Remarks on the security of the strong proxy signature scheme with proxy signer privacy protection

Amit K. Awasthi

Department of Mathematics
Pranveer Singh Institute of Technology
Bhuati, Kanpur-208020, UP, India
E-mail: awasthi@psit.in

Abstract: In 1996, Mambo et al. introduced the proxy signature scheme for digital applications to delegate the signing capability to a proxy signer. Various constructions were made to devise a strong nondesignated proxy signature scheme. In 2002, Shum and Wei proposed an extended scheme to hide the identity of the proxy signer. A Trusted Authority (TA) can reveal the proxy signer’s identity if required. In this paper we show some possible attacks on this scheme.

Keywords: proxy signature; warrant; cryptography; digital signatures.


Biographical notes: Amit K. Awasthi received his PhD in Mathematics (Cryptography) from the Dr. BRA University of Agra, India. He is an Assistant Professor at the Department of Mathematics, PSIT Kanpur, UP, India. His research interests include issues related to conceptual frameworks of authentication, smartcards and digital signatures.

1 Introduction

A great achievement of modern cryptography is the invention of digital signatures which function similarly to handwritten signatures. Digital signatures should reflect the identity of the signer and the recipient must be able to identify the signer and verify the signature. Digital signatures have very important applications in the digital world.

Various kinds of delegation are very common in society. Delegation of signing power is one of them. Digital signature schemes rely on a secret signing key which is known only to the signer. One solution to the delegation of digital signatures is to give this secret key to the person designated as a proxy. But in this case the secret can no longer be identified with that original signer. This violates the basic assumption of the digital signature. To overcome this problem the concept of proxy signatures was first proposed by Mambo et al. (1996). A proxy signature scheme allows a signer to delegate the signing capability to a designated person, called a proxy signer.
Mambo *et al.* (1996) developed a systematic approach to a proxy signature. After introducing the proxy problem, they first divided the delegation into three kinds, which was further extended by Kim *et al.* (1997) by adding one more kind – partial delegation with warrant. We now have the following types of delegation in the literature:

1. **Full delegation** – This happens when the original signer Alice gives her private key $x_{Alice}$ to the proxy signer Bob. In this case a proxy signature by Bob is indistinguishable from the original signature of Alice.

2. **Partial delegation** – This happens when the original signer Alice computes a proxy key $s$ from her private key $x_{Alice}$ and gives it to the proxy signer in a secure way. In this case the proxy signature by Bob is distinguishable from the original signature created by Alice.

3. **Delegation by warrant** – A warrant is a certificate composed of a message part that the proxy signer is authorised to sign and a public key which ensures the involvement of the original signer.

4. **Partial delegation with warrant** – In this delegation a proxy key is computed by the original signer Alice from her private key $x_{Alice}$ and a warrant message issued by her. The proxy signature scheme of Kim *et al.* (1997) included a warrant in their proxy signature key.

Lee *et al.* (2001) proposed a strong nondesignated proxy signature scheme. In the nondesignated proxy signature scheme, the original signer does not specify his/her proxy signer in the proxy key issuing phase. Anyone who owns the warrant and some secret parameters issued by the original signer can construct his/her proxy signing key. Then, he/she can use it to sign messages on behalf of the original signer. Still, the following security measures were taken:

- **Verifiability**: A verifier should be convinced of the original signer’s agreement on the message signed with a proxy signature.

- **Strong unforgeability**: Only the designated proxy signer can generate a valid proxy signature.

- **Strong nonrepudiation**: A proxy signer cannot deny a valid proxy signature that he/she generated.

- **Nondesignated**: The warrant issued by the original signer does not specify who the proxy signer is. It is also transferable among proxy signers.

- **Strong identifiability**: From a proxy signature, any verifier can determine the identity of the proxy signer.

- **Proxy privacy**: No one can determine the identity of the proxy signer only from the proxy signature.

- **Privacy revocation**: Once needed, a Trusted Authority (TA) can reveal the proxy signer’s identity.

Shum and Wei (2002) further extended Lee *et al.*’s scheme, while maintaining all the above properties except the strong identifiability.
In this paper, we found some possible attacks on the Shum-Wei scheme. First we show that the original signer can delegate another proxy signer, who may generate a proxy signature for any other identity. In the second attack we will show that the TA, if corrupted, may produce a key for proxy signing. Using this key, a valid proxy signature could be generated for any original signer. This intruder proxy can also change the warrant information.

2 Review of the Shum-Wei scheme

In Shum and Wei (2002), there is a trusted alias issuing authority $T$. This authority $T$ is responsible for issuing an alias for every proxy signer. $O$ and $P$ denote an original signer and a proxy signer, respectively. $V$ denotes a verifier. Some parameters used in this paper are as follows:

- $p, q$ large prime numbers, where $q|(p – 1)$
- $g$ an element of order $q$ in $\mathbb{Z}_q$
- $h(.)$ a one-way hash function
- $x_i$ the secret key of user $i$
- $y_i = g^{x_i} \mod p$ the public key of user $i$
- $m$ the signing message
- $m_w$ the warrant issued by $O$
- $\sigma = \text{Sign}(m, x)$ a discrete-logarithm-based signature algorithm by using a secret key $x$ to sign a message $m$ and $\sigma$ is the digital signature of $m$
- $\text{verify}(m, \sigma, y)$ the corresponding signature verification algorithm by using a public key $y$ to verify the validation of the signature $\sigma$.

2.1 Issuing alias phase

$T$ issues an alias $h_P$, a public parameter $r_T$ and a secret key $s_T$ to $P$ and records the triple $(h_P, k_P, ID_P)$ into the database, where $ID_P$ is the identity of $P$. $P$ will check the validation of the secret key $s_T$. If it holds, $P$ will use $s_T$ in the proxy signature generation phase. Once needed, $T$ can reveal the $ID_P$ to revoke the proxy privacy according to $h_P$.

Figure 1 Issuing protocol

$$
\begin{align*}
\text{P} & \quad \text{T} \\
\xrightarrow{ID_P} & \\
& k_P \in R \quad Z \\
& h_P = h(k_P, ID_P) \\
& k_T \in R \quad \mathbb{Z}_q^*, \quad r_T = g^{k_T} \quad \text{mod } p \\
& g^{s_T} = y_T^{h(h_P, r_T)} r_T \quad \text{mod } p \\
& h_P, r_T, s_T, x_T, h(h_P, r_T) + k_T \quad \text{mod } q
\end{align*}
$$
2.2 Delegation phase

M generates a secret $s_M$ and sends it to P along with the warrant $m_w$ and a public parameter $r_M$. P will check the correction of the $s_M$. If it holds, P will combine it with $s_T$ to sign messages.

**Figure 2** Delegation protocol

\[
\begin{align*}
M & \quad P \\
\text{let} & \\
\begin{array}{c}
M \\
\text{computes} \\
\end{array} & \\
\begin{array}{c}
k_M \in \mathbb{Z}_q^* \\
r_M = g^{k_M} \mod p \\
S_M = x_M h(m_w, r_M) + k_M \mod q \\
g^{S_M} = y_M^{(m_w r_M)} r_M \mod p
\end{array}
\]

2.3 Sign and verify phase

P computes the proxy signing key $x$ as $x = s_M + s_T \mod q$. Then, P can use x to sign messages. The proxy signature of m is $(\sigma, r_M, ID_M, h_P, r_T)$. If V wants to verify the correction of the proxy signature, He/she first computes the public verification key y, where $y = y_M^{(m_w r_M)} r_M \cdot y_T^{(h_P r_T)} r_T$. Then V checks the validation of $\text{verify}(m, \sigma, y)$. If it holds, the proxy signature is valid.

**Figure 3** Signing protocol

\[
\begin{align*}
M & \quad P \\
\text{computes} & \\
\begin{array}{c}
x = S_M + S_T \mod q \\
\sigma = \text{sign}(m, x) \\
\end{array} & \\
\begin{array}{c}
y_1 = y_M^{(m_w r_M)} r_M \mod p \\
y_2 = y_T^{(h_P r_T)} r_T \mod p \\
y = y_1 y_2 \mod p \\
\end{array} & \\
\text{verify}(m, \sigma, y)
\end{align*}
\]

2.4 Revoking privacy phase

Once needed, V sends $h_P$ to T. T will reveal ID_P to revoke P’s privacy.

By using the certified alias, the Shum-Wei scheme can hide the identity of the proxy signer. Once needed, privacy can be revoked by T in the revoking privacy phase. In the delegation phase, $(m_w, r_M, s_M)$ is transferable among the proxy signers. Each proxy signer can sign messages by combining $s_M$ and his/her secret key $s_T$ as a proxy signing key $x$. Thus, the non-designated property is achieved.
3 Security flaws in the Shum-Wei scheme

In this section we show some security flaws in the Shum-Wei scheme.

3.1 Attack 1: The original signer repudiates the proxy user with the key alias $h_p$

The original signer $M$ computes a commitment $r_T = y_T^e \mod p$ for trusted alias issuing authority $T$, where $y_T$ is the public key of $T$. Now $M$ chooses the user whom he/she wants to repudiate. Let this user have the alias $h_p$. $M$ computes $r_M = y_T^{\gamma(h_p, h_T)}$. Then $M$ computes the proxy secret $x = x_M h(m_w, r_M)$. With this key he/she may produce a valid signature tuple $(\sigma, r_M, m_w, ID_M, h_p, r_T)$.

Any verifier may verify that $(\sigma, r_M, m_w, ID_M, h_p, r_T)$ is a valid proxy signature, by computing:

$$\frac{y_{h(p, h_p, r_T)}}{y_T} \cdot y_T^{h(p, h_p, r_T)} \cdot r_T = \frac{y_{h(p, h_p, r_T)}}{y_T} \cdot y_T^{h(p, h_p, r_T)} \cdot y_T^e = \frac{y_{h(p, h_p, r_T)}}{y_T} \cdot y_T^e = g^{h(p, h_p, r_T)} = y.$$

As $y$ is a public proxy key, this will convince the verifier that signature $\sigma$ is a valid proxy signature.

3.2 Attack 2: The original signer with a proxy user $P'$ repudiates the proxy user $P$ with key alias $h_p$

The original signer $M$ computes a commitment $r_T = y_T^e \mod p$ for trusted alias issuing authority $T$, where $y_T$ is the public key of $T$. Now $M$ chooses the user whom he/she wants to repudiate. Let this user have the alias $h_p$. $M$ computes $r_M = y_T^{\gamma(h_p, h_T)}$. Then $M$ computes the proxy secret $x = x_M h(m_w, r_M) + b$. He/She sends $x$, $r_M$, $r_T$, $h_p$ to a new proxy signer $P'$. Now $P'$ may use this information to sign some messages on behalf of the original signer. He/she may produce a valid signature tuple $(\sigma, r_M, m_w, ID_M, h_p, r_T)$. But his/her identity $ID_P$ is completely hidden. He/She repudiates the identity $ID_P$. During revocation, the verifier may be convinced that the signature is produced by proxy $ID_P$.

Any verifier may verify that $(\sigma, r_M, m_w, ID_M, h_p, r_T)$ is a valid proxy signature by computing:
Remarks on the security of the strong proxy signature scheme

\[ y_M^{\lambda(\gamma_m, \gamma) \cdot r_T \cdot y_T^{\lambda(b(\gamma_m, \gamma) \cdot x)} \cdot r_T} = y_M^{\lambda(\gamma_m, \gamma) \cdot (y_T^{\lambda(b(\gamma_m, \gamma) \cdot x)} \cdot y_T^{\lambda(b(\gamma_m, \gamma) \cdot x)}) \cdot y_T} \\
= y_M^{\lambda(\gamma_m, \gamma) \cdot y_T^{\lambda(b(\gamma_m, \gamma) \cdot x) \cdot y_T}} \\
= g^{y_T^{\lambda(b(\gamma_m, \gamma) \cdot x) \cdot y_T}} \\
= g^y = y. \]

As \( y \) is a public proxy key, it will convince the verifier that signature \( \sigma \) is a valid proxy signature. Now if the verifier wants to verify the proxy signer for \( h_p \), he/she may communicate with the alias issuing authority, who will return a corresponding identity \( ID_p \). Thus the new proxy user (with the help of the original signer) will repudiate the previously registered proxy signers.

4 Conclusion

In this paper, we showed two attacks on the Shum-Wei scheme. We showed that a malicious original signer is able to generate a valid proxy signature by himself/herself repudiating any proxy signer enrolled with the alias issuing authority. In the second attack we showed that at the verifier end, the proxy revocation protocol may produce the wrong identity.

References