# On the Security of Some Proxy Signature Schemes

Hung-Min Sun and Bin-Tsan Hsieh[†]
Department of Computer Science,
National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
hmsun@cs.nthu.edu.tw
[†]Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
bintsan@csie.ncku.edu.tw

**Abstract**

Digital signature scheme is an important research topic in cryptography. An ordinary digital signature scheme allows a signer to create signatures of documents and the generated signatures can be verified by any person. A proxy signature scheme, a variation of ordinary digital signature scheme, enables a proxy signer to sign messages on behalf of the original signer. To be used in different applications, many proxy signatures were proposed. In this paper, we review Lee *et al.*'s strong proxy signature scheme, multi-proxy signature scheme, and its application to a secure mobile agent, Shum and Wei's privacy protected strong proxy signature scheme, and Park and Lee's nominative proxy signature scheme, and show that all these proxy signature schemes are insecure against the original signer's forgery. In other words, these schemes do not possess the unforgeability property which is a desired security requirement for a proxy signature scheme.

**Keywords**: Cryptanalysis, Proxy Signature, Strong Proxy Signature, Nominative Signature

## 1 Introduction

Digital signature scheme [1][2][3][4] is an important research topic in cryptography. An ordinary digital signature scheme allows a signer to create signatures of documents and the generated signatures can be verified by any person. Due to its importance, many variations of digital signature scheme were proposed, such as blind signature scheme [10], undeniable signature scheme [11][12], ... etc. which can be used in different application situations.

A proxy signature scheme [5][6][7][8][9], a variation of ordinary digital signature scheme, enables a proxy signer to sign messages on behalf of the original signer. Proxy signature schemes have been shown to be useful in many applications. For example, a manager can delegate his secretaries to sign documents while he is on vacation. Proxy signature schemes can also be used in electronics transaction [13] and mobile agent environment [14][15][16].

To categorize delegation types, Mambo *et al.* [5] defined three levels of delegation: full delegation, partial delegation, and delegation by warrant. In full delegation, the original signer gives his secret key to the proxy signer. The proxy signer uses the key to sign documents. In partial delegation, the proxy signature signing key is generated by both the original signer and proxy signer. In delegation by warrant, the original signer signs the warrant which describes the relative rights and information of the original signer and proxy signer such that a signature verifier can use the warrant as a part of verification information. In [6], Kim *et al.* proposed a partial delegation with warrant proxy signature scheme which enjoys the computational advantage over the proxy signature by warrant and the structure advantage over the proxy signature for partial delegation. No matter what kind of proxy signature schemes, a proxy signature scheme should provide good security properties.

In 1999, Sun and Hsieh [17] pointed out that the Mambo *et al.*'s scheme [5] is unfair to the original signer because the proxy signer can transfer the signing right to others, and that the Kim *et al.*'s scheme [6] is insecure to the public key substitution attack in which an attacker can create a valid proxy signature by updating his own public key with other's public key. To overcome these security flaws, Sun and Hsieh also proposed two modified schemes [17]. Later, Sun and Hseih's modified schemes are found that they still suffer from the public key substitution attack and a kind of forgery attack. Consequently, they are further improved in [18][19]. Here we note that in order to prevent the public key substitution attack, we can make the certificate authority, CA, to check the corresponding secret key while updating user's public key [18]. Hence, we will not take the public key substitution attack into consideration in this paper since the CA can afford the work of prevention.

Recently, Lee *et al.* [15] defined properties that a strong proxy signature scheme should provide:

**Strong Unforgeability:** Only the legitimate proxy signer can generate a valid proxy signature; even the original signer can not.

**Verifiability:** Anyone can verify the signature and the signed message should conform to the delegation warrant.

**Strong Identifiability:** Anyone can determine the identity of the corresponding proxy signer.

**Strong Undeniability:** The proxy signer can not repudiate the signature which he ever generated.

**Prevention of Misuse:** The proxy key pair should be used in any place conforms to the warrant.

As they mentioned in [15], the Mambo *et al.*'s scheme [5] and the Kim *et al.*'s scheme [6] do not satisfy the prevention of misuse property. Therefore, following the above definition, Lee *et al.* [15] proposed a strong proxy signature scheme which we call it LKK-SPS scheme for short hereafter. They also applied the LKK-SPS scheme to design a multi-proxy signature scheme [15] in which multiple original signers can delegate their signing rights to one proxy signer. Based on the LKK-SPS scheme, they further proposed an application of strong proxy signature to secure mobile agent [20]. Later, based on the LKK-SPS scheme, Shum and Wei [21] also proposed a privacy-protected strong proxy signature scheme which is an enhanced version to protect the proxy signer's identity behind an alias.

On the other hand, Kim *et al.* [22] proposed a nominative signature scheme which is also a variation of digital signature scheme. A nominative signature scheme includes two parties: a nominator who signs a digital signature and a nominee who is able to verify the validity of the signature. A nominative signature should achieve two requirements:

(1) only the nominee can verify the nominator's signature.

(2) only the nominee can prove to the third party that the signature is issued to him and is valid.

Based on Kim *et al.*'s nominative signature scheme, Park and Lee [14] proposed a digital nominative proxy signature scheme for mobile communication which is a combination of nominative signature scheme and proxy signature scheme. Therefore, the scheme should simultaneously provide the security requirements of both the proxy signature scheme and nominative signature scheme.

In this paper, we show that Lee *et al.*'s strong proxy signature scheme, multi-proxy signature scheme and its application to a secure mobile agent, Shum and Wei's privacy-protected strong proxy signature scheme, and Park and Lee's nominative proxy signature scheme are all insecure against the original signer's forgery. In other words, these schemes do not possess the unforgeability property which is a desired security requirement for a proxy signature scheme.

## 2 Notations

In this section, we give the notations for reviewing the proxy signature schemes through this paper.

| | |
|---|---|
| $A$ | the nominator |
| $B$ | the nominee |
| $T$ | the trust third party |
| $O$ | the original signer |
| $P$ | the proxy signer |
| $V$ | the signature verifier |
| $p, q$ | large prime number with $q\|(p-1)$ |
| $g$ | a element of order $q$ in $Z_p^*$ |
| $h()$ | a secure one-way hash function |
| $m_w$ | a warrant |
| $x_u$ | the secret key of user $u$ |
| $y_u$ | the public key of user $u$, $y_u = g^{x_u} \bmod p$ |
| $A \rightarrow B$ | $A$ sends message to $B$ |

| | |
|---|---|
| $sign()$ | signing algorithm |
| $verify()$ | verification algorithm |

# 3 On the Security of the LKK-SPS Scheme and Its Applications

## 3.1 The LKK-SPS Scheme

We first briefly review the LKK-SPS scheme as follows.

### 3.1.1 Proxy Delegation

In the proxy delegation phase, the original signer $O$ needs to generate a triplet $(m_w, r_o, s_o)$ and sends the triplet to the proxy signer $P$.

First, $O$ selects $k_o \in_R Z_q^*$, and computes $r_o = g^{k_o} \bmod p$ and $s_o = x_o h(m_w, r_o) + k_o \bmod q$. Next, he sends $(m_w, r_o, s_o)$ to $P$. In fact, $(r_o, s_o)$ is the signature of $m_w$ using Schnorr's signature scheme.

$P$ accepts $(m_w, r_o, s_o)$ if the equation $g^{s_o} = y_o^{h(m_w, r_o)} r_o \bmod p$ holds. The proxy delegation scenario is depicted as Figure 1.

| | |
|---|---|
| $O$ computes: | $k_o \in_R Z_q^*, r_o = g^{k_o} \bmod p$ |
| | $s_o = x_o h(m_w, r_o) + k_o \bmod q$ |
| $O \rightarrow P$ | $(m_w, r_o, s_o)$ |
| $P$ checks: | $g^{s_o} \stackrel{?}{=} y_o^{h(m_w, r_o)} r_o \bmod p$ |

Figure 1: LKK-SPS Proxy Delegation

### 3.1.2 Signing and Verification

The proxy signer $P$ computes the proxy signature signing key $x_{pr} = s_o + x_p \bmod q$ and uses it to generate the proxy signature $\sigma$ of a message $m$ by using a conventional DLP-like signature scheme. The proxy signer then sends $(m, \sigma, r_o, m_w)$ to the verifier $V$. $V$ computes the corresponding proxy signature public key $y_{pr} = y_o^{h(m_w, r_o)} r_o y_p \bmod p$ and uses it to verify the proxy signature through the signature verification phase of the used DLP-like signature scheme. The scenario of signing and verification is depicted as Figure 2.

| | |
|---|---|
| $P$ computes: | $x_{pr} = s_o + x_p \bmod q$ |
| | $\sigma = sign(m, x_{pr})$ |
| $P \rightarrow V$ | $(m, \sigma, r_o, m_w)$ |
| $V$ checks: | $y_{pr} = y_o^{h(m_w, r_o)} r_o y_p \bmod p$ |
| | $verify(m, \sigma, y_{pr}) \stackrel{?}{=} true$ |

Figure 2: LKK-SPS Signing and Verification

## 3.2 Multi-Proxy Signature Scheme Based on the LKK-SPS

Based on the LKK-SPS scheme, Lee *et al.* proposed a multi-proxy signature scheme in which multiple original signers can delegate their signing rights to one proxy signer. Let $O_i, (i = 1$ to $n)$ denote the group of $n$ original signers and $(x_i, y_i)$ denote the public key and secret key pair of an original signer $O_i$.

### 3.2.1 Proxy Delegation

Each $O_i$ selects $k_i \in_R Z_q^*$ and computes $r_i = g^{k_i} \bmod p$ and $s_i = x_i h(m_{w_i}, r_i) + k_i \bmod q$. Next, he sends $(m_{w_i}, r_i, s_i)$ to $P$. $P$ accepts $(m_{w_i}, r_i, s_i)$ if $g^{s_i} = y_i^{h(m_{w_i}, r_i)} r_i \bmod p$.

$$
\begin{array}{ll}
O_i \text{ computes:} & k_i \in_R Z_q^*, r_i = g^{k_i} \bmod p \\
 & s_i = x_i h(m_{w_i}, r_i) + k_i \bmod q \\
O_i \rightarrow P & (m_{w_i}, r_i, s_i) \\
P \text{ checks:} & g^{s_i} \stackrel{?}{=} y_i^{h(w_i, r_i)} r_i \bmod p
\end{array}
$$

Figure 3: Multi-Proxy Signature Proxy Delegation

### 3.2.2 Signing and Verification

If $P$ wants to create a proxy signature, he first computes the proxy signature signing key $x_{pr} = s_1 + ... + s_n + x_p \bmod q$. Next, he uses $x_{pr}$ to generate the proxy signature $\sigma = sign(m, x_{pr})$ of a message $m$. Anyone who wants to verify the signature computes the proxy signature public key $y_{pr} = y_{o_1}^{h(m_{w_1}, r_1)} r_1 ... y_{o_n}^{h(m_{w_n}, r_n)} r_n y_p \bmod p$ and checks whether $verify(m, \sigma, y_P) = true$.

$$
\begin{array}{ll}
P \text{ computes:} & x_{pr} = s_1 + ... + s_n + x_p \bmod q \\
 & \sigma = sign(m, x_{pr}) \\
P \rightarrow V & (m, \sigma, r_1 ... r_n, m_{w_1} ... m_{w_n}) \\
V \text{ checks:} & y_{pr} = y_{o_1}^{h(m_{w_1}, r_1)} r_1 ... y_{o_n}^{h(m_{w_n}, r_n)} r_n y_p \bmod p \\
 & verify(m, \sigma, y_{pr}) \stackrel{?}{=} true
\end{array}
$$

Figure 4: Multi-Proxy Signature Signing and Verification

## 3.3 Cryptanalysis of the LKK-SPS Scheme and Its Application

In this subsection, we show that both the LKK-SPS scheme, multi-proxy signature scheme, and its application to secure mobile agent are insecure against the original signer's forgery. In the LKK-SPS scheme, in order to forge a proxy signature, a dishonest original signer computes $r'_o = y_p^{-1} \bmod p$.[1] Thus, $x_p r' = x_o h(m_w, r'_o) \bmod q$ is a valid proxy signature signing key and $(m, \sigma, r'_o)$ is a valid proxy signature. This is because:

$$\begin{aligned} y_{pr} &= y_o^{h(m_w, r'_o)} * r'_o * y_p \bmod p \\ &= g^{x_o h(m_w, r'_o)} * y_p^{-1} * y_p \bmod p \\ &= g^{x_o h(m_w, r'_o)} \bmod p \\ &= g^{x'_{pr}} \bmod p \end{aligned}$$

In the multi-proxy signature scheme, in order to forge a multi-proxy signature, the original signer $O_1$computes $r'_1 = (y_{o_2}^{h(m_{w_2}, r_2)} r_2 ... y_{o_n}^{h(m_{w_n}, r_n)} r_n y_p)^{-1} \bmod p$. Thus, $x'_{pr} = x_o h(m_w, r'_1) \bmod q$ is a valid proxy signature signing key and $(m, \sigma, r'_o)$ is a valid proxy signature. This is because:

$$\begin{aligned} y_{pr} &= y_o^{h(m_w, r'_1)} * r'_1 * (y_{o_2}^{h(m_{w_2}, r_2)} r_2 ... y_{o_n}^{h(m_{w_n}, r_n)} r_n y_p) \bmod p \\ &= g^{x_o h(m_w, r'_1)} * (y_{o_2}^{h(m_{w_2}, r_2)} r_2 ... y_{o_n}^{h(m_{w_n}, r_n)} r_n y_p)^{-1} \\ &* (y_{o_2}^{h(m_{w_2}, r_2)} r_2 ... y_{o_n}^{h(m_{w_n}, r_n)} r_n y_p) \bmod p \\ &= g^{x_o h(m_w, r'_1)} \bmod p \\ &= g^{x'_{pr}} \bmod p \end{aligned}$$

Here we also note that in [20], Lee *et al.* applied the LKK-SPS scheme to design a secure mobile agent. The above attack can work successfully on the proposed secure mobile agent directly.

# 4 On the Security of Shum and Wei's Scheme

## 4.1 Shum and Wei's scheme

Based on the LKK-SPS scheme, Shum and Wei also proposed a privacy-protected proxy signature scheme which protects the proxy signer's identity behind an alias. Their proposed scheme contains the alias issuing , the proxy delegation, the signing and verification, and the privacy revoking phases. The details of each phases are showed as follows.

### 4.1.1 Alias Issuing

In this phase, $T$ issues an alias $h_P$ to $P$. The alias issuing protocol runs as follows.

[1]The general form of $r'_o$ is $y_p^{-1} g^a \bmod p$, where $a$ is a random number. We remain $r'_o = y_p^{-1} \bmod p$ in the article to make reader understanding clearly. We omit the description in other cryptanalysis since they are similar.

step 1. $P$ sends his identity $ID_P$ to $T$.

step 2. $T$ selects $k_P \in_R Z$ and $k_T \in_R Z_q^*$. He computes $h_P = h(k_P, ID_P)$, $r_T = g^{k_T} \bmod p$, and $s_T = x_T h(h_P, r_T) + k_T \bmod q$. Next, he sends $(h_P, r_T, s_T)$ to $P$.

step 3. $P$ accepts the triplet $(h_P, r_T, s_T)$ if the equation $g^{s_T} = y_T^{h(h_P, r_T)} r_T \bmod p$ holds. We depict the protocol as Figure 5.

$$
\begin{array}{ll}
P \rightarrow T & ID_P \\
T \text{ computes:} & k_P \in_R Z,\ k_T \in_R Z_q^* \\
 & h_P = h(k_P, ID_P) \\
 & r_T = g^{k_T} \bmod p \\
 & s_T = x_T h(h_P, r_T) + k_T \bmod q \\
T \rightarrow P & (h_P, r_T, s_T) \\
P \text{ checks:} & g^{s_T} \stackrel{?}{=} y_T^{h(h_P, r_T)} r_T \bmod p
\end{array}
$$

Figure 5: Shum and Wei's Alias Issuing Protocol

### 4.1.2 Proxy Delegation

In the proxy delegation phase, $O$ generates the proxy delegation $(m_w, r_o, s_o)$ to the proxy signer $P$.

First, $O$ selects $k_o \in_R Z_q^*$, and computes $r_o = g^{k_o} \bmod p$ and $s_o = x_o h(m_w, r_o) + k_o \bmod q$. Next, he sends $(m_w, r_o, s_o)$ to $P$.

$P$ accepts $(m_w, r_o, s_o)$ if the equation $g^{s_o} = y_o^{h(m_w, r_o)} r_o \bmod p$ holds. The proxy delegation protocol is depicted as Figure 6.

$$
\begin{array}{ll}
O \text{ computes:} & k_o \in_R Z_q^*, r_o = g^{k_o} \bmod p \\
 & s_o = x_o h(m_w, r_o) + k_o \bmod q \\
O \rightarrow P & (m_w, r_o, s_o) \\
P \text{ checks:} & g^{s_o} \stackrel{?}{=} y_o^{h(m_w, r_o)} r_o \bmod p
\end{array}
$$

Figure 6: Shum and Wei's Proxy Delegation Protocol

### 4.1.3 Signing and Verification

The proxy signer $P$ computes the proxy signature signing key $x_{pr} = s_o + s_T \bmod q$ and uses it to generate the proxy signature $\sigma$ of a message $m$ by using a conventional DLP-like signature scheme. The proxy signer then sends $(m, \sigma, r_o, m_w, ID_O, h_P, r_T)$ to the verifier $V$. $V$ computes the corresponding proxy signature public key $y = y_o^{h(m_w, r_o)} r_o y_T^{h(h_P, r_T)} r_T \bmod p$ and uses it to check the validity of the proxy signature via the signature verification phase of the used DLP-like signature scheme. The scenario of signing and verification scenario is depicted as Figure 7.

$$
\begin{array}{ll}
P \text{ computes:} & x_{pr} = s_o + s_T \bmod q \\
 & \sigma = sign(m, x_{pr}) \\
P \rightarrow V & m, \sigma, r_o, m_w, ID_O, h_P, r_T \\
V \text{ checks:} & y_{pr} = y_o^{h(m_w, r_o)} r_o y_T^{h(h_P, r_T)} r_T \bmod p \\
 & verify(m, \sigma, y_{pr}) \stackrel{?}{=} true
\end{array}
$$

Figure 7: Shum and Wei's Signing and Verification

#### 4.1.4 Privacy Revoking

If the verifier $V$ wants to know the identity of the signer, he sends the alias $h_P$ to $T$. Next, $T$ returns $k_P$ and $ID_P$ to $V$. Finally, $V$ will be convinced that the signer's identity is $ID_P$ if $h_P = h(k_P, ID_P)$. The privacy revoking scenario is depicted as Figure 8.

$$
\begin{array}{ll}
V \rightarrow T & h_P \\
T \rightarrow V & k_P, ID_P \\
V \text{ checks:} & h_P \stackrel{?}{=} h(k_P, ID_P)
\end{array}
$$

Figure 8: Shum and Wei's Privacy Revoking

### 4.2 Cryptanalysis of Shum and Wei's scheme

In this subsection, we show that Shum and Wei's proxy signature scheme is insecure against the original signer's forgery. A dishonest original signer chooses $h'_P$ and $r'_T$ and computes $r'_o = (y_T^{h(h'_P, r'_T)} r'_T)^{-1} \bmod p$.[2] Thus, $x_p r' = x_o h(m_w, r'_o)$ is a valid proxy signature signing key and $(m, \sigma, r'_o, m_w, ID_o, h'_P, r'_T)$ is a valid proxy signature. This is because:

$$
\begin{aligned}
y &= y_o^{h(m_w, r'_o)} * r'_o * y_T^{h(h'_P, r'_T)} r'_T \bmod p \\
&= g^{x_o h(m_w, r'_o)} * (y_T^{h(h'_P, r'_T)} r'_T)^{-1} * y_T^{h(h'_P, r'_T)} r'_T \bmod p \\
&= g^{x_o h(m_w, r'_o)} \bmod p \\
&= g^{x'_{pr}} \bmod p
\end{aligned}
$$

## 5 On the Security of Park and Lee's Scheme

Park and Lee's nominative proxy signature scheme is a combination of a nominative signature scheme and a proxy signature scheme. Therefore, before we review their scheme, we first introduce the nominative signature scheme as follows.

[2]The general form of $r'_o$ is $(y_T^{h(h'_P, r'_T)} r'_T)^{-1} g^a \bmod p$, where $a$ is a random number.

## 5.1 The Nominative Signature Scheme

The nominator $A$ selects $r, R \in_R Z_q^*$ and computes $t = g^{R-r} \bmod p$, $T = y_B^R \bmod p$, $e = h(y_B, t, T, m)$, and $z = r - x_a e \bmod q$. The signature of a message $m$ is a quadruplet $(t, T, z, y_B)$. Next, $A$ sends $(m, t, T, z, y_B)$ to the nominee $B$. $B$ computes $e = h(y_B, t, T, m)$ and accepts the signature if the equation $(g^z y_A^e t)^{x_B} = T$ holds. The derivation is showed as follows:

$$\begin{aligned}
&(g^z y_A^e t)^{x_B} \bmod p \\
&= (g^{r-x_a e} g^{x_a e} g^{R-r})^{x_B} \bmod p \\
&= (g^R)^{x_B} \bmod p \\
&= y_B^R \bmod p \\
&= T
\end{aligned}$$

We depict the nominative signature scheme as Figure 9.

| | |
|---|---|
| $A$ computes: | $r, R \in_R Z_q^*, t = g^{R-r} \bmod p$ |
| | $T = y_B^R \bmod p$,$e = h(y_B, t, T, m)$ |
| | $z = r - x_a e \bmod q$ |
| $A \rightarrow B$ | $(m, t, T, z, y_B)$ |
| $B$ checks: | $e = h(y_B, t, T, m)$ |
| | $(g^z y_A^e t)^{x_B} \stackrel{?}{=} T$ |

Figure 9: The Nominative Signature Scheme

## 5.2 Park and Lee's Nominative Proxy Signature Scheme

### 5.2.1 Proxy Delegation

In the proxy delegation phase, $O$ generates the proxy delegation $(T_i, M, s_o, r_o)$ to the proxy signer $P$.

First, $O$ selects $k_o \in_R Z_q^*$, and computes $r_o = g^{k_o} \bmod p$ and $s_o = x_o h(M, T_i) + k_o r_o \bmod q$, where $T_i$ is a time stamp. Next, he sends $(T_i, M, s_o, r_o)$ to $P$.

$P$ accepts $(T_i, M, s_o, r_o)$ if the equation $g^{s_o} = y_o^{h(M,T_i)} r_o^{r_o} \bmod p$ holds. The proxy delegation protocol is depicted as Figure 10.

| | |
|---|---|
| $O$ computes: | $k_o \in_R Z_q^*, r_o = g^{k_o} \bmod p$ |
| | $s_o = x_o h(M, T_i) + k_o r_o \bmod q$ |
| $O \rightarrow P$ | $(T_i, M, s_o, r_o)$ |
| $P$ checks: | $g^{s_o} \stackrel{?}{=} y_o^{h(M,T_i)} r_o^{r_o} \bmod p$ |

Figure 10: Park and Lee's Proxy Delegation Protocol

### 5.2.2 Signing and Verification of the Nominative Proxy Signature

To sign the nominative signature, the nominator $P$ selects $r_1, r_2 \in_R Z_q^*$, and computes $K = g^{r_1 - r_2 x_p} \bmod p$, $D = y_v^{r_1} \bmod p$, $z = (y_v, K, D, M)$, and $sz = (r_2 x_p - r_1 s_o e) \bmod q$. The nominator $P$ sends $(M, T_i, r_o, K, D, r_1, sz)$ to the nominee $V$. To verify the nominative signature, $V$ checks whether the equation holds: $(g^{sz}(y_o^{h(M,T_i)} r_o^{r_o})^{r_1 e} K)^{x_v} \bmod p = D$. We depict the signing and verification scenario as Figure 11.

$$
\begin{array}{ll}
P \text{ computes:} & r_1, r_2 \in_R Z_q^*, \\
 & K = g^{r_1 - r_2 x_p} \bmod p \\
 & D = y_v^{r_1} \bmod p \\
 & z = (y_v, K, D, M), e = h(Z) \\
 & sz = (r_2 x_p - r_1 s_o e) \bmod q \\
P \rightarrow V & (M, T_i, r_o, K, D, r_1, sz) \\
V \text{ checks:} & (g^{sz}(y_o^{h(M,T_i)} r_o^{r_o})^{r_1 e} K)^{x_v} \bmod p \stackrel{?}{=} D
\end{array}
$$

Figure 11: Park and Lee's Signing and Verification

## 5.3 Cryptanalysis of Park and Lee's Nominative Proxy Signature Scheme

In this subsection, we show that Park and Lee's nominative proxy signature scheme is insecure against the original signer's forgery. A dishonest original signer chooses $r_e, r_f \in_R Z_q^*$, and computes $K = g^{r_e - r_f} \bmod p$, $D = y_v^{r_e} \bmod p$, $z = (y_v, K, D, M)$, $e = h(Z)$, and $sz = (r_f - r_e s_o e) \bmod q$. Thus, $(M, T_i, r_o, K, D, r_e, sz)$ is a valid proxy signature. This is because:

$$
\begin{aligned}
&(g^{sz}(y_o^{h(M,T_i)} r_o^{r_o})^{r_e e} K)^{x_v} \bmod p \\
&= (g^{r_f - r_e s_o e} * (g^{x_o h(M,T_i)} * g^{k_o r_o})^{r_e e} * g^{r_e - r_f})^{x_v} \bmod p \\
&= (g^{r_f - r_e s_o e} * g^{s_o r_e e} * g^{r_e - r_f})^{x_v} \bmod p \\
&= (g^{r_e})^{x_v} \bmod p \\
&= y_v^{r_e} \bmod p \\
&= D
\end{aligned}
$$

# 6 Conclusions

As our cryptanalysis, the LKK-SPS scheme is insecure against the original signer's forgery. Other applications based on the LKK-SPS schemes hence suffer from the same weakness. These includes Lee *et al.*'s multi-proxy signature scheme and Lee *et al.*'s secure mobile agent, and Shum and Wei's privacy-protected strong proxy signature scheme. In Park and Lee's nominative proxy signature scheme, the using of proxy signer's secret key while creating a nominative proxy signature does not limit it to that only proxy signer can create the nominative proxy signature. Therefore, the original signer who knows the delegation information, $s_o$, can forge the nominative proxy signature.

## References


[1] T. ElGamal, "Cryptography and logarithms over finite fields", Standford University, CA., UMI Order No. DA 8420519, 119 pages, 1984.

[2] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms", IEEE Tran. Information Theory, Vol. 31, No. 4, pp. 469-472, 1985.

[3] L. Harn, "New digital signature scheme based on discrete logarithm", Electronics Letters, Vol. 30, No. 5, pp. 396-398, 1994.

[4] R. L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining igital signatures and public-key cryptosystems", Communications of the ACM, Vol. 21, No. 2, pp. 120-126, 1978.

[5] M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signature: Delegation of the Power to Sign Messages", IEICE Trans. Fundamentals, E79-A:9, pp. 1338-1353, 1996.

[6] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited", Proc. of ICICS'97, International Conference on Information and Communications Security, LNCS 1334, Springer-Verlag, pp. 223-232, 1997.

[7] H. Petersen and P. Horster, "Self-certified keys - concepts and applications," pp. 102-116, Chapman &Hall, 1997.

[8] M. Mambo, K. Usuda, E. Okamoto, "Proxy signatures for delegating signing operation", Proc. 3rd ACM Conference on Computer and Communications Security, New Dehli, India, ACM Press New York, pp. 48-57, 1996.

[9] K. Zhang, "Threshold proxy signature schemes", 1997 Information Security Workshop, pp. 191-197, 1997.

[10] D. Chaum, "Blind signatures for untraceable payments", Advances in Cryptology: Proceedings of Crypto 82, Plenum Press, pp. 199-203, 1983.

[11] D. Chaum and H. van Antwerpen, "Undeniable signatures", Advances in Cryptology-CRYPTO '89 Proceedings, Springer-Verlag, pp. 212-216, 1990.

[12] D. Chaum, "Zero-knowledge undeniable signatures", Advances in Cryptology-EUROCRYPT '90 Proceedings, Springer-Verlag, pp. 458-464, 1991.

[13] P. Kotzanikolaous, M. Burmcster, and V. Chrisskopoulos, "Secure transactions with mobile agents in hostile environments", Proc. ACISP, LNCS 1841, pp. 289-297, 2000.

[14] H.-U. Park and I.-Y. Lee, "A digital nominative proxy signature scheme for mobile communication", ICICS 2001, LNCS 2229, pp. 451-455, 2001.

[15] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its applications", Proc of SCIS, pp. 603-608, 2001.

[16] T. Sander and C. Tschudin, "Towards mobile cryptography", Tech. Rep. 97-409, Int'l Computer Science Inst., Berkeley, 1997.

[17] H.-M. Sun and B.-T. Hsieh, "Remarks on two nonrepudiable proxy signature schemes", Proceeding of Ninth National Conference on Information Security, pp. 241-246. 1999.

[18] H.-M. Sun, "On proxy (multi-) signature schemes", Proceedings of the 2000 ICS: Workshop on Cryptology and Information Security, pp. 65-72, 2000.

[19] S.-M. Yen, C.-P. Hung, and Y.-Y. Lee, "Remarks on some proxy signature schemes", Proceedings of the 2000 ICS: Workshop on Cryptology and Information Security, pp. 54-59, 2000.

[20] B. Lee, H. Kim and K. Kim, "Secure mobile agent using strong nondesignated proxy signature", Proc. of ACISP, LNCS 2119, Springer-Verlag, pp. 474-486, 2001.

[21] K. Shum and Victor K. Wei, "A strong proxy signature scheme with proxy signer privacy protection", Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'02), pp. 55-56, 2002.

[22] S. J. Kim, S. J. Park and D. H. Won, "Nominative signatures", Proc. ICEIC'95, pp. II-68-II-71, 1995.