Research Article
A Novel Anonymous Proxy Signature Scheme

Jue-Sam Chou

Department of Information Management, Nanhua University, Chiayi 622, Taiwan

Correspondence should be addressed to Jue-Sam Chou, jschou@mail.nhu.edu.tw

Received 30 April 2012; Accepted 18 July 2012

Academic Editor: Joonki Paik

Copyright © 2012 Jue-Sam Chou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, several studies about proxy signature schemes have been conducted. In 2009, Yu et al. proposed an anonymous proxy signature scheme attempting to protect the proxy signer’s privacy from outsiders. They claimed that 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. Our proxy signature scheme consists of two contributions. First, we mainly use random numbers and bilinear pairings to attain the anonymous property. Secondly, we increase the security and efficiency of our proxy in the design.

1. Introduction

Proxy signature schemes can be used in many business applications such as signing important documents when the original signer is not present. 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. Any proxy signature scheme must meet the identifiability, undeniability, verifiability, and unforgeability security requirements. It may be necessary to protect the proxy signer’s privacy from outsiders or third parties. In 1996, Mambo et al. [1] first proposed the concept of proxy signature. In their proposal, there are three parties: a user 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.

Since Mambo et al.’s 1996 scheme, many proxy signature schemes have been proposed [1–27] (some other schemes though are signature schemes whereas not proxy signatures such as [28–33]). 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. In the former, there is one original signer and one proxy signer, but in the latter, except for the original signer, there are a group of proxy signers. The one-to-one schemes are [4, 7, 10, 12, 13, 15–17, 25–27] and the proxy blind signature [2], which is based on a special digital signature scheme first introduced by Chaum [34] in 1983. In the one-to-many, there are two subsets, one is the proxy multisignature and the other is the (t, n) threshold proxy signature. In the proxy multisignature [5, 6, 9, 19–22], the original signer has an authorized proxy signer group, each proxy signer has to generate a partial proxy signature. If all partials of signatures are correct, the proxy signature will be generated by summation or multiplication operations of the partial proxy signatures. In the (t, n) threshold proxy signature [3, 11, 18, 23, 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 four security properties are required.

(i) Unforgeability. 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.
(ii) Verifiability. 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.

(iii) Undeniability. The proxy signer cannot repudiate the signature he produced.

(iv) Identifiability. Anyone including the original signer can determine the corresponding proxy signer’s identity from the proxy signature. That is, from the proxy signature any verifier can determine the proxy signer’s identity.

Although proxy signatures incorporate the above-mentioned security functions, they still face many threats such as man-in-the-middle, replay, frame, and public-key substitute attacks. In frame attacks [23], the malicious original signer can forge a signature after intercepting sent information and the forged signature can be accepted by the original signer. After checking and verifying the proxy signature, a verifier can be convinced that the received message is from the proxy signer. That is, from the proxy signature any verifier can determine the proxy signer’s identity from the proxy signature. Any verifier who the signer is. And the anonymity revocation indicates that when designing a proxy signature scheme, care should be taken to avoid these kinds of attacks.

Researchers, Shum and Wei’s [26] and Yang, and Peng [10], presented two one-to-one anonymous proxy signature (APS) schemes. They point that an APS scheme should possess not only the security features of unforgeability, verifiability, and undeniability, but also the properties of anonymity and anonymity revocation. The anonymity means that only one of the proxy signers can sign the message in the proxy signer group, other proxy signers cannot know who the signer is. And the anonymity revocation indicates that once required, the proxy signer can assure the others that he is the real signer. However, N. Y. Lee and M. F. Lee [27] indicate that Shum and Wei’s scheme [26] violates the property of the unforgeability. Yang and Peng [10] therefore proposed a modified one-to-one APS scheme. In 2009, Yu et al. [8] first proposed a one-to-many APS scheme. 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. want to provide privacy and anonymous protection for the real proxy signer. They claim that their scheme is provably secure. However, based on our research by just using some of the transmitted data along with public information, we were able to isolate and identify the proxy signer. More details of the analysis were 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 of 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 for this design.

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

1. Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, where $P, Q \in G_1$ and all $a, b \in \mathbb{Z}_q^*$
2. Nondegeneracy: 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. Computability: 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 [8] 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.

1. In the parameter generation phase, on input of security parameter $k$, a system parameter generation algorithm outputs 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: $H_0 : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}_q^*$ and $H_1 : \{0,1\}^* \rightarrow G_1$.
2. In the key generation phase as shown in Figure 1, the original signer Alice selects $x_i \in \mathbb{Z}_q^*$ as her private key and computes her public key as $Y_o = x_iP$. Each proxy signer $u_i \in \mathcal{U}$ randomly selects $x_i \in \mathbb{Z}_q^*$ as his/her private key and sets the corresponding public key as $Y_i = x_iP$.
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 Alice and the proxy signers, the expiration time of the delegation, and the signing power in the warrant. Then, Alice randomly picks a number $r \in \mathbb{Z}_q^*$ and computes $R = rP$ and $s = r + x_iH_0(m_w, R) \mod q$. Finally, Alice sends $(m_w, R, s)$ to the proxy signers in set $\mathcal{U} = \{u_1, \ldots, u_n\}$.
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 $psk_i = s + x_iH_0(m_w, R) \mod q$. 

In the APS generation phase as shown in Figure 2, proxy signer $u_i \in \mathcal{U}$ signs on a message $m$ with his proxy secret key $\text{psk}_i$, on behalf of the original signer, $Alice$, in an anonymous way. $u_i$ first chooses random numbers $r_i \in \mathbb{Z}_q^*$, where $i \in \{1, 2, \ldots, n\}$ and $i \neq s$, computes both $\sigma_i = r_i P$ and $\sigma_s = (1/\text{psk}_i)(H_1(m\|m_w) - \sum_{i \neq s} r_i(R + H_0(m_w, R)(Y_o + Y_i)))$, and sends $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n, m, m_w, R)$ to the verifier.

\[(5)\]

In the APS verification phase, given public keys $Y_o, Y_1, \ldots, Y_n$ and a received anonymous proxy signature $\sigma$, the verifier can examine the validity of the signature $\sigma$ by checking whether the following expression holds:

\[
\prod_{i=1,i \neq s}^n e(r_i(R + H_0(m_w, R)(Y_o + Y_i)), P) \\
= \prod_{i=1,i \neq s}^n e(R + H_0(m_w, R)(Y_o + Y_i), \sigma_i) \\
\cdot e\left(\frac{1}{\text{psk}_i}, H_1(m\|m_w) - \sum_{i \neq s} r_i(R + H_0(m_w, R)(Y_o + Y_i))\right) \\
= e(P, H_1(m\|m_w)).
\]

\[
(1)
\]

### 3.2. Weakness of Yu et al’s Scheme

After reviewing Yu et al.'s scheme above, we now explain the violation of the scheme’s anonymous property which they emphasized as follows.

Since $R, H_0(m_w, R)$, and $(Y_o + Y_s)$ are public, we can obtain $\text{psk}_i P$ by deducing $\text{psk}_i P = R + H_0(m_w, R)(Y_o + Y_s)$ because

\[
\text{psk}_i P = (s + x_i H_0(m_w, R)) P \\
= (r + x_i H_0(m_w, R)) P \\
= (r + (x_o + x_i) H_0(m_w, R)) P \\
= (r P + (x_o + x_i) H_0(m_w, R)) P \\
= R + H_0(m_w, R)(Y_o + Y_s).
\]

\[(2)\]

Next, we define an inspector $X$ to be $e(\text{psk}_x, P, \sigma_j)$, where $\text{psk}_x$ is $u_x$’s secret proxy key, $\sigma_j$ is a specific subsignature in $\sigma$, and $x, j \in \{1, \ldots, n\}$. In addition, we define $Y$ to be $\prod_{i=1,j \neq s}^n e((R + H_0(m_w, R)(Y_o + Y_i)), \sigma_i)$. Then, if there exist some $x$ and $j$ satisfying $X \cdot Y = e(P, H_1(m\|m_w))$, we can determine that $x$ should be equal to $j$, and $u_i$ is then the right proxy signer. This is because if $u_i$ is the right proxy signer, then the corresponding subsignature $\sigma_j$ must have the factor $1/\text{psk}_i$, and therefore only applying the right $\text{psk}_x P$, that is, $x = j$, can cancel the factor result in the holing of the end. Otherwise, we continue to examine next possible $x$ or $j$. By

![Figure 1: Key generation, delegation signing, and delegation verification phases of Yu et al.'s scheme.](image-url)
doing this way, we can deduce the right proxy signer at most \( n^2 \) times.

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 \( \sigma_1 = r_1 P, \sigma_2 = (psk_2)^{-1} (H_1(m || m_w) - \sum_{i=1, i \neq 1}^3 r_i (R + H_0(m_w, R)(Y_o + Y_i))) \) and \( \sigma_3 = r_3 P \).

If we first try \( \sigma_1 \) with different \( x = 1, 2, 3 \), then we have three tries as in the following.

(1.1) When \( x = 1 \) and thus \( X = e(ps_k_1, \sigma_1) \), the value \( X \cdot Y \) should be

\[
e(psk_1, \sigma_1) \cdot \prod_{i=1, i \neq 2}^3 e(r_i (R + H_0(m_w, R)(Y_o + Y_i)), P)
= e(P, psk_1 \cdot r_1 P) \cdot e((R + H_0(m_w, R)(Y_o + Y_1)), \sigma_2)
\]

\[
\ne(P, H_1(m || m_w)).
\]

(3)

(1.2) When \( x = 2 \) and thus \( X = e(ps_k_2, \sigma_1) \), the value \( X \cdot Y \) should be

\[
e(psk_2, \sigma_1) \cdot \prod_{i=1, i \neq 2}^3 e(r_i (R + H_0(m_w, R)(Y_o + Y_i)), P)
= e(P, psk_2 \cdot r_1 P) \cdot e((R + H_0(m_w, R)(Y_o + Y_1)), \sigma_2)
\]

\[
\ne(P, H_1(m || m_w)).
\]

(4)

(1.3) When \( x = 3 \) and thus \( X = e(ps_k_3, \sigma_1) \), the value \( X \cdot Y \) should be

\[
e(psk_3, \sigma_1) \cdot \prod_{i=1, i \neq 2}^3 e((R + H_0(m_w, R)(Y_o + Y_1)), P)
= e(P, psk_3 \cdot r_1 P) \cdot e((R + H_0(m_w, R)(Y_o + Y_2)), \sigma_2)
\]

\[
\ne(P, H_1(m || m_w)).
\]

(5)

Secondly, if we try \( \sigma_2 \) with different \( x = 1, 2, 3 \), then we have three tries as in the following.

(2.1) When \( x = 1 \) and thus \( X = e(ps_k_1, \sigma_2) \), the value \( X \cdot Y \) should be

\[
e(psk_1, \sigma_2) \cdot \prod_{i=1, i \neq 2}^3 e((R + H_0(m_w, R)(Y_o + Y_1)), P)
= e(P, psk_1 \cdot r_2 P) \cdot e((R + H_0(m_w, R)(Y_o + Y_1)), \sigma_3)
\]

\[
\ne(P, H_1(m || m_w)).
\]

(6)
(2.2) When \( x = 2 \) and thus \( X = e(psk_2P, \sigma_2) \), the value \( X \cdot Y \) should be

\[
e(psk_2P, \sigma_2) \cdot \prod_{i=1,i \neq 1}^{3} e(r_i(R + H_0(m_w, R)(Y_o + Y_i)), P)
\]

\[
e(P, psk_2\sigma_2) \cdot \prod_{i=1,i \neq 1}^{3} e(r_i(R + H_0(m_w, R)(Y_o + Y_i)), P)
\]

\[
e \left( P, psk_2 \cdot \frac{1}{psk_2} \left( H_1(m \parallel m_w) - \sum_{i \neq 3} r_i(R + H_0(m_w, R)(Y_o + Y_i)) \right) \right)
\]

\[
\cdot \prod_{i=1,i \neq 1}^{3} e(r_i(R + H_0(m_w, R)(Y_o + Y_i)), P)
\]

\[
e \left( P, H_1(m \parallel m_w) - \sum_{i \neq 1} r_i(R + H_0(m_w, R)(Y_o + Y_i)) \right)
\]

\[
\cdot \prod_{i=1,i \neq 1}^{3} e(r_i(R + H_0(m_w, R)(Y_o + Y_i)), P)
\]

\[
e \left( P, H_1(m \parallel m_w) \right)
\]

\[
e \left( P, r_1(R + H_0(m_w, R)(Y_o + Y_1)) \right) \cdot e(P, r_3(R + H_0(m_w, R)(Y_o + Y_3)))
\]

\[
e \left( P, r_1(R + H_0(m_w, R)(Y_o + Y_1)) \right) \cdot e(P, r_3(R + H_0(m_w, R)(Y_o + Y_3)))
\]

\[
e \left( P, H_1(m \parallel m_w) \right)
\]

\[
e \left( P, \sigma_1, (R + H_0(m_w, R)(Y_o + Y_1)) \right) \cdot e(\sigma_3, (R + H_0(m_w, R)(Y_o + Y_3)))
\]

\[
e \left( P, \sigma_1, (R + H_0(m_w, R)(Y_o + Y_1)) \right) \cdot e(\sigma_3, (R + H_0(m_w, R)(Y_o + Y_3)))
\]

\[
e \left( P, H_1(m \parallel m_w) \right).
\]

(7)

(2.3) When \( x = 3 \) and thus \( X = e(psk_3P, \sigma_2) \), the value \( X \cdot Y \) should be

\[
e(psk_3P, \sigma_2) \cdot \prod_{i=1,i \neq 2}^{3} e(r_i(R + H_0(m_w, R)(Y_o + Y_i)), P)
\]

\[
e(P, psk_3\sigma_2) \cdot \prod_{i=1,i \neq 2}^{3} e((R + H_0(m_w, R)(Y_o + Y_i)), P)
\]

\[
e(P, psk_3 \cdot r_2P) \cdot e((R + H_0(m_w, R)(Y_o + Y_1)), s_1)
\]

\[
e((R + H_0(m_w, R)(Y_o + Y_3)), s_3)
\]

\[
\ne(P, H_1(m \parallel m_w)).
\]

From the above demonstration, for inspector \( X = e(psk_iP, \sigma_j) \), only when the subscript \( x = j = 2 \), the result of \( X \cdot Y \) is \( e(P, H_1(m \parallel m_w)) \). 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 one-to-many 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 details of our APS are shown in Section 4.1. Its correctness is demonstrated in Section 4.2 and the APS requirements are analyzed in Section 4.3. Before describing our protocol, we define some basic notations listed in Table 1.

4.1. The New Proposed APS Scheme. In our APS scheme, there also exist an original signer Alice and a proxy signer
Figure 4. Section 3.1. We omit these phases in the following but show same as in Yu et al.’s scheme, which has been delineated in
and (6) APS verification phase. Phases (1) and (2) are the
delegation verification phase, (5) APS generation phase,
phase, (2) key generation phase, (3) delegation signing phase,
scheme consists of six phases: (1) the parameter generation
we show our scheme in detail as follows. The proposed
proxy signers group can sign the message. For more clarity,
phases (3) and (4) in Figure 3 and phases (5) and (6) in

Table 1: The definitions of used notations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>A cyclic additive group on an Elliptic Curve with order $q$ generated and a base point $P$</td>
</tr>
<tr>
<td>$G_2$</td>
<td>A cyclic multiplicative group with order $q$</td>
</tr>
<tr>
<td>$e$</td>
<td>A bilinear map which is defined as $e : G_1 \times G_1 \rightarrow G_2$</td>
</tr>
<tr>
<td>$\mathcal{P}_1, \mathcal{P}_2, \ldots , \mathcal{P}_n$</td>
<td>A proxy signer group</td>
</tr>
<tr>
<td>$m$</td>
<td>A message to be signed</td>
</tr>
<tr>
<td>$m_w$</td>
<td>A warrant which contains the original signer’s and proxy signer’s identities, delegation, authorization period, valid period, and so forth</td>
</tr>
<tr>
<td>$\mathcal{P}_i$</td>
<td>A proxy signer in the proxy signer group ${\mathcal{P}_1, \mathcal{P}_2, \ldots , \mathcal{P}_n}$</td>
</tr>
<tr>
<td>$\mathcal{P}_i$</td>
<td>The real signer in the proxy signer group ${\mathcal{P}_1, \mathcal{P}_2, \ldots , \mathcal{P}_n}$</td>
</tr>
<tr>
<td>$(x_0, Y_0)$</td>
<td>The private/public key pair of the original signer</td>
</tr>
<tr>
<td>$(x_i, Y_i)$</td>
<td>The private/public key pair of $\mathcal{P}_i$</td>
</tr>
<tr>
<td>$\text{psk}_i$</td>
<td>The proxy secret key computed by $\mathcal{P}_i$</td>
</tr>
<tr>
<td>$r_i$</td>
<td>A random integer in $Z_q^*$</td>
</tr>
<tr>
<td>$\parallel$</td>
<td>A concatenation of two strings</td>
</tr>
<tr>
<td>$H_0(\cdot)$</td>
<td>A hash function mapping from ${0,1}^* \times G_1$ to $Z_q^*$</td>
</tr>
<tr>
<td>$H_1(\cdot)$</td>
<td>A hash function mapping from ${0,1}^<em>$ to $Z_q^</em>$</td>
</tr>
<tr>
<td>$H_2(\cdot)$</td>
<td>A hash function mapping from ${0,1}^* \times G_1 \times G_1$ to $Z_q^*$</td>
</tr>
<tr>
<td>$\sigma_i$</td>
<td>The random point ($= r_i V$) constructed by $\mathcal{P}_i$ to stand for the signatures as if they were really made by $\mathcal{P}_i$, where $s \neq i$, correspondingly</td>
</tr>
<tr>
<td>$\sigma_s$</td>
<td>The signature generated by $\mathcal{P}_s$, where $s \neq i$</td>
</tr>
<tr>
<td>$p\sigma$ sum</td>
<td>$= (\sum_{i \neq s=1}^n \sigma_i) + \sigma_s$ (the summation of partial proxy signatures)</td>
</tr>
<tr>
<td>$A, B, C, D, L, U, V, R$</td>
<td>The points in $G_1$</td>
</tr>
</tbody>
</table>

group $\{\mathcal{P}_1, \mathcal{P}_2, \ldots , \mathcal{P}_n\}$, and only one proxy signer in the
proxy signers group can sign the message. For more clarity,
we show our scheme 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. Phases (1) and (2) are the
same as in Yu et al.’s scheme, which has been delineated in
Section 3.1. We omit these phases in the following but show phases (3) and (4) in Figure 3 and phases (5) and (6) in
Figure 4.

(3) In the delegation signing phase, as shown in
Figure 3, the original signer randomly selects a
number $r \in Z_q^*$ and uses $r$ to compute $R = rP$ and $v = r + x_0 H_0(m_w, R)$. Then, the original
signer sends $(m_w, R, v)$ to each proxy signer $\mathcal{P}_i \in 
\{\mathcal{P}_1, \mathcal{P}_2, \ldots , \mathcal{P}_n\}$ with warrant $m_w$, where warrant contains the records of the original signer’s and proxy
signer’s identities, delegation, authorization period, valid period, and so forth.

(4) In the delegation verification phase, after receiving $(m_w, R, v)$ the proxy signer $\mathcal{P}_i$ first checks whether
the equation $vP = R + H_0(m_w, R)Y_s$ holds. If it does not,
stop the protocol, otherwise, he stores $(m_w, R)$. Second, when signing message $m$, $\mathcal{P}_i$ chooses random numbers $r_i \in Z_q^*$, $i = 1$ to $n$, and $V = vP$ computes $c = H_1(r_1V \cdots r_nV)$, $U = cP$, and the proxy secret key, $\text{psk}_i = r_i^{-1} * x_i^{-1} * H_2(m_w || m, V, U)$.

(5) In the APS generation phase, as shown in Figure 4,
let $\mathcal{P}_i$ be the real proxy signer. He computes $\sigma_i = r_i V$, where $i \in \{1, 2, \ldots , n\}$ and $i \neq s$ and computes $L = c * x_s^{-1} * V$, then sets $Y, \sigma_s, p\sigma$ sum = $\sum_{i=1}^n \sigma_i, A, B, C, D, a$, $Y, Y = r_s^{-1} * x_s^{-1} * H_2(m_w || m, V, U)$, $A = r_s * c * \text{psk}_P, B = r_s \sigma_s, C = r_s * p\sigma$ sum, and $D = r_s * c * V$, respectively. Finally, $\mathcal{P}_i$ outputs
Advances in Multimedia 7

Checks

\[ x_0 \in \mathbb{Z}_q^* \text{ (private key)} \]

\[ Y_0 = x_0 P \text{ (public key)} \]

Delegation signing \( m_w \) (warrant)

\[ r \in \mathbb{Z}_q^* \]

\[ R = rP \]

\[ v = r + x_0 H_0(m_w, R) \]

Delegation verification

\[ (m_w, R, v) \rightarrow \]

Checks

\[ vP = R + H_0(m_w, R) Y_0 \]

If it holds, stores \((m_w, R)\).

When signing on \( m \), he computes

\[ V = vP \]

\[ r_i \in \mathbb{Z}_q^*, i = 1 \text{ to } n \]

\[ c = H_1(r_1 || \cdots || r_n) \]

\[ U = cP \]

\[ \text{psk}_i = r_i^{-1} x_i^{-1} H_2(m_w || m, V, U) \]

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

Proxy signer \( P_i \)

Anonymous proxy signature generation

\[ a_i = r_i V, \text{ where } i = 1 \text{ to } n, i \neq s \]

\[ a_s = \text{psk}_s * Y = r_s^{-1} x_s^{-1} H_2(m_w || m, V, U) * Y \]

\[ \rho \text{sum} \sum_{i=1}^{n} a_i \]

\[ A = r_s * c * \text{psk}_s P \]

\[ B = r_s a_i \]

\[ C = r_s * \rho \text{sum} \]

\[ D = r_s * c * V \]

\[ L = c * x_s^{-1} V \]

\[ \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n, m, m_w, c, A, B, C, D, L, U, V) \]

Anonymous proxy signature verification

Checks

\[ \prod_{i=1}^{n} e(D, a_i) \cdot e(A, Y) \]

\[ = e(Y, C - B) e(L, H_2(m_w || m, V, U)Y) \cdot e(U, B) \]

Figure 4: Anonymous proxy signature generation phase and the verification phase of our scheme.
\[
\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n, m, m_w, c, A, B, C, D, L, U, V) \]

as the anonymous proxy signature and sends \(\sigma\) to the verifier.

(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 \(e(D, \sum_{i=1}^{n} \sigma_i) \cdot e(A, Y) = e(cV, C-B) \cdot e(L, H_2(m_w \| m, V, U) Y) \cdot e(U, B)\) holds. If it holds, the verifier accepts the signature, otherwise rejects it.

4.2. Correctness. In the delegation verification phase, each proxy signer can check whether the equation \(vP = R + H_0(m_w, R) Y_o\) holds as follows.

**Proof (first proof).**

\[
vP = R + H_0(m_w, R) Y_o
\]

\[
vP = (r + x_0 H_0(m_w, R)) P
\]

\[
vP = r P + x_0 H_0(m_w, R) P = R + H_0(m_w, R) Y_o. \qed\]

If it holds, the proxy signer can know that the message is sent from the original signer. Because in the verification equation, he use the original signer’s public key \(Y_o\) to examine it. If any adversary intercepts the message and modify it, it cannot pass the verification equation.

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

**Proof (second proof).**

\[
\left( \prod_{i=1}^{n} e(D, \sigma_i) \right) \cdot e(A, Y) = e(cV, C-B) \cdot e(L, H_2(m_w \| m, V, U) Y) \cdot e(U, B)
\]

\[
\left( \prod_{i=1}^{n} e(D, \sigma_i) \right) \cdot e(A, Y) = \left( \prod_{i=1,i \neq s}^{n} e(cr_i V, \sigma_i) \cdot e(cr_i V, \sigma_i) \cdot e(r_i * c * psk, P, Y) \right)
\]

\[
= \left( \prod_{i=1,i \neq s}^{n} e(cr_i V, \sigma_i) \cdot e(cV, r_i * x_i^{-1} * H_2(m_w \| m, V, U) Y) \cdot e(P, r_i psk, Y) \right)
\]

\[
= \left( \prod_{i=1,i \neq s}^{n} e(cr_i V, \sigma_i) \cdot e(cV, r_i * x_i^{-1} * H_2(m_w \| m, V, U) Y) \cdot e(P, r_i \sigma_i) \right)
\]

\[
= \prod_{i=1,i \neq s}^{n} e(cr_i V, \sigma_i) \cdot e(cV, r_i * x_i^{-1} * H_2(m_w \| m, V, U) Y) \cdot e(P, r_i \sigma_i) \cdot e(cV, C-B) \cdot e(L, H_2(m_w \| m, V, U) Y) \cdot e(U, B)
\]

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) anonymity revocation. Now, we demonstrate why our scheme can satisfy these five security properties as follows.

(1) **Verifiability.** In APS verification phase, after checking and verifying the proxy signature \(\sigma\), where \(\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n, m, m_w, c, A, B, C, D, L, U, V)\), the verifier can calculate to check whether the verification equation \(\left( \prod_{i=1}^{n} e(D, \sigma_i) \right) \cdot e(A, Y) = e(cV, C-B) \cdot e(L, H_2(m_w \| m, V, U) Y) \cdot e(U, B)\) 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 \(V = vP = R + H_0(m_w, R) Y_o\) in the verification equation.

(2) **Unforgeability.** It means that any entity (other than the real proxy signer \(p_s\)), including the original signer, cannot generate a valid proxy signature. Only an authorized proxy signer \(p_s\) can create a valid proxy signature \(\sigma\). 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 \(psk\), to compute \(\sigma\). However, this is impossible since the identity of the attacker was not in \(m_w\) signed by the original signer. Not to mention, he does not know \(psk\). Under this situation, even if he want to (1) fake the proxy signer key as \(psk'\), (2) change value \(c\) to \(c'\), or (3) randomly select \(r_i^* \in Z_q^*\), trying to counterfeit the proxy
signature, we demonstrate the reasons that his attempt deems 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 $psk_i$, he cannot generate valid $\sigma_i(= psk_i \ast Y)$, $p\sigma$ sum ($= \sum_{i=1}^{n} a_i)$, $A(= r_i \ast c \ast psk_i P), B(= r_i \sigma_i), \text{and } C(= r_i \ast p\sigma \text{ sum})$. Even if he uses a random $psk'_i$ to sign the message, since $psk'_i = r_i^{-1} \ast x_i^{-1} \ast H_2(m_w || m, V, U)$, he cannot evaluate the right value $x_i^{-1}$ for computing $L$ to be successfully verified in the verification equation.

Case 2. Because $c$ is changed to $c'$, this results in at least one of the random numbers $r_i$ should also be modified. Without loss of generality, we let $r_i = r_i \neq r_i$. Accordingly, all the parameters $U(= c P), psk_i(= r_i^{-1} \ast x_i^{-1} \ast H_2(m_w || m, V, U)), \sigma_i(= psk_i \ast Y), p\sigma$ sum ($= \sum_{i=1}^{n} a_i), A(= r_i \ast c \ast psk_i P), B(= r_i \sigma_i), C(= r_i \ast p\sigma \text{ sum}), D(= r_i \ast c \ast V),$ and $L(= c \ast x_i^{-1} \ast V)$ are changed as well. That is $\sigma' = (\sigma_1', \sigma_2', ..., \sigma_n'), \sigma_i, m, m_w, c', A', B', C', D', L', U', V)$. Apparently, the verification equation $(\prod_{i=1}^{n} e(D, \sigma_i)) \ast e(A, Y) = e(cV, C - B) \ast e(L, H_2(m_w || m, V, U)) \ast e(U, B)$ cannot hold. Below, we only show the inequality of portion of the verification equation $e(A', Y) = e(U', B')$:

$$e(A', Y) = e(r_i' \ast c' \ast psk'_i P, Y)$$
$$= e(c' P, r_i' psk'_i Y)$$
$$= e(c' P, r_i' \sigma_i)$$
$$\neq e(U, B).$$

Case 3. In this case, if any attacker randomly selects $r_i' \in Z_q^*$, trying to generate the valid proxy signature $\sigma'$. Accordingly, the parameters $U(= c P), psk_i(= r_i^{-1} \ast x_i^{-1} \ast H_2(m_w || m, V, U)), \sigma_i(= r_i^{-1} \ast x_i^{-1} \ast H_2(m_w || m, V, U) \ast Y), p\sigma$ sum ($= \sum_{i=1}^{n} a_i), A(= r_i \ast c \ast psk_i P), B(= r_i \sigma_i), C(= r_i \ast p\sigma \text{ sum}), D(= r_i \ast c \ast V),$ and $L(= c \ast x_i^{-1} \ast V)$ are all changed. Therefore, the signature now becomes $\sigma' = (\sigma_1', \sigma_2', ..., \sigma_n', m, m_w, c', A', B', C', D', L', U', V)$. As in Case 1, the verifier checks whether $e(A', Y) = e(U, B')$ holds or not. Apparently, it cannot pass the verification.

(3) Undeniability. As in Section 4.2 proof (second proof), the verifier uses the verification equation: $(\prod_{i=1}^{n} e(D, \sigma_i)) \ast e(A, Y) = e(cV, C - B) \ast e(L, H_2(m_w || m, V, U)) \ast e(U, B)$ to check whether the proxy signature comes from one of the members in the proxy signer group. Since the equation $V = vP = R + H_0(m_w, R)Y_0$ includes the original signer’s public key $Y_0$ 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 generation.

(4) Anonymity. In the APS generation phase, all the parameters $A, B, C, D, L$ have to be multiplied by $r_i \in Z_q^*$ to make the proxy signature $\sigma$ anonymous. If any attacker wants to know who is the real proxy signer, he must know the value $r_i$ to use $r_i^{-1}$ for unrandomizing all parameters to get $A'(= c' \ast psk'_i P), B(= \sigma'_i), C'(= p\sigma \text{ sum'}), D'(= c' \ast V)$, and $\sigma'_i(= x_i^{-1} \ast H_2(m_w || m, V, U))$. But now $\sigma_i = r_i V, i \neq s$, even the attacker knows $r_i$ without the knowledge of $r_i$ and $x_i$, he cannot know who the real signer is. Not to mention, he cannot know the value of $r_i$. It means that anyone cannot know who signs the signature. Hence, the anonymity holds.

(5) Anonymity Revocation. In our scheme, only the proxy signer knows $r_i^{-1}$ and the secret $x_i^{-1}$. He can convince the others that he is the real proxy signer by just showing them $r_i^{-1}$ and the holdness of the equation $r_i \ast x_i \ast \sigma_i = H_2(m_w || m, V, U) \ast Y$ without revealing $x_i$ in polynomial time.

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 2. From Table 2, we denote by $e$ the pairing operation, $P_m$ and $Pa$ the point multiplication and point addition on $G_1$ respectively, and by $n$ the number of proxy signers. In Yu et al.’s APS scheme, the generation and verification of $psk$ should be $(2n + 1)Pm + nPa$ 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 $(n + 1)e + nPm + 2nPa$ rather than the original $(n + 1)e + nPm + (n + 1)Pa$ as listed in the table of [8]. From Table 2, we can see that our scheme is more efficient than Yu et al.’s protocol.

6. Conclusions

In 2009, Yu et al. first proposed a one-to-many APS scheme attempting to protect the proxy signer’s privacy while maintaining secrecy to outsiders. However, after analyses, we determined that Yu et al.’s original protocol could not satisfy the anonymous property. Accordingly, we proposed a novel one-to-many APS scheme to reach the goal. Our construction makes use of a random number $r$, one-way hash function and bilinear pairings to make the proxy signature anonymous. After comparisons, we conclude that
our new protocol is a significant improvement against attackers trying to reveal the identity of the real signer and is more efficient in computational cost as demonstrated in Table 2.

References


Submit your manuscripts at http://www.hindawi.com