,\ m,\ m_w,\ c',\ A',\ B',\ C',\ D',\ L',\ U',\ V)$. As in **Case 2**, when the verifier checks whether $e(A',\ Y) = e(U,\ B')$ holds, he will found it doesn't.

(3) Undeniability. As in Section 4.2 **Proof 2**, the verifier uses the verification equation $(\prod_{i=1}^{n} e(D,\ \sigma_i)) \bullet e(A,\ Y) = e(cV,\ C-B) \bullet e(L,\ H_0(m_w,\ m,\ V,\ U)Y) \bullet e(U,\ B)$ to check whether the proxy signature comes from one member of the proxy signer group. Since in the equation $V(=vP = R + H_0(m_w,\ R)Y_o)$ includes the original signer's public key Y_o and $Y = \sum_{i=1}^{n} Y_i$, it means the original signer and the proxy signer group cannot repudiate their participations in the signature creation.

(4) Anonymity. In the APS generation phase, all the parameters *A*, *B*, *C*, *D*, and *L* have to be multiplied by $r_s \in Z_q^*$ to make the proxy signature σ anonymous. If any attacker wants to know who is the real signer, he must know the value r_s to use r_s^{-1} to unrandomize all parameters to get $A'(=c'*psk_s'P)$, $B(=\sigma_s')$, $C'(=p\sigma sum')$, $D'(=c'*V)$, and $\sigma_s'(=x_s^{-1}*H_0(m_w, m, V, U)*Y)$. But now $\sigma_i = r_i V,\ i \neq s$, each is randomized by r_i respectively. Even the attack knows r_s, without the knowledge of r_i and x_s, he cannot know who the real signer is. Not to mention in reality, he in reality cannot know the value of r_s. It means that anyone cannot know who signs the signature. So our APS scheme can achieve the anonymous property.

5. Comparisons

In this section, we compare the computational cost between Yu et al.'s APS scheme and ours and summarize the result in Table 1. We denote e as the pairing operation Pm and Pa as the point multiplication and point addition on G_1 respectively, and n denote the number of proxy signers. In Yu et al.'s APS scheme, the generation and verification of *psk* in column 3 of Table 1 should be $2n\,Pm + nPa$ instead of $(n+1)\,Pm$ operations. Because in Yu et al.'s scheme, the generation and verification of psk are $R = rP$ and $sP = R + H_0(m_w, R)Y_0$, the sP should be computed by n proxy signers. The APS verification should be $(n+1)e+n\,Pm+2n\,Pa$ rather than the original $(n+1)e+n\,Pm+(n+1)\,Pa$ as listed in the table of [13]. From Table 1, we can see that our scheme is more efficient then Yu et al.'s.

Table 1: Comparison of computational costs of our scheme and Yu's scheme

	Key generation	**Generation and verification of *psk***	**APS generation**	**APS verification**
Yu's scheme	*Same*	*2nPm+nPa*	*(3n-2)Pm+(n+1)Pa*	*(n+1)e+nPm+ 2nPa*
Our scheme	*Same*	*4nPm+nPa*	*(n+5)Pm+nPa*	*5e+2Pm+(n+1)Pa*

6. Conclusions

In 2009, Yu et al. proposed an APS scheme attempting to protect the proxy signer's privacy. Based on our analysis using the above information, we determined that Yu et al.'s original protocol was not secured and could not satisfy the anonymous property. Accordingly, we proposed a novel APS scheme to reach the goal. Our construction uses a random number r_s, one-way hash function and bilinear pairings to make the proxy signature attain the anonymous property. After analyses and comparisons, we conclude that our new protocol is a significant improvement against attackers concerning security and is more efficient in computation overhead as demonstrated in this paper.

Reference

[1] M. Mambo, K. Usuda, E. Okamoto, "Proxy signature: delegation of the power to sign messages, "*IEICE Trans. Fundam. Volume E79-A(9), September*, pp.1338–1354, 1996.

[2] S. Saeednia, "An identity-based society oriented signature scheme with anonymous signers, "*Information Processing Letters 83*, pp.295–299, 2002.

[3] C. L. Hsua, T. S. Wu, T. C. Wu, "Group-oriented signature scheme with distinguished signing authorities, "*Future Generation Computer Systems 20*, pp.865–873, 2004.

[4] C. Y. Lin, T. C. Wu, F. Zhang, and J. J. Hwang, "New identity-based society oriented signature schemes from pairings on elliptic curves, "*Applied Mathematics and Computation 160* , pp.245–260, 2005.

[5] R. X. Lu, Z. F. Cao, and Y. Zhou, "Proxy blind multi-signature scheme without a secure channel, "*Applied Mathematics and Computation 164,* pp.179–187, 2005.

[6] H. F. Huang, and C. C. Chang, "A novel efficient (t, n) threshold proxy signature scheme, "*Information Sciences 176*, pp.338–1349, 2006.

[7] Z. Shao, "Certificate-based verifiably encrypted signatures from pairings, " *Information Sciences 178*, pp.2360 – 2373, 2008.

[8] B. Kang, C. Boyd, and E. Dawson, "Identity-based strong designated verifier signature schemes: Attacks and new construction, "*Computers and Electrical Engineering*, 2008.

[9] J. Zhang, and J. Mao, "A novel ID-based designated verifier signature scheme, "*Information Sciences 178*, pp.766 – 773, 2008.

[10] K. L. Wu, J. Zou, X. H. Wei, and F. Y. Liu, "Proxy group signature: a new anonymous proxy signature scheme, "*Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming,* pp.12-15, 2008.

[11] Z. Shao, "Improvement of identity-based proxy multi-signature scheme, "*The Journal of Systems and Software 82*, pp.794 – 800, 2009.

[12] Z. H. Liu, Y. P. Hu, X. S. Zhang, and H. Ma, "Secure proxy signature scheme with fast revocation in the standard model, "*The Journal of China Universities of*

Posts and Telecommunications, 16(4): 116–124, 2009.

[13] Y. Yu, C. Xu, X. Huang, and Y. Mu, “An efficient anonymous proxy signature scheme with provable security, ”*Computer Standards & Interfaces 31*, pp.348 – 353, 2009.

[14] F. Cao, and Z. Cao, “A secure identity-based proxy multi-signature scheme, ”*Information Sciences 179*, pp.292 – 302, 2009.

[15] A. Yang, and W. P. Peng, “A Modified Anonymous Proxy Signature With a Trusted Party, ”*First International Workshop on Education Technology and Computer Science*, 2009.

[16] J. H. Hu, and J. Z. Zhang, “Cryptanalysis and improvement of a threshold proxy signature scheme, ”*Computer Standards & Interfaces 31*, pp.169 – 173, 2009.

[17] Y. Yu, C. X. Xu, X. S. Zhang, and Y. J. Liao, “Designated verifier proxy signature scheme without random oracles, ”*Computers and Mathematics with Applications 57*, pp.1352–1364, 2009.

[18] J. H. Zhang, C. L. Liu, and Y. I. Yang, “An efficient secure proxy verifiably encrypted signature scheme, ”*Journal of Network and Computer Applications 33*, pp.29–34, 2010.

[19] B. D. Wei, F. G. Zhang, and X. F. Chen, “ID-based Ring Proxy Signatures, ”*ISIT2007, Nice, France*, pp.24–29, 2007.

[20] T. S. Wu, and H. Y Lin, “Efficient self-certified proxy CAE scheme and its variants, ”*The Journal of Systems and Software 82*, pp.974 – 980, 2009.

[21] Y. F. Chung, Z. Y. Wu, and T. S. Chen, “Ring signature scheme for ECC-based anonymous signcryption, ”*Computer Standards & Interfaces 31*, pp.669-674, 2009.

[22] L. J. Mordell, Diophantine equations, Academic Press. *ISBN 0-12-506250-8*, 1969.

[23] S. Lal, V. Verma, “Identity base strong designated verifier proxy signature schemes, ”*Cryptography eprint Archive Report 394*, 2006.

[24] C. H. Yang, S.F. Tzeng, M. S. Hwang, “On the efficiency of nonrepudiable threshold proxy signature scheme with known signers, ”*Syst. Softw. 73 (3)* pp.507 – 514, 2004.

[25] D. Chaum, “Blind signatures for untraceable payments, ”*Advances in Cryptology - Crypto '82, Springer-Verlag*, pp.199-203, 1983.

[26] H. Xiong, J. Hua, Z. Chen, F. Li, “On the security of an identity based multi-proxy signature scheme, ”*Computers and Electrical Engineering’ 37*, pp. 129-135, 2011.

[27] Y. Sun, C. Xu, Y. Yu, Y. Mu, “Strongly unforgeable proxy signature scheme secure in the standard model,”*Journal of Systems and Software’84*,

pp.1471-1479, 2011.

[28] Y. Sun, C. Xu, Y. Yu, B. Yang, "Improvement of a proxy multi-signature scheme without random oracles,"*Computer Communications 34*, pp. 257-263, 2011.

[29] Z. Liu, Y. Hu, X. Zhang, H. Ma, "Provably secure multi-proxy signature scheme with revocation in the standard model," *Computer Communications 34*, pp. 494-501, 2011.

[30] H. Bao, Z. Cao, S. Wang, "Improvement on Tzeng et al.'s nonrepudiable threshold multi-proxy multi-signature scheme with shared verification," *Appl. Math. Comput. 169*, pp. 1419-1430, 2005.

[31] J. Li, Z. Cao, "An improvement of a threshold proxy signature scheme," *Comput. Res. Dev. 39 (11)* pp. 1513-1518, 2002.