Research Issues and Challenges for Multiple Digital Signatures

Min-Shiang Hwang, and Cheng-Chi Lee,

***Abstract*— In this paper, we survey several well-known batch verification multiple digital signatures. These schemes can batch verify multiple digital signatures which need only one verification instead of t verifications. However, a number of weaknesses of these schemes are pointed out. According to our proposed issues and challenges, we compare and analyze them. To sum up these schemes, a secure and efficient multiple digital signatures scheme which need only one verifications is still an open problem.**

***Index Terms*— Cryptography, Digital signature, DSA, RSA.**

I. INTRODUCTION

Digital signature is a method for Internet which is similar to traditional signature. People sign their true names on papers in traditional signature. No one can forge others's signatures because it is difficult to imitate others's handwritings. To provide digital signature, it uses the known public key cryptosystem. Each one has a pair keys, private key and public key. The private key is kept secret and the public key is public. A sender can sign a electronic document called digital signature using his/her private key and a receiver can verify the digital signature by the sender's public key. No one can forge others's digital signatures because of not knowing private key. A digital signature scheme has the following properties [2], [15], [17]:

1) Only the sender can sign electronic document.
2) The receiver can verify the validity of the digital signature.
3) No one can forge the digital signatures of others.
4) It can achieve integrity. An attacker should not be able to substitute a false document for a legitimate one.
5) It can achieve non-repudiation. A sender should not be able to deny that he/she sent a document.

There are two famous public key cryptosystems to provide digital signatures. One is the RSA digital signature scheme [9], [14]. The security of this scheme is based on the difficulty of solving the factoring problem. The scheme can apply to various areas. The other is ElGamal digital signature scheme [3]. The security of this scheme is based on the difficulty of solving the discrete logarithm problem [8]. It also can apply to various areas. Here, we brief the ElGamal digital signature scheme as follows. Let p is a large prime and g is a generator of Z_p^*. A key pair (private key, public key) is (x, y), where $y = g^x \bmod p$. (p, g, y) are public. Assume that a sender, Alice, wants to send a message m to a receiver, Bob, where $1 \leq m \leq p-1$. Alice chooses a random number, k, to satisfy $GCD(k, p-1) = 1$. Then, compute (r, s) as $r = g^k \bmod p$, and $s = k^{-1}(m - xr) \bmod (p-1)$. (r, s) is digital signature for message m. Alice sends (m, r, s) to Bob. After receiving (m, r, s), Bob can verify the digital signature by using the equation $g^m = y^r r^s \bmod p$. It is seen that the ElGamal digital signature scheme can achieve the above properties of digital signature scheme.

If Alice wants to send a message m, where $m \geq p$, to Bob, the m must be divided into t copies $m_1, m_2, \cdots, m_t$. Then Alice signs these messages t times to generate multiple digital signatures and sends these messages and multiple digital signatures to Bob. Upon receiving these messages and multiple digital signatures, Bob should spend t times to verify the validity of the multiple digital signatures. It is seen that it costs many modular exponentiations. In 1994, Naccache et al. proposed an efficient batch verifying multiple digital signatures [13]. The verifier can verify these multiple digital signatures by the signer's public key which need only one verification instead of t verifications. However, Lim and Lee pointed out their scheme can be easily forged multiple digital signatures to make a false batch verification valid [12]. And then Harn proposed a secure interactive batch verification protocols [4]. In 1998, Harn proposed two type batch verifying multiple digital signatures [5], [6]. However, Hwang et al. pointed out their schemes are also insecure [7], [10]. An attacker can forge multiple digital signatures to make a false batch verification valid. Therefore, they proposed two improvements [11]. In 2001, Shao also proposed an improvement on Harn's scheme [16]. We can see that if the multiple digital signatures are forged by an attacker, the verifier should verify each digital signature. It is back to the original digital signature scheme which need t verifications. In 2002, Changchien and Hwang proposed an efficient detecting forged multiple digital signatures [1].

From the above related researches, some issues and challenges for multiple digital signatures are discussed and then addressed in the design of multiple digital signatures. These are in the following:

1) In multiple digital signatures, only valid signer can sign multiple electronic documents by his/her private key.
2) No one can forge the multiple digital signatures of others to make a false batch verification valid. Many papers cannot against this challenge. An attacker should not forge multiple digital signatures to make a false batch verification valid.

M.S. Hwang is with the Department of Management Information System, National Chung Hsing University, 250 Kuo Kuang Rd., Taichung, Taiwan 402, R.O.C. (Email: mshwang@nchu.edu.tw)

C.C. Lee is with the Department of Computer Science, National Chung Hsing University, 250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C. and Department of Computer & Communication Engineering Taichung Healthcare and Management University, No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan, R.O.C.

3) Any verifier can batch verify the validity of the multiple digital signatures. Any verifier can verify these multiple digital signatures by the signer's public key which need only one verification instead or t verifications. It is seen that batch verifying multiple digital signatures is more efficient than original method which verify each digital signature individually.
4) It should achieve integrity. An attacker should not be able to substitute false documents for a legitimate ones. In other words, multiple documents should not be modified, deleted, or replaced.
5) It should achieve non-repudiation. If a sender can forge multiple digital signatures to make a false batch verification valid. The sender can deny that he/she sent these documents. It cannot achieve non-repudiation. In multiple digital signatures, a sender should not be able to deny that he/she sent these documents.
6) It should be able to detect forged multiple digital signatures efficiently when multiple digital signatures are modified or replaced.

This paper is organized as follows. Section 2-7 describe various multiple digital signatures and their weaknesses. In Section 8, we shall compare them. Finally, the conclusion will be in Section 9.

II. Naccache et al.'s Multiple Digital Signatures Scheme and Its Weakness

In this section, we first brief DSA digital signature [11]. To speed up verification of multiple digital signatures, Naccache et al. proposed a scheme to batch verifying multiple digital signatures [13]. Then, Lim and Lee pointed out Naccache et al.'s scheme has a security flaw [12]. We shown them in this section.

A. DSA

Assume that Alice wants to send a electronic document to Bob. Alice applies DSA algorithm to create a digital signature and send along with document to Bob. Upon receiving document and digital signature, Bob can verify the correctness of the document. The signing signature of DSA algorithm is briefed as follows. In Table I, we list the abbreviations and notations used in DSA algorithm.

1) Alice randomly chooses a number $k \in Z_q$. Then she computes r and s as follows:

$$\begin{aligned} r &= (g^k \bmod p) \bmod q, \\ s &= k^{-1}(m + xr) \bmod q. \end{aligned}$$

r and s are DSA digital signature for message m.
2) Alice sends (m, r, s) to Bob.

The verifying signature of DSA algorithm is briefed as follows:

1) After receiving (m, r, s) from Alice, Bob can verify the correctness of the signature using the following equation:

$$r = (g^{ms^{-1}} y^{rs^{-1}} \bmod p) \bmod q. \tag{1}$$

2) If the above equation holds, Bob can confirm that the digital signature r and s are signed by Alice.

It is seen that If Alice wants to send t multiple digital signatures to Bob, Bob must verify the t multiple digital signatures t times using the verifying signature of DSA algorithm.

B. Naccache et al.'s Scheme

To speed up verification of multiple digital signatures, Naccache et al. proposed a scheme to batch verifying multiple digital signatures. The verifier can verify multiple digital signatures by the signer's public key which need only one verification instead of t verifications. The scheme is as follows:

1) Assume that Alice wants to send t messages $m_1, m_2, ..., m_t$ and multiple digital signatures $(r_1, s_1), (r_2, s_2), ..., (r_t, s_t)$ to Bob. The multiple digital signatures are created by the signing signature of DSA algorithm. It is the same as the above DSA signature.
2) Upon receiving the t messages $m_1, m_2, ..., m_t$ and multiple digital signatures $(r_1, s_1), (r_2, s_2), ..., (r_t, s_t)$, Bob can verify the correctness of these multiple digital signatures on messages $m_1, m_2, ..., m_t$ using Alice's public key y in the following equation:

$$\prod_{i=1}^{t} r_i = (g^{\sum_{i=1}^{t} m_i s_i^{-1}} y^{\sum_{i=1}^{t} r_i s_i^{-1}} \bmod p) \bmod q. \tag{2}$$

3) If the above equation holds, Bob can confirm that the multiple digital signatures $(r_1, s_1), (r_2, s_2), ..., (r_t, s_t)$ are signed by Alice.

It is seen that Bob can batch verify multiple digital signatures which need only one equation 2. Therefore, Naccache et al.'s scheme is more efficient to batch verifying multiple digital signatures.

C. The Weakness of Naccache et al.'s Scheme

Lim and Lee shown that Naccache et al.'s scheme is not secure. It has a security flaw that an attacker can forge the multiple digital signatures and pass the batch verification equation 2. The attack is briefed as follows.

1) An attacker randomly chooses random numbers (u_i, v_i), $i = 1, 2, ..., t$, and computes $r_i = g^{u_i} y^{v_i} \bmod p$, $i = 1, 2, ..., t$.
2) Computes $s_b^{-1} \bmod q$ to satisfy $v_b = r_b s_b^{-1} \bmod q$, $b = 1, 2, ..., t-2$.
3) The attacker can derive the s_{t-1} and s_t from the following equations.

$$\begin{aligned} u_1 + u_2 + ... + u_t &= m_1 s_1^{-1} + m_2 s_2^{-1} + ... + m_t s_t^{-1} \bmod q, \\ v_1 + v_2 + ... + v_t &= r_1 s_1^{-1} + r_2 s_2^{-1} + ... + r_t s_t^{-1} \end{aligned}$$

It is seen that the attacker can forge t messages $m_1, m_2, ..., m_t$ and multiple digital signatures $(r_1, s_1), (r_2, s_2), ..., (r_t, s_t)$ to pass the batch verification equation 2. However, each of the forged signatures does not pass the DSA verification equation 1 individually.

TABLE I
NOTATIONS

x	private key of signer
y	public key of signer
p	a large prime
q	a large prime divisor of $p-1$
g	an element in Z_p of order q

$*$ (g, p, q, y) are public parameters,
where $y = g^x \bmod p$.
And x is kept secure by signer.

III. Harn's DSA-type Multiple Digital Signatures Scheme and Its Weakness

To against the above Lim-Lee's attack, Harn proposed a batch verifying multiple DSA-type digital signatures. In this section, we first brief DSA-type digital signature. Then, we review Harn's batch verifying multiple digital signatures [5]. Next, Hwang et al. pointed out Harn's scheme has a security flaw [10]. We shown them in this section.

A. DSA-type

The DSA-type is similar to DSA algorithm. Assume that Alice wants to send a electronic document to Bob. Alice applies DSA-type algorithm to create a digital signature and send along with document to Bob. Upon receiving document and digital signature, Bob can verify the correctness of the document. The signing signature of DSA-type algorithm is briefed as follows. The algorithm also uses the same abbreviations and notations in Table I.

1) Alice randomly chooses a number $k \in Z_q$. Then she computes r and s as follows:

$$\begin{aligned} r &= (g^k \bmod p) \bmod q, \\ s &= rk - mx \bmod q. \end{aligned}$$

r and s are DSA-type digital signature for message m.

2) Alice sends (m, r, s) to Bob.

The verifying signature of DSA-type algorithm is briefed as follows:

1) After receiving (m, r, s) from Alice, Bob can verify the correctness of the signature using the following equation:

$$r = (g^{sr^{-1}} y^{mr^{-1}} \bmod p) \bmod q. \qquad (3)$$

2) If the above equation holds, Bob can confirm that the digital signature r and s are signed by Alice.

It is also seen that If Alice wants to send t multiple digital signatures to Bob, Bob must verify the t multiple digital signatures t times using the verifying signature of DSA-type algorithm.

B. Harn's DSA-type Scheme

Harn proposed a batch verifying DSA-type multiple digital signatures to speed up verification of multiple digital signatures. The scheme is as follows:

1) Assume that Alice wants to send t messages $m_1, m_2, ..., m_t$ and multiple digital signatures $(r_1, s_1), (r_2, s_2), ..., (r_t, s_t)$ to Bob. The multiple digital signatures are created by the signing signature of DSA-type algorithm. It is the same as the above DSA-type signature.

2) Upon receiving the t messages $m_1, m_2, ..., m_t$ and multiple digital signatures $(r_1, s_1), (r_2, s_2), ..., (r_t, s_t)$, Bob can verify the correctness of these multiple digital signatures on messages $m_1, m_2, ..., m_t$ using Alice's public key y in the following equation:

$$\prod_{i=1}^{t} r_i = (g^{\sum_{i=1}^{t} s_i r_i^{-1}} y^{\sum_{i=1}^{t} m_i r_i^{-1}} \bmod p) \bmod q. \qquad (4)$$

3) If the above equation holds, Bob can confirm that the multiple digital signatures $(r_1, s_1), (r_2, s_2), ..., (r_t, s_t)$ are signed by Alice.

It is also seen that Bob can batch verify multiple digital signatures which need only one equation 4. Therefore, Harn's scheme is more efficient to batch verifying multiple digital signatures.

C. The Weakness of Harn's Scheme

Hwang et al. shown that Harn's scheme is not secure. It has a security flaw that a singer can forge the multiple digital signatures and pass the batch verification equation 4. Then, the signer can deny the digital signatures. It cannot achieve the property non-repudiation. The attack is briefed as follows.

1) Alice sends the forged pairs (m_i, r_i, s'_i), $i = 1, 2, \cdots, t$, to Bob, where $s'_i = s_i + a_i r_i \bmod q$ and a_i is an integer such that $\sum_{i=1}^{t} a_i = 0$.

2) Upon receiving forged pairs (m_i, r_i, s'_i), Bob can pass the batch verification equation 4. And Bob can confirm that the multiple digital signatures $(r_1, s'_1), (r_2, s'_2), ..., (r_t, s'_t)$ are signed by Alice.

It is seen that each of the forged signatures does not pass the DSA-type verification equation 3 individually. Therefore, Alice can deny these multiple digital signatures. It cannot achieve the property non-repudiation because of $r_i \neq (g^{s'_i r_i^{-1}} y^{m_i r_i^{-1}} \bmod p) \bmod q$, $i = 1, 2, \cdots, t$.

IV. Harn's RSA Multiple Digital Signatures Scheme and Its Weakness

In this section, we first brief Harn's RSA multiple digital signatures scheme [6] and then shown its weakness [7].

A. Harn's Scheme

The scheme is based on RSA algorithm. Let p and q are two large primes. Computes $n = p \times q$ and chooses e and d such that $e \times d \bmod (p-1)(q-1) \equiv 1$. (e, n) is a signer's public key and d is signer's private key. Assume that a singer, Alice, wants to send a message to a receiver, Bob. Alice generates her digital signature S by using RSA algorithm, where $S = h(m)^d \bmod n$ and $h(\cdot)$ is a public one-way hash function. Alice sends (m, S) to Bob. After receiving them, Bob can verify the correctness of the signature by checking the equation $h(m) = S^e \bmod n$. If it holds, Bob can confirm that the digital signature S are signed by Alice.

Now, assume that Alice wants to send t messages $m_1, m_2, \cdots, m_t$ and multiple digital signatures $S_1, S_2, \cdots, S_t$ to Bob. The multiple digital signatures are created by the above RSA algorithm. Then, Alice sends (m_i, S_i), $i = 1, 2, \cdots, t$, to Bob.

Upon receiving the t messages $m_1, m_2, ..., m_t$ and multiple digital signatures $S_1, S_2, \cdots, S_t$, Bob can verify the correctness of these multiple digital signatures on messages $m_1, m_2, ..., m_t$ using Alice's public key e in the following equation:

$$(\prod_{i=1}^{t} S_i)^e = \prod_{i=1}^{t} h(m_i) \bmod n. \tag{5}$$

If the above equation holds, Bob can confirm that the multiple digital signatures $S_1, S_2, \cdots, S_t$ are signed by Alice.

It is seen that Bob can batch verify multiple digital signatures which need only one equation 5. Therefore, Harn's scheme is more efficient to batch verifying multiple digital signatures.

B. The Weakness of Harn's Scheme

Hwang et al. proposed two attacks on Harn's scheme. They shown that a singer can forge the multiple digital signatures and pass the batch verification equation 5. Then, the signer can deny the digital signatures. It cannot achieve the property non-repudiation. The attacks is briefed as follows.

Attack 1:

Alice sends her forged pairs (m_i, S_i'), $i = 1, 2, \cdots, t$ to Bob, where $S_i' = h(m_{f(i)})^d \bmod n$, $i = 1, 2, \cdots, t$; $f(\cdot)$ is a one to one and onto function such that $f(i) = j$, $i = 1, 2, \cdots, t$ and $j = 1, 2, \cdots, t$. Upon receiving forged pairs (m_i, S_i'), Bob can pass the batch verification equation 5. And Bob can confirm that the multiple digital signatures $S_1', S_2', \cdots, S_t'$ are signed by Alice.

Attack 2:

Alice sends her forged pairs (m_i, S_i'), $i = 1, 2, \cdots, t$ to Bob, where $S_i' = a_i \times S_i \bmod q$, $i = 1, 2, \cdots, t$ and $\prod_{i=1}^{t} a_i = 1$. Upon receiving forged pairs (m_i, S_i'), Bob can pass the batch verification equation 5. And Bob can confirm that the multiple digital signatures $S_1', S_2', \cdots, S_t'$ are signed by Alice.

It is seen that each of the forged signatures does not pass the RSA verification equation $h(m_i) = S_i'^e \bmod n$ individually. Therefore, Alice can deny these multiple digital signatures. It cannot achieve the property non-repudiation.

V. Hwang et al.'s Multiple Digital Signatures Schemes

To remedy the weaknesses of Harn's DSA-type multiple digital signatures scheme and RSA multiple digital signatures scheme, Hwang et al. proposed two improved schemes [11]. One is improved DSA-type multiple digital signatures scheme (shortly called BV-DSA). The other is improved RSA multiple digital signatures scheme (shortly called BV-RSA). Two improved schemes are shown as follows.

A. Improved DSA-type Multiple Digital Signatures Scheme

The improved scheme is similar to Harn's scheme. The only difference is in equation 4. We modify it to

$$\prod_{i=1}^{t} r_i^{v_i} = (g^{\sum_{i=1}^{t} s_i r_i^{-1} v_i} y^{\sum_{i=1}^{t} m_i r_i^{-1} v_i} \bmod p) \bmod q, \tag{6}$$

where v_i, $i = 1, 2, \cdots, t$, are small random numbers which are chosen by a verifier.

B. Improved RSA Multiple Digital Signatures Scheme

The improved scheme is similar to Harn's scheme. The only difference is in equation 5. We modify it to

$$(\prod_{i=1}^{t} S_i^{v_i})^e = \prod_{i=1}^{t} h(m_i)^{v_i} \bmod n. \tag{7}$$

where v_i, $i = 1, 2, \cdots, t$, are small random numbers which are chosen by a verifier.

VI. Shao's DSA-type Multiple Digital Signatures Scheme

In 2001, Shao proposed a DSA-type multiple digital signatures scheme [16]. His scheme is similar to Hwang et al.'s scheme. The only difference is in equation 7. The batch verifying criterion is as follows.

$$\prod_{i=1}^{t} (e_i(s_i))^{u_i} = \prod_{i=1}^{t} (f_i(s_i))^{u_i} \bmod p, \tag{8}$$

where $u_i \in (1, 2^{32})$, $i = 1, 2, \cdots, t$, are random numbers which are chosen by a verifier and s_i are multiple digital signatures individually satisfying the verification equations: $e_i(s_i) = f_i(s_i) \bmod p$, $i = 1, 2, \cdots, t$.

VII. Changchien et al.'s Detecting RSA Forged Multiple Digital Signatures Scheme

In section 4, it is seen that If the batch verification fails, that is $(\prod_{i=1}^{t} S_i)^e \neq \prod_{i=1}^{t} h(m_i) \bmod n$, $i = 1, 2, \cdots, t$, the receiver, Bob, must check each multiple digital signatures using the verification equation $h(m_i) = S_i^e \bmod n$. It needs t exponential computations to detect the forged digital signature. In 2002, Changchien and Hwang proposed a scheme to detect forged multiple digital signature [1], which only need one exponential and t modulus computations.

Changchien and Hwang redefine $h()$ as a prime one-way hash function and $\prod_{i=1}^{t} h(m_i) \leq n$. This can let the length of $h()$ is $\lfloor |n|/t \rfloor$ bits, where $\lfloor \cdot \rfloor$ is floor function and $|n|$ is length of n. To detect the forged multiple digital signature, we can use the following steps:

1) Bob calculates $L = (\prod_{i=1}^{t} S_i)^e \bmod n$.
2) Bob can detect the forged digital signature S_i by checking whether $L \bmod h(m_i) = 0$, for $i = 1, 2, \cdots, t$.

VIII. Comparisons

A lot of multiple digital signatures have been proposed in recent years. The security of these schemes are based on the difficulty of solving the factoring problem [14] and discrete logarithm problem [3]. In this section, we compare them in terms of properties and efficiency.

A. Properties

In Introduction, some issues and challenges for multiple digital signatures are discussed and then addressed in the design of multiple digital signatures. These are also the properties of multiple digital signatures. Table II shows these schemes in properties. We compare them as follows:

1) *Only valid signer can sign multiple electronic documents.*
   All schemes can meet this property. If the signer has his/her private key, he/she can do that.
2) *No one can forge the multiple digital signatures.*
   In these schemes, it is seen that only Hwang's BV-DSA, Hwang's BV-RSA, and Shao's scheme can meet this property. The multiple digital signatures of their scheme cannot be forged to make a false batch verification valid.
3) *Any verifier can batch verify the validity of the multiple digital signatures.*
   All schemes can meet this property. If the verifier has the signer's public key, he/she can batch verify the correctness of the multiple digital signatures which need only one verification.
4) *It should achieve integrity.*
   All schemes can meet this property. An attacker should not be able to substitute false documents for a legitimate ones because he/she does not know the private key of the signer. Only the signer can generate digital signatures for particular documents.
5) *It should achieve non-repudiation.*
   If the multiple digital signatures can be forged by a sender, it cannot meet this property because the sender can deny that he/she signed these multiple digital signatures. Therefore, only Hwang's BV-DSA, Hwang's BV-RSA, and Shao's scheme can meet this property.
6) *It should be able to detect forged multiple digital signatures efficiently.*
   Most of these schemes cannot meet this property. Only Changchien et al.'s scheme can meet this property. When the multiple digital signatures are forged, the verifier can detect these forged multiple digital signatures efficiently.

B. Efficiency

Compare with the original verification of digital signature, these multiple digital signatures can speed up verification of multiple digital signatures. These schemes batch verify multiple digital signatures which need only one verification instead of t verifications. It can save many modular exponentiations.

IX. Conclusions

In this paper, we have reviewed some multiple digital signatures and proposed some issues and challenges for them. Compare with these multiple digital signatures, we can see that not all these schemes can against these issues and challenges. However, any verifier in these schemes can batch verify the validity of the multiple digital signatures. They can save many modular exponentiations. In future, a secure and efficient multiple digital signatures scheme which meets all issues and challenges is still an open problem.

Acknowledgements
Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC93-2213-E-005-005.

References

[1] S. Wesley Changchien and Min-Shiang Hwang, "A batch verifying and detecting multiple RSA digital signatures," *International Journal of Computational and Numerical Analysis and Applications*, vol. 2, no. 3, pp. 303–307, 2002.

[2] Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

[3] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, July 1985.

[4] L. Harn, "DSA type secure interactive batch verification protocol," *Electronics Letters*, vol. 31, no. 4, pp. 257–258, 1995.

[5] L. Harn, "Batch verifying multiple DSA-type digital signatures," *Electronics Letters*, vol. 34, no. 9, pp. 870–871, 1998.

[6] L. Harn, "Batch verifying multiple RSA digital signatures," *Electronics Letters*, vol. 34, no. 12, pp. 1219–1220, 1998.

[7] M. S. Hwang, I. C. Lin, and K. F. Hwang, "Cryptanalysis of the batch verifying multiple RSA digital signatures," *Informatica*, vol. 11, no. 1, pp. 15–19, 2000.

[8] Min-Shiang Hwang, Chin-Chen Chang, and Kuo-Feng Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.

[9] Min-Shiang Hwang, Cheng-Chi Lee, and Yan-Chi Lai, "Traceability on RSA-based partially signature with low computation," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 465–468, 2003.

[10] 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*, vol. 1, no. 3, pp. 287–288, 2001.

[11] Min-Shiang Hwang, Cheng-Chi Lee, and Yuan-Liang Tang, "Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13–16, Xian, China, 2001.

[12] C. H. Lim and P. J. Lee, "Security of interactive DSA batch verification," *Electronics Letters*, vol. 30, no. 19, pp. 1592–1593, 1994.

[13] D. Naccache, D. Mraihi, D. Rapheali, and S. Vaudenay, "Can DSA be improved: Complexity trade-offs with the digital signature standard," in *Proceedings of Eurocrypt'94*, pp. 85–94, Lecture Notes in Computer Science, 1994.

[14] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[15] Bruce Schneier, *Applied Cryptography, 2nd Edition*. New York: John Wiley & Sons, 1996.

[16] Zuhua Shao, "Batch verifying multiple DSA-type digital signatures," *Computer Networks*, vol. 37, no. 3-4, pp. 383–389, 2001.

TABLE II
COMPARISONS AMONG THE MULTIPLE DIGITAL SIGNATURES SCHEMES

	IC1	IC2	IC3	IC4	IC5	IC6	SP
Naccache et al.[13]	Y	N	Y	Y	N	N	DLP
Harn's DSA[5]	Y	N	Y	Y	N	N	DLP
Harn's RSA[6]	Y	N	Y	Y	N	N	FP
Hwang's BV-DSA[11]	Y	Y	Y	Y	Y	N	DLP
Hwang's BV-RSA[11]	Y	Y	Y	Y	Y	N	FP
Shao[16]	Y	Y	Y	Y	Y	N	DLP
Changchien et al.[1]	Y	N	Y	Y	N	Y	FP

ICi: Proposed issues and challenges in Section 1, Y: Supported, N: Not supported,
SP: Based on the difficulty of solving problem, DLP: Discrete logarithm problem, FP: Factoring problem.

[17] Shiang-Feng Tzeng, Cheng-Ying Yang, and Min-Shiang Hwang, "A new digital signature scheme based on factoring and discrete logarithms," *International Journal of Computer Mathematics*, vol. 81, no. 1, pp. 9–14, 2004.

Min-Shiang Hwang was born on August 27, 1960 in Tainan, Taiwan, Republic of China (ROC.). He received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, ROC, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field ”Electronic Engineer” in 1988. He also passed the National Telecommunication Special Examination in field ”Information Engineering”, qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He obtained the 1997, 1998, 1999, 2000, and 2001 Outstanding Research Award of National Science Council of the Republic of China. He is currently a professor of the department of Management Information System, National Chung Hsing University, Taiwan, ROC. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang had published over 100 articles on the above research fields in international journals.

Cheng-Chi Lee received the B.S. and M.S. in Information Management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, Republic of China, in 1999 and in 2001. He researched in Computer and Information Science from National Chiao Tung University (NCTU), Taiwan, Republic of China, from 2001 to 2003. He is currently pursuing his Ph.D. in Computer Science from National Chung Hsing University (NCHU), Taiwan, Republic of China. He is a Lecturer of Computer and Communication, Taichung Healthcare and Management University (THMU), from 2004. His current research interests include information security, cryptography, and mobile communications. Dr. Lee had published over 25 articles on the above research fields in international journals.