Parallel Processing Letters


# A BATCH VERIFICATION FOR MULTIPLE PROXY SIGNATURE

Shiang-Feng Tzeng

*Department of Information Management, Chaoyang University of Technology, 168 Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, R.O.C.*

Cheng-Chi Lee

*Department of Library and Information Science, Fu Jen Catholic University, 510 Jhongjheng Rd., Sinjhuang City, Taipei County 24205, Taiwan, R.O.C.*

Min-Shiang Hwang

*Department of Management Information Systems, National Chung Hsing University, 250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.*
*Corresponding Email: mshwang@nchu.edu.tw*

Received April 2010
Revised June 2010
Communicated by S.G. Akl

ABSTRACT

In this article, we proposed a batch verification scheme for multiple proxy signatures to reduce the proxy signature verification time. The proposed scheme is not only efficient because it does not verify each individual proxy signature separately, but also secure because it can detect forged multiple proxy signatures without failure.

*Keywords*: Digital signature, Proxy signature, Batch verifying, Multiple signatures, Parallel Processing.

## 1. Introduction

The proxy signature is one of the major research topics in digital signatures. The concept of a proxy signature scheme was first introduced by Mambo et al. [15, 16] in 1996. So far, several proxy signature schemes have been widely discussed [7, 12, 13, 19, 20, 22, 25, 26, 27]. The proxy signature scheme allows the original signer to delegate her/his signing capability to a designated person, called a proxy signer. The proxy signer can generate a proxy signature on behalf of the original signer [2, 3, 11, 14, 21, 23, 24]. However, Mambo et al.'s scheme [15, 16] suffers from some weaknesses. Many modified versions of their scheme have been proposed [20].

To generalize the concept of the proxy signature, Yi et al. [26] proposed a new type of proxy signature scheme named the proxy multi-signature scheme. The proxy multi-signature scheme allows a proxy signer to generate a proxy signature on behalf of two or more original signers. However, this scheme is insecure. In [20], Sun showed that the proxy multi-signature by Yi et al. [26] was also vulnerable to some attacks. Consequently, he proposed a new scheme to defeat these weaknesses.

In digital signature schemes, the sender uses her/his private key to sign a message, and the receiver can verify this signature with the sender's public key. If the sender needs to sign $t$ messages to the receiver, the sender has to generate $t$ signatures, and the receiver needs to verify these $t$ signatures one by one. It is inefficient. In 1994, Naccache et al. [17] proposed an efficient scheme to batch-verify multiple DSA digital signatures. Since then, several variations of batch verifying multiple digital signatures have been proposed [1, 4, 5, 8, 9, 10, 18].

In this paper, a new secure and simple batch verification proxy signature scheme will be proposed. The verifier can verify theses multiple digital signatures by using the signer's public key, and this verification process takes only one instead of $t$ verification time units. Furthermore, our scheme can detect forged multiple signatures. The detail analysis is shown in Section 4. In this article, we shall first briefly review the proxy signature scheme and the proxy multi-signature scheme in the following section. In Section 3, our new batch verifying scheme will be proposed. The security analysis and discussion about the proposed scheme will be in Section 4. Finally, the concluding remarks will be given in the last section.

## 2. Review of the Proxy Signature Scheme

In the following two sections, we will briefly review the proxy signature scheme and the proxy multi-signature scheme [20]. Initially, the system parameters are defined as follows. Let $p$ be a large prime, $q$ be a large prime factor of $p-1$, $g$ be a generator for $Z_p^*$, $h(\cdot)$ be a one-way hash function, and $m_w$ be a warrant which records the identities of the original signer, the proxy signer and the valid delegation time, etc. Each original signer $U_{Oi}$ owns a private key $x_{oi} \in Z_q^*$ and a public key $y_{oi} = g^{x_{oi}} \bmod p$ which is certified by a certificate authority (CA). Similarly, the proxy signer $U_P$ also owns a private key $x_p \in Z_q^*$ and a public key $y_p = g^{x_p} \bmod p$ which is also certified by CA. Let $G_O = \{U_{O1}, U_{O2}, \cdots, U_{On}\}$ be the group of $n$ original signers. Here is a brief review of the proxy signature scheme [20]. Assumes that $U_{O1}$ allows a designated $U_P$ to sign on behalf for her/him, and the verifier can check the validity of the proxy signatures.

(1) *Proxy generation:* $U_{O1}$ chooses a random number $t \in Z_q^*$ and computes $k = g^t \bmod p$ and $\sigma = x_{o1}y_{o1} + th(m_w, k) \bmod q$.
(2) *Proxy delivery:* $U_{O1}$ gives $(\sigma, k, m_w)$ to $U_P$ over a public channel.
(3) *Verification and alteration of the proxy:* After receiving $(\sigma, k, m_w)$, $U_P$ checks whether the following equation holds: $g^\sigma = y_{o1}^{y_{o1}} k^{h(m_w,k)} \bmod p$. If it does, $U_P$ computes an alternative proxy signature key $\sigma_p = \sigma + x_p h(m_w, k, y_p) \bmod q$.
(4) *Signing by the proxy signer:* $U_P$ signs a message $m$ by using an ordinary signature scheme with a secret key $\sigma_p$. Assume that the resulting signature is

$Sign_{\sigma_p}(m)$. The proxy signature on $m$ is $(m, Sign_{\sigma_p}(m), k, m_w)$.

(5) *Verification of the proxy signature:* The verifier computes the corresponding public key in the ordinary signature scheme as:

$$y = y_{o1}^{y_{o1}} y_p^{h(m_w,k,y_p)} k^{h(m_w,k)} \bmod p. \tag{1}$$

Then, she/he verifies the validity of $Sign_{\sigma_p}(m)$ by checking the validity of the verification equation in the ordinary signature scheme with the newly generated public key $y$.

Following is a brief review of the proxy multi-signature scheme [20]. $U_P$ is allowed to generate a proxy multi-signature on behalf of $G_O$, and the verifier can check the validity of the proxy multi-signature.

(1) *Subproxy key generation:* $U_{Oi}$ chooses $t_{oi} \in Z_q^*$ and computes $k_{oi} = g^{t_{oi}} \bmod p$ and $\sigma_{oi} = x_{oi} y_{oi} + t_{oi} h(m_w, k_{oi}) \bmod q$ for $i = 1, 2, \cdots, n$.
(2) *Subproxy key delivery:* $U_{Oi}$ gives $(\sigma_{oi}, k_{oi}, m_w)$ to $U_P$ over a public channel for $i = 1, 2, \cdots, n$.
(3) *Subproxy key verification:* After receiving $(\sigma_{oi}, k_{oi}, m_w)$, $U_P$ checks whether the following equation holds: $g^{\sigma_{oi}} = y_{oi}^{y_{oi}} k_{oi}^{h(m_w,k_{oi})} \bmod p$. If it does, she/he accepts it as a valid subproxy key; otherwise, she/he rejects it and requests $U_{oi}$ for a valid one, or she/he terminates this protocol.
(4) *Proxy signature key generation:* If $U_P$ confirms the validity of all $(\sigma_{oi}, k_{oi}, m_w)$ for $i = 1, 2, .., n$, she/he computes $\sigma_p = x_p y_p + \sum_{i=1}^{n} \sigma_{oi} \bmod q$ as a valid proxy signature key.
(5) *Signing by the proxy signer:* $U_P$ executes the signing operation of an ordinary signature scheme using $\sigma_p$. Assume that the resulting signature is $Sign_{\sigma_p}(m)$. The proxy signature of $m$ for $G_O$ is $(m, Sign_{\sigma_p}(m), k_{o1}, \cdots, k_{on}, m_w)$.
(6) *Verification of the proxy multi-signature:* The verifier computes the corresponding public key in the ordinary signature scheme as:

$$y = y_p^{y_p} y_{o1}^{y_{o1}} \cdots y_{on}^{y_{on}} k_{o1}^{h(m_w,k_{o1})} \cdots k_{on}^{h(m_w,k_{on})} \bmod p. \tag{2}$$

Then, she/he verifies the validity of $Sign_{\sigma_p}(m)$ by checking the validity of the verification equation in the ordinary signature scheme with the newly generated public key $y$.

## 3. The Proposed Batch Verification Scheme

In a batch verification for multiple digital signatures, a signer signs $t$ messages by using her/his private key separately and sends the multiple digital signatures to a verifier. The verifier can verify these multiple digital signatures by using the signer's public key, and this verification process takes only one instead of $t$ verification time units.

We assume that the ordinary proxy signature scheme and the ordinary proxy multi-signature scheme share the same parameters as well as signing and verification forms as the original DSA scheme [6, 8]. Let $p$ be a large prime; $q$ be a factor of $(p-1)$, $g$ be a generator with order $q$ in $GF(p)$, and $x$ and $y$ be a signer's private key and public key, respectively. Here, $y = g^x \bmod p$. When a signer wants to send a signed message $m$ to the verifier, she/he must generate a digital signature $(r, s)$ as: $r = g^k \bmod p$, $s = rk - mx \bmod q$, and $k$ is a random number which is generated by the singer. Once receiving $(m, r, s)$ from the signer, the verifier can verify the signature for the message $m$ by checking the equation $r = g^{sr^{-1}} y^{mr^{-1}} \bmod p$.

Assume that $U_P$ wants to send $t$ messages $m_1$, $m_2$, $\cdots$, $m_t$ and thus has to generate $t$ proxy signatures $(m_1, k, m_w, (r_1, s_1))$, $(m_2, k, m_w, (r_2, s_2))$, $\cdots$, $(m_t, k, m_w, (r_t, s_t))$ on behalf of $U_{O1}$ and send them to the verifier, where $r_i = g^{k_{pi}} \bmod p$, $s_i = r_i k_{pi} - m_i \sigma_p \bmod q$, $i = 1, 2, \cdots, t$, $k_{pi}$ is a random number, and $\sigma_p$ is $U_P$'s proxy signature key.

After receiving these proxy signatures from $U_P$, the verifier can verify these multiple proxy signatures for messages $m_1$, $m_2$, $\cdots$, $m_t$. These $t$ proxy signatures satisfy the following $t$ equations. The concept of batch verification is illustrated in Figure 1.

> Input: A batch instance $(m_1, r_1, s_1)$, $(m_2, r_2, s_2)$, $\cdots$, $(m_t, r_t, s_t)$
> Objective: Check if $r_i = g^{s_i r_i^{-1}} y^{m_i r_i^{-1}} \bmod p$ for all $1 < i < t$
> 1) Compute $A = \prod_{i=1}^{t} r_i$.
> 2) Compute $B = g^{\sum_{i=1}^{t} s_i r_i^{-1}} y^{\sum_{i=1}^{t} m_i r_i^{-1}} \bmod p$.
> 3) Accept if $A = B$, else reject.

Fig. 1. The concept of batch verification

$$
\begin{aligned}
r_1 &= g^{s_1 r_1^{-1}} y^{m_1 r_1^{-1}} \bmod p \\
r_2 &= g^{s_2 r_2^{-1}} y^{m_2 r_2^{-1}} \bmod p \\
&\vdots \\
r_t &= g^{s_t r_t^{-1}} y^{m_t r_t^{-1}} \bmod p
\end{aligned}
$$

where $y$ is the same as it is in Equation (1). By multiplying these $t$ equations together, we obtain the batch verification criterion as

$$\prod_{i=1}^{t} r_i = g^{\sum_{i=1}^{t} s_i r_i^{-1}} y^{\sum_{i=1}^{t} m_i r_i^{-1}} \bmod p, \tag{3}$$

where $y$ is the same as it is in Equation (1). However, $U_P$ is allowed to represent $G_O$ to sign message $m_1$, $m_2$, $\cdots$, $m_t$ and generate proxy multi-signatures $(m_1, k_{o1}, ..., k_{on}, m_w, (r_1, s_1))$, $(m_2, k_{o1}, ..., k_{on}, m_w, (r_2, s_2))$, $\cdots$, $(m_t, k_{o1}, ..., k_{on}, m_w, (r_t, s_t))$ and send them to the verifier, where $r_i = g^{k_{pi}} \bmod p$, $s_i = r_i k_{pi} - m_i \sigma_p \bmod q$, $i = 1, 2, \cdots, t$, $k_{pi}$ is a random number, and $\sigma_p$ is $U_P$'s proxy signature key. After receiving these proxy multi-signatures, the verifier verifies the validity of these multiple proxy multi-signatures by Equation (3) except $y$ is the same as it is in Equation (3).

In our scheme, the verifier can verify these $t$ proxy signatures or proxy multi-signatures simultaneously by checking this batch verification criterion. According to our batch verifying proxy signature scheme, if both sides of Equation (3) are equal, the $t$ signatures $(m_1, k, m_w, (r_1, s_1))$, $(m_2, k, m_w, (r_2, s_2))$, $\cdots$, $(m_t, k, m_w, (r_t, s_t))$ for messages $m_1$, $m_2$, $\cdots$, $m_t$, can be verified. However, if not, which may result from invalid individual proxy signatures, then we need to verify each individual proxy signature separately. This situation is similar to what happens to the batch verifying proxy multi-signature scheme. It is obvious that to verify these $t$ proxy signatures or proxy multi-signatures according to the batch verification criterion requires 2 modular exponentiations. However, if the verifier verifies each individual proxy signature or proxy multi-signature separately, it requires $2t$ modular exponentiations.

## 4. Security Analysis and Discussion

Batch verification scheme is simple and efficient to verify multiple proxy signatures and proxy multi-signatures. However, there is a weakness in the above two schemes. A dishonest proxy signer, $U_P$, can forge individual proxy signatures or proxy multi-signatures and make a false batch verification valid [8].

Assume that $U_P$ sends $t$ messages $m_i$ and forges proxy signatures $(m_i, k, m_w, (r_i, s_i))$ to deliver to the verifier. Suppose $s'_i = s_i + a_i r_i \bmod q$, where $a_i$ is an integer such that $\sum_{i=1}^{t} a_i = 0$ for $i = 1, 2, \cdots, t$.

Since these multiple proxy signatures satisfy the verification in Equation (3), the verifier will be convinced that these messages are signed by the dishonest proxy signer, $U_P$. The method is similar to that of batch verifying multiple proxy multi-signatures.

For example, $U_P$ forges three proxy signatures $(m_1, k, m_w, (r_1, s'_1))$, $(m_2, k, m_w, (r_2, s'_2))$ and $(m_3, k, m_w, (r_3, s'_3))$. Let $s'_1 = s_1 + 3r_1 \bmod q$, $s'_2 = s_2 + 5r_2 \bmod q$, $s'_3 = s_3 - 8r_3 \bmod q$, and $s'_i = r_i k_i - m_i \sigma_p \bmod q$, $r_i = g^{k_i} \bmod p$ for $i = 1, 2, 3$. Then $U_P$ sends $(m_1, k, m_w, (r_1, s'_1))$, $(m_2, k, m_w, (r_2, s'_2))$ and $(m_3, k, m_w, (r_3, s'_3))$ to a verifier. The verifier can verify the correctness of the proxy signatures of the messages $m_1$, $m_2$ and $m_3$ by checking the following equation:

$$r_1 r_2 r_3 = g^{s'_1 r_1^{-1} + s'_2 r_2^{-1} + s'_3 r_3^{-1}} y^{m_1 r_1^{-1} + m_2 r_2^{-1} + m_3 r_3^{-1}} \bmod p$$

$$= g^{(s_1+3r_1)r_1^{-1}+(s_2+5r_2)r_2^{-1}+(s_3-8r_3)r_3^{-1}} y^{m_1r_1^{-1}+m_2r_2^{-1}+m_3r_3^{-1}} \bmod p$$
$$= g^{s_1r_1^{-1}+s_2r_2^{-1}+s_3r_3^{-1}} y^{m_1r_1^{-1}+m_2r_2^{-1}+m_3r_3^{-1}} \bmod p,$$

where $y$ is the same as it is in Equation (1). Since the above equation holds, the verifier will believe that the proxy signatures $(m_1, k, m_w, (r_1, s'_1))$, $(m_2, k, m_w, (r_2, s'_2))$ and $(m_3, k, m_w, (r_3, s'_3))$ are valid proxy signatures, respectively. However, $U_O$ and $U_P$ can deny signing these messages to the verifier, because $r_i \neq g^{s'_i r_i^{-1}} y^{m_i r_i^{-1}} \bmod p$ for $i = 1, 2, 3$.

**The Improvement of Batch Verifying Scheme**

The difference of the batch verifying scheme and the improved scheme is only in Equation (3). The improved scheme, as in [9], is modified as

$$\prod_{i=1}^{t} r_i^{v_i} = g^{\sum_{i=1}^{t} s_i r_i^{-1} v_i} y^{\sum_{i=1}^{t} m_i r_i^{-1} v_i} \bmod p, \tag{4}$$

where $y$ is the same as it is in Equation (1) or in Equation (3), $v_i$'s are small random numbers chosen by the verifier. A dishonest proxy signer cannot use the same method in Section 3 to cheat the verifier and pass the batch verifying of the multiple proxy signatures or proxy multi-signatures. After receiving some multiple proxy signatures or proxy multi-signatures, a verifier randomly chooses some integers and verifies the validity of these multiple signatures by using Equation (4). Once one or more signatures are modified, they will fail the validity test. If a dishonest signer wants to make some false multiple proxy signatures $(m_i, k, m_w, (r_i, s'_i))$ or proxy multi-signature $(m_i, k_{o1}, ..., k_{on}, m_w, (r_i, s'_i))$ valid, she/he must make the following equation hold.

$$\prod_{i=1}^{t} s_i r_i^{-1} v_i = \prod_{i=1}^{t} s'_i r_i^{-1} v_i (\bmod q). \tag{5}$$

Since $U_P$ does not know the values $v_i$, he/she is unable to make Equation (5) hold.

## 5. Conclusions

In this paper, we have proposed an efficient and secure scheme to verify multiple proxy signatures and proxy multi-signatures. Instead of verifying each individual proxy signature or proxy multi-signature separately, our new scheme can verify the multiple proxy signatures or proxy multi-signatures simultaneously. Furthermore, the proposed batch verification scheme can save much of the time for verifying multiple signatures.

**Acknowledgements**

This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC96-2219-E-009-013 and NSC98-2221-E-468-002.

**References**

[1] F. Bao, C. C. Lee, and M. S. Hwang, Cryptanalysis and improvement on batch verifying multiple RSA digital signatures, *Applied Mathematics and Computation*, **172** (2006) 1195–1200. **27** (1983) 400–433.
[2] M. L. Das, A. Saxena, D. B. Phatak, Proxy signature scheme with effective revocation using bilinear pairings, *International Journal of Network Security*, **4** (2007) 312–317.
[3] M. L. Das, A. Saxena, D. B. Phatak, Algorithms and approaches of proxy signature: A survey, *International Journal of Network Security*, **9** (2009) 264–284.
[4] L. Harn, Batch verifying multiple DSA-type digital signatures, *Electronics Letters*, **34** (1998) 870–871.
[5] L. Harn, Batch verifying multiple RSA digital signatures, *Electronics Letters*, **34** (1998) 1219–1220.
[6] L. Harn and Y. Xu, Design of generalised ElGamal type digital signature schemes based on discrete logarithm, *Electronics Letters*, **30** (1994) 2025–2026.
[7] M. S. Hwang, I. C. Lin, and Eric J. L. Lu, A secure nonrepudiable threshold proxy signature scheme with known signers, *International Journal of Informatica*, **11** (2000) 1–8.
[8] Min-Shiang Hwang, Cheng-Chi Lee, and Eric Jui-Lin Lu, Cryptanalysis of the batch verifying multiple DSA-type digital signatures, *Pakistan Journal of Applied Sciences*, **1** (2001) 287–288.
[9] Min-Shiang Hwang, Cheng-Chi Lee, and Yuan-Liang Tang, Two simple batch verifying multiple digital signatures, *Proc. of ICICS'2001, LNCS 2229*, 2001, 233–237.
[10] Min-Shiang Hwang, Iuon-Chung Lin, and Kuo-Feng Hwang, Cryptanalysis of the batch verifying multiple RSA digital signatures, *Informatica*, **11** (2000) 15–19.
[11] K. Kim, I. Yie, and S. Lim, Remark on Shao et al.'s bidirectional proxy re-signature scheme in Indocrypt'07, *International Journal of Network Security*, **9** (2009) 8-11.
[12] S. Kim, S. Park, and D. Won, Proxy signatures, revisited, *Proc. of ICICS'97, LNCS 1334*, 1997, 223–232.
[13] N.Y. Lee, T. Hwang, and C.H. Wang, On Zhang's nonrepudiable proxy signature schemes, *ACISP'98, LNCS 1438*, (1998) 415–422.
[14] Eric Jui-Lin Lu and Cheng-Jian Huang, A time-stamping proxy signature scheme using time-stamping service, *International Journal of Network Security*, **2** (2006) 43–51.
[15] M. Mambo, K. Usuda, and E. Okamoto, Proxy signatures: Delegation of the power to sign message, *IEICE Trans. Fundamentals*, **E79-A** (1996) 1338–1353.
[16] M. Mambo, K. Usuda, and E. Okamoto, Proxy signatures for delegating signing operation, *Proc. Third ACM Conf. on Computer and Communications Security*, 1996, 48–57.
[17] D. Naccache, D. Mraihi, D. Rapheali, and S. Vaudenay, Can DSA be improved: Complexity trade-offs with the digital signature standard, *Advances in Cryptology, Eurocrypt'94*, (1994), 85–94.
[18] Z. Shao, Batch verifying multiple DSA-type digital signatures, *Computer Networks*, **37** (2001) 383–389.
[19] Z. Shao, "Proxy signature schemes based on factoring, *Information Processing Letters*, **85** (2003) 137–143.

[20] H. M. Sun, On proxy (multi-) signature schemes, *2000 International Computer Symposium*, Chiayi, Taiwan, Dec. (2000) 65–72.

[21] Zuo-Wen Tan, Improvement on Nominative Proxy Signature Schemes, *International Journal of Network Security*, **7** (2008) 175–180.

[22] Shiang-Feng Tzeng, Chen-Ying Yang, and Min-Shiang Hwang, A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification, *Proceeding of* $12^{th}$ *National Conference on Information Security, R.O.C.*, (2002) 285–292.

[23] G. K. Verma, A proxy blind signature scheme over braid groups, *International Journal of Network Security*, **9** (2009) 214–217.

[24] Qi Xie and Xiu-Yuan Yu, Cryptanalysis of two nonrepudiable threshold proxy signature schemes, *International Journal of Network Security*, **3** (2006) 18–22.

[25] Q. S. Xue and Z. F. Cao, Factoring based proxy signature schemes, *Journal of Computational and Applied Mathematics*, **195** (2006) 229–241.

[26] L. Yi, G. Bai, and G. Xiao, Proxy multi-signature scheme: A new type of proxy signature scheme, *Electronics Letters*, **36** (2000) 527–528.

[27] K. Zhang, Threshold proxy signature schemes, *1997 Information Security Workshop*, (1997) 191–197.