# Security Analysis of Some Batch Verifying Signatures from Pairings

Tianjie Cao[1,2,3], Dongdai Lin[2], and Rui Xue[2]
*(Corresponding author: Tianjie Cao)*

School of Computer Science and Technology, China University of Mining and Technology[1]
Xuzhou 221008, P.R. China (Email: tjcao@is.iscas.ac.cn)
State Key Laboratory of Information Security of Institute of Software, Chinese Academy of Sciences[2]
Beijing 100080, P.R. China
Graduate School of the Chinese Academy of Sciences[3]
Beijing 100039, P.R. China

*(Received Aug. 5, 2005; revised and accepted Sept. 12 & Oct. 4, 2005)*

## Abstract

Batch verification can provide large computational savings when multiple signatures are verified together. Recently, some batch verifying signature schemes have been proposed from bilinear pairings. In this paper, we show that an attacker can cheat a verifier to accept invalid signatures in these batch verifying schemes. We also show that randomized batch verification technique can be used to avoid these attacks.

*Keywords: batch verification, signature, security analysis, randomization*

## 1 Introduction

Batch verification can reduce large computational cost when multiple signatures are verified together. When a collection of signatures passes the batch verifications, the verifier accepts all the signatures as valid. Otherwise, the collection is rejected. Undoubtedly, security of batch verification scheme is to be of utmost importance. If an attacker can cheat a verifier to accept invalid signatures in batch verifying schemes, it means that a consumer can forge coins in electronic pay system and a voter can forge votes in electronic voting system.

The idea of batch cryptography was introduced by Fiat [4, 5]. Fiat proposed a modified version of RSA suitable for batch signature generations. In 1994, Naccache et al. [12] proposed the first DSA batch verification scheme. The authors introduced batch verification to verify several DSA signatures at once and is much more efficient than sequential verification of individual DSA signatures. An earlier version of the paper [12] included an additional interactive batch verifier. Lim and Lee showed that this version is not secure since any attacker can easily forge multiple individual signatures to make a false batch verification valid [11]. Bellare et al. [2] proposed small exponents test technique to overcome this security problem. In 1998, Harn proposed two efficient non-interactive batch verification protocols for DSA-type and RSA-type multiple signatures [6, 7]. However, Harn's both schemes are insecure [8, 9]. In [13], Yen and Laih proposed a randomized batch verification of a modification of the Schnorr or Brickell-McCurley signature schemes as well as for RSA. In [10], Hwang and Lee surveyed several well-known batch verification multiple digital signatures. Some issues and challenges for multiple digital signatures are discussed.

Recently, Yoon et al. classified multiple signatures (i.e. input of batch verification) into the following three types, according to the number of signers and messages [14]:

**Type 1.** multiple signatures on a single message generated by multiple signers.

**Type 2.** multiple signatures on multiple messages generated by a single signer.

**Type 3.** multiple signatures on multiple messages generated by multiple signers, where each message is signed by a distinct user.

Blind signatures allow the user to obtain a signature of a message in a way that the signer learns neither the message nor the resulting signature. In [15], Zhang et al. proposed an ID-based blind signature scheme from bilinear pairings. Since the computation of the pairing is the most time-consuming, in order to enhance the efficiency of verification process the authors gave a batch verification algorithm of Type 2. Zhang et al. also suggested the same batch verification in Cha-Cheon ID-based signature scheme.

Partially blind signatures allow the signer to explicitly include some agreed information in the blind signature. In [16], Zhang et al. proposed an efficient partially blind signature scheme from bilinear pairings. To improve the efficiency of their scheme, the authors also presented a batch verification algorithm.

In [14], Yoon et al. provided a loose security reduction of batch verification of Type 2 in Cha-Cheon ID-based signature scheme [3] to the computational Diffie-Hellman problem. They showed that Cha-Cheon scheme is not secure in batch verification of Type 1 or 3. Yoon et al. also proposed a new ID-based signature scheme from bilinear pairings in batch verification of Types 1 and 3 and provide security proof under random oracle model.

The purpose of this paper is to illustrate flaws in a number of recent published batch verifying signatures from pairings [14, 15, 16]. We show Zhang et al.'s batch verifying algorithm and Yoon et al. batch verifying algorithm are all insecure and an attacker can cheat a verifier to accept invalid signatures. Finally, applying randomized technique [2, 9, 12, 13] we propose a randomized batch verifying multiple signatures algorithm to avoid these attacks.

# 2 Basic Concepts on Bilinear Pairings

In this section, we introduce the bilinear pairings and the related mathematical problems [1]. Let $G_1$ be a cyclic additive group generated by $P$, whose order is a prime $q$, and $G_2$ be a cyclic multiplicative group with the same order $q$: Let $e : G_1 \times G_1 \longrightarrow G_2$ be a bilinear pairing with the following properties:

**Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1, a, b \in Z_q$.

**Non-degeneracy:** 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 $G_2$.

**Computability:** There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

A bilinear map satisfying the three properties above is said to be an admissible bilinear map. Suppose that $G_1$ is an additive group. Now we describe four mathematical problems.

1) Discrete Logarithm Problem (DLP): Given two group elements $P$ and $Q$, find an integer $n$, such that $Q = nP$ whenever such an integer exists.

2) Decision Diffie-Hellman Problem (DDHP): For $a, b, c \in Z_q^*$, given $P, aP, bP, cP$ decide whether $c \equiv ab \bmod q$. If so, $(P, aP, bP, cP)$ is called a valid Diffie-Hellman tuple.

3) Computational Diffie-Hellman Problem (CDHP): For $a, b \in Z_q^*$, given $P, aP, bP$ compute $abP$.

4) Diffie-Hellman Problem (GDHP): A class of problems where DDHP is easy while CDHP is hard.

When the DDHP is easy but the CDHP is hard on the group $G_1$, we call $G_1$ a Gap Diffie-Hellman (GDH) group.

# 3 Descriptions of Some Batch Verifying Signature Schemes from Pairings

In this section, we briefly review Zhang et al.'s batch verifying partially blind signature scheme, Zhang et al.'s batch verifying blind signature scheme and Yoon et al.'s new ID-based batch verifying signature scheme.

## 3.1 Zhang et al.'s Batch Verifying Partially Blind Signature Scheme

In [16], Zhang et al. proposed a partially blind signature scheme which can work on any Gap Diffie-Hellman group. The system parameters are defined as follows: Let $P$ be a generator of $G_1$ with order $q$; the bilinear pairing is given by $e : G_1 \times G1 \longrightarrow G_2$. Define two cryptographic hash function $H : \{0,1\}^* \longrightarrow \{0,1\}^\lambda$, in general, $|q| \geq \lambda \geq 160$, and $H_0 : \{0,1\}^* \longrightarrow G_1^*$. The system parameters are: $params = \{G_1, G_2, e, q, \lambda, P, H, H_0\}$.

**Key generation:** Let $\in_R$ denote the uniform random selection. The signer picks random $x \in_R Z_q^*$, and computes $P_{pub} = xP$. The public key is $P_{pub}$. The secret key is $x$.

**Partially blind signature issuing protocol:** Suppose that $m$ be the message to be signed and $c$ be the public information.

Generation of the public information: The user and signer generate the public information $c$ together.

Blinding: The user randomly chooses a number $r \in_R Z_q^*$, computes $U = H_0(m||c) + r(H(c)P + P_{pub})$, and sends $U$ to the signer.

Signing: The signer sends back $V$, where $V = (H(c) + x)^{-1}U$.

Unblinding: The user computes $S = V - rP$.

Then $(S, m, c)$ is the partially blind signature of the message $m$ and public information $c$.

**Verification:** A verifier can accept this partially blind signature if and only if $e(H(c)P + P_{pub}, S) = e(P, H_0(m||c))$

Assuming that $S_1, S_2, \ldots, S_n$ are partially blind signatures on messages $m_1, m_2, \ldots, m_n$ with the same public

information $c$. Zhang et al. suggested the following batch verifying algorithm [16].

**Batch Verification (For the same public information $c$):** Assuming that $S_1, S_2, \ldots, S_n$ are partially blind signatures on messages $m_1, m_2, \ldots, m_n$ with the same public information $c$. The batch verification is then to test if the following equation holds:

$$e(H(c)P + P_{pub}, \sum_{i=1}^{n} S_i) = e(P, \sum_{i=1}^{n} H_0(m_i||c)).$$

### 3.2 Zhang et al.'s Batch Verifying Blind Signature Scheme

In [3], Cha and Cheon proposed an ID-based signature scheme using gap GDH groups. Under the random oracle model, Cha-Cheon scheme is proved to be secure against existential forgery on adaptively chosen message and ID attack assuming CDHP is intractable. In [15], Zhang et al. proposed a new ID-based blind signature scheme, which can be regarded as the blind version of Cha-Cheon's ID-based signature scheme.

**Setup:** Let $(G_1, +)$ and $(G_2, \bullet)$ denote cyclic groups of prime order $q$, let $P$ be a generator of $G_1$ and the bilinear pairing is given as $e : G_1 \times G_1 \longrightarrow G_2$. Pick a random $s \in Z_q^*$ and set $P_{pub} = sP$, choose two cryptographic hash functions $H_2 : \{0,1\}^* \longrightarrow G_1^*$ and $H_1 : 0,1\}^* \times G_1^* \longrightarrow Z_q^*$. The system parameters are **params** $= < q, G_1, G_2, e, P, P_{pub}, H_1, H_2 >$. The **master-key** is $s \in Z_q^*$.

**Extract:** For a given string $ID \in \{0,1\}^*$, the PKG computes $Q_{ID} = H_2(ID)$, and sets the private key $d_{ID}$ to be $d_{ID} = sQ_{ID}$ where $s$ is the master key.

**Blind signature issuing protocol:** Suppose that $m$ is the message to be signed. The signer randomly chooses a number $r \in Z_q^*$, compute $U = rQ_{ID}$, and sends $U$ to the user as a commitment.

Blinding: The user randomly chooses $\alpha, \beta \in Z_q^*$, as blinding factors. He/She computes $U' = \alpha U + \alpha\beta Q_{ID}$ and $h = \alpha^{-1} H_1(m, U') + \beta$, sends $h$ to the signer.

Signing: The signer sends back $V$, where $V = (r+h)d_{ID}$.

Unblinding: The user computes $V' = \alpha V$ . He/She outputs $(m, U', V')$.
Then $(U', V')$ is the blind signature of the message $m$.

**Verification:** To verify a signature $(U', V')$ of a message $m$ for an identity $ID$, check whether $e(P, V') = e(P_{pub}, U' + hQ_{ID})$ or not where $h = H_1(m, U')$.
Let $\sigma_i = (m_i, U_i', h_i, V_i')$ be the signatures signed by a single user with $ID$ on distinct $k$-messages $m_i$. Zhang et al. suggested the following batch verifying algorithm [15].

**Batch Verification (Type 2):** To verify all $k$-signatures at once. The verifier computes $Q_{ID} = H_1(ID)$ and $h_i = H_1(m_i, U_i')$ for all $i = 1, \ldots, k$. The verifier check whether $e(P, \sum_{i=1}^{k} V_i') = e(P_{pub}, \sum_{i=1}^{k} U_i' + (\sum_{i=1}^{k} h_i)Q_{ID})$ or not.

A similar batch verifying algorithm in Cha-Cheon signature scheme of Type 2 was independently proposed by Zhang et al. [15] and Yoon et al. [14]. The detail description of batch verification of Type 2 in Cha-Cheon scheme can be found in [14].

### 3.3 Yoon et al.'s New ID-based Batch Verifying Signature Scheme

In [14], Yoon et al. showed that Cha-Cheon scheme is not secure in batch verification of Type 1 or 3. They then proposed a new ID-based signature scheme from bilinear pairings in batch verification of Types 1 and 3. The proposed signature scheme consists of four phases: Setup, Extract, Signing, and Verification.

**Setup:** Let $(G_1, +)$ and $(G2, \bullet)$ denote cyclic groups of prime order $q$, let $P$ be a generator of $G_1$ and the bilinear pairing is given as $e : G_1 \times G_1 \longrightarrow G_2$. Pick a random $s \in Z_q^*$ and set $P_{pub} = sP$, choose two cryptographic hash functions $H_2 : \{0,1\}^* \longrightarrow G_1^*$ and $H_1 : 0,1\}^* \times G_1^* \longrightarrow Z_q^*$. The system parameters are **params** $= < q, G_1, G_2, e, P, P_{pub}, H_1, H_2 >$. The **master-key** is $s \in Z_q^*$.

**Extract:** For a given string $ID \in \{0,1\}^*$, the PKG computes $Q_{ID} = H_2(ID)$, and sets the private key $d_{ID}$ to be $d_{ID} = sQ_{ID}$ where $s$ is the master key.

**Signing:** Given a secret key $d_{ID}$ and a message $m$, choose a random number $r \in Z_q^*$, compute $U = rP$, $h = H_1(m, U)$, and $V = rQ_{ID} + hd_{ID}$. Output a signature $\sigma = (U, V)$.

**Verification:** To verify a signature $\sigma = (U, V)$ of a message $m$ for an identity $ID$, check whether $e(P, V) = e(Q_{ID}, U + hP_{pub})$ where $h = H_1(m, U)$.
Given $k$ signatures $(ID_1, m_1, U_1, h_1, V_1), \ldots, (ID_k, m_k, U_k, h_k, V_k)$, Yoon et al. proposed a following batch verifying algorithm [10].

**Batch Verification (Type 3):** To verify all $k$-signatures at once, the verifier computes $Q_{ID} = H_1(ID)$ and $h_i = H_1(m_i, U_i)$ for all $i = 1, \ldots, k$. The verifier check whether $e(P, \sum_{i=1}^{k} V_i) = \prod_{i=1}^{k} e(Q_i, U_i + h_i P_{pub})$ or not.

## 4 Security Analysis

In the case of individual signature verification, an attacker is forced to break the underlying signature scheme if he

wants to generate a valid signature. In the case when the batch verification is applied, the attacker may also explore weaknesses existing in the verification equations. In this section, we show how an attacker can cheat a verifier to accept invalid signatures in Zhang et al.'s and Yoon et al. batch verifying algorithm.

### 4.1 Attack 1

We firstly discuss the security of Zhang et al.'s batch verifying partially blind signature scheme.

Assuming that $S_1, S_2, \ldots, S_n$ are partially blind signatures on messages $m_1, m_2, \ldots, m_n$ with the same public information $c$. Choose the $n$ - 1 values $S'_1, S'_2, \ldots, S'_{n-1}$ randomly and finally solve the equation $S'_1+S'_2+\ldots+S'_n = S_1+S_2+\ldots+S_n$ to obtain the value $S'_n$. Then the batch $(S'_1, m_1, c), \ldots, (S'_n, m_n, c)$ satisfies the batch verification but almost certainly none of the signatures is correct.

The attack 1 also exists in Zhang et al.'s batch verifying blind signature scheme, batch verifying Cha-Cheon signature scheme and Yoon et al.'s new batch verifying ID-based signature scheme.

In Yoon et al.'s new batch verifying ID-based signature scheme, there are $k + 1$ pairing operations for verifier. Obviously, This is inefficient and undesirable in practice.

### 4.2 Attack 2

There is another attack to against Zhang et al.'s batch verifying multiple signatures algorithm.

In Zhang et al.'s batch verifying partially blind signature scheme, the user randomly chooses a number $r \in_R Z_q^*$, computes $U = H_0(m_1||c) + H_0(m_2||c) + \ldots + H_0(m_n||c) + r(H(c)P + P_{pub})$, and sends $U$ to the signer. The signer sends back $V$, where $V = (H(c) + x)^{-1}U$. The user obtains $S = V - rP$. The $(n + 2)$-tuple $(S, m_1, m_2, \ldots, m_n, c)$ satisfies the following equation:

$$\begin{aligned} & e(H(c)P + P_{pub}, S) \\ = \; & e(P, H_0(m_1||c) + H_0(m_2||c) + \ldots + H_0(m_n||c)). \end{aligned}$$

The user can choose the $n-1$ values $S'_1, S'_2, \ldots, S'_{n-1}$ randomly and finally solve the equation $S'_1 + S'_2 + \ldots + S'_n = S$ to obtain the value $S'_n$. Then the batch $(S'_1, m_1, c), \ldots, (S'_n, m_n, c)$ satisfies the batch verification but almost certainly none of the signatures is correct.

In Zhang et al.'s ID-based batch verifying blind signature scheme, the detail attack is as follows.

The signer randomly chooses a number $r \in Z_q^*$, compute $U = rQ_{ID}$, and sends $U$ to the user.

The user randomly chooses $\alpha, \beta \in Z_q^*$, as blinding factors. He/She computes $U' = \alpha U + \alpha\beta Q_{ID}$ and chooses the $k$ values $U'_1, U'_2, \ldots, U'_k$ randomly such that $U'_1 + U'_2 + \ldots + U'_k = U'$. The user computes $h = \alpha^{-1}\sum_{i=1}^{k} h_i + \beta$ where $h_i = H_1(m_i, U'_i)$, sends $h$ to the signer.

The signer sends back $V$, where $V = (r + h)d_{ID}$.

The user computes $V' = \alpha V$. The $(2n + 1)$-tuple $(U'_1, U'_2, \ldots, U'_k, m_1, m_2, \ldots, m_n, V')$ satisfies the following equation:

$$e(P, V') = e(P_{pub}, U' + \sum_{i=1}^{k} h_i Q_{ID}).$$

The user choose the $k$ values $V'_1, V'_2, \ldots, V'_k$ randomly such that $V'_1 + V'_2 + \ldots + V'_k = V'$. Then the batch $(m_i, U'_i, h_i, V'_i)$ satisfies the batch verification but almost certainly none of the signatures is correct.

## 5 Randomized Batch Verifications

To remedy the weaknesses of above batch verification algorithms, we can apply randomized technique [2, 9, 12, 13]. The key point of randomisation is that applying random factors in batch verification equation. An attacker who wishes to have an incorrect batch accepted has to anticipate which random values will be used.

**Batch Verification (Type 2 in Zhang et al.'s partially blind signature scheme):** Assuming that $S_1, S_2, \ldots, S_n$ are partially blind signatures on messages $m_1, m_2, \ldots, m_n$ with the same public information $c$ in Zhang et al.'s signature scheme. The verifier randomly chooses $n-1$ random factors $w_2, w_3, \ldots, w_n \in_R Z_q^*$ and injects these random factors into the batch verification equation. The batch verification is then to test if the following equation holds:

$$\begin{aligned} & e(H(c)P + P_{pub}, S_1 + w_2S_2 + \ldots + w_nS_n) = \\ & e(P, H_0(m_1||c) + w_2H_0(m_2||c) + \ldots + w_nH_0(m_n||c)). \end{aligned}$$

In our batch verification algorithm, a dishonest user cannot use the same methods in Section 4 to cheat a verifier of passing the batch verification equation.

We first consider attack 1. After receiving some multiple signatures $(S_1, m_1, c), (S_2, m_2, c), \ldots, (S_n, m_n, c)$ a verifier randomly chooses $n-1$ randomizing factors $w_2, w_3, \ldots, w_n \in_R Z_q^*$ and verifies the validation of theses multiple signatures by the batch verification equation. Once one or more signatures are modified, the verifier fails the validation of the batch verifying signatures. If a dishonest user wants to choose some false multiple digital signatures $(S'_i, m_i, c)$ valid, he must to make the following equation holds:

$$S'_1 + w_2S'_2 + \cdots + w_nS'_n = S_1 + w_2S_2 + \ldots + w_nS_n.$$

Since the user did not know randomizing factors $w_2, w_3, \ldots, w_n$, he is difficult to choose some $S'_i \neq S_i$ satisfying $S'_1 + w_2S'_2 + \ldots + w_nS'_n = S_1 + w_2S_2 + \ldots + w_nS_n$.

Now we consider attack 2. Since the user cannot predict the random factors $w_2, w_3, \ldots, w_n$ chosen by the verifier, he cannot prepare a proper $U$ to require a partially blind signature.

Our suggested batch verification is more efficient than separate verified signatures. We assume that Zhang et al. scheme is using the GDH group derived from the curve $\mathbf{E}/\mathbf{F}_3^{163}$ defined by the equation $y^2 = x^3 - x + 1$. The group provides 1551-bit discrete-log security. The computation of the pairing is the most time-consuming. For example, according to the best result in [1, 17], one pairing operation is about 11110 multiplications in $\mathbf{F}_3^{163}$, while a point scalar multiplication of $\mathbf{E}/\mathbf{F}_3^{163}$ is a few hundred multiplications in $\mathbf{F}_3^{163}$ . Thus, our batch verification is very efficient compare with linearly verifying each individual signature one by one.

Similarly, we can construct randomized batch verifications in Zhang et al.'s blind signature scheme, Cha-Cheon scheme and Yoon et al.'s new ID-based signature scheme. Here we only describe batch verifications in Zhang et al.'s ID-based blind signature scheme.

**Batch Verification (Type 2 in Zhang et al.'s blind signature):** Let $\sigma_i = (m_i, U_i', h_i, V_i')$ be the signatures using Zhang et al.'s blind signature scheme signed by a single user with $ID$ on distinct $k$-messages $m_i$, to verify all $k$-signatures at once, the verifier chooses $k-1$ randomizing factors $w_2, w_3, \ldots, w_k \in_R Z_q^*$ and computes $Q_{ID} = H_1(ID)$ and $h_i = H_1(m_i, U_i')$ for all $i = 1, \ldots, k$. The verifier check whether $e(P, V_1' + \sum_{i=2}^{k} w_i V_i') = e(P_{pub}, U_i' + \sum_{i=2}^{k} w_i U_i' + (h_1 + \sum_{i=2}^{k} w_i h_i) Q_{ID})$ or not.

**Batch Verification (Types 1 and 3 in Zhang et al.'s blind signature):** Given $k$ blind signatures $(ID_1, m_1, U_1', h_1, V_1'), \ldots, (ID_k, m_k, U_k', h_k, V_k')$, to verify all $k$-signatures at once, the verifier computes $Q_{ID} = H_1(ID)$ and $h_i = H_1(m_i, U_i')$ for all $i = 1, \ldots, k$. The verifier check whether $e(P, V_1' + \sum_{i=2}^{k} w_i V_i') = e(P_{pub}, U_i' + h_1 Q_1 + \sum_{i=2}^{k} w_i (U_i' + h_i Q_i))$ or not.

# 6 Conclusion

In this paper we have shown some attacks on Zhang et al.'s batch verifying partially blind signature scheme, Zhang et al.'s ID-based batch verifying blind signature scheme, batch verifying Cha-Cheon scheme and Yoon et al.'s new batch verifying signature scheme. We have shown how these attacks may be avoided by a randomized batch verifying multiple signatures algorithm.

# Acknowledgements

We thank the anonymous reviewers for their many helpful comments on this paper. We thank the support of the National Grand Fundamental Research 973 Program of China (No. 2004CB318004), the National Natural Science Foundation of China (NSFC90204016, NSFC60373048) and the National High Technology Development Program of China under Grant (863, No. 2003AA144030).

# References

[1] P. S. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," in *SAC'2003*, pp. 17–25, 2004.

[2] M. Bellare and J.A. Garay, "Fast batch verification for modular exponentiation and digital signatures," in *EUROCRYPT'98*, pp. 236–250, 1998.

[3] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *PKC'2003*, pp. 18–30, 2003.

[4] A. Fiat, "Batch RSA," in *Crypto'89*, LNCS 435, pp. 175–185, 1990.

[5] A. Fiat, "Batch RSA," *Journal of Cryptology*, vol. 10, no. 2, pp. 75–88, 1997.

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

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

[8] 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.

[9] M. S. Hwang, C. C. Lee, and Y. L. Tang, "Two simple batch verifying multiple digital signatures," in *Information and Communications Security (ICICS'01)*, pp. 233–237, 2001.

[10] M. S. Hwang and C.C. Lee. "Research issues and challenges for multiple digital signatures," *International Journal of Network Security*, vol. 1, no. 1, pp. 1–6, 2005.

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

[12] D. Naccache, D. M'Rïhi, D. Raphaeli, and S. Vaudenay, "Can DSA be improved: complexity trade-offs with the digital signature standard," in *Eurocrypt'94*, pp. 77–85, 1994.

[13] S. M. Yen and C. Laih, "Improved digital signature suitable for batch verification," *IEEE Transactions on Computers*, vol. 44, no. 7, pp. 957–959, 1995.

[14] H. Yoon, J. H. Cheon, and Y. Kim, "Batch verifications with ID-based signatures," in *Information Security and Cryptology - ICISC 2004*, pp. 233–248, 2005.

[15] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," in *ACISP'03*, Australia, pp. 312–323, 2003.

[16] F. Zhang, R. Safavi-Naini, and W. Susilo, "Efficient verifiably encrypted signature and partially blind signature from bilinear pairings," in *Indocrypt 2003*, New Delhi, pp. 191–204, 2003.

[17] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *PKC 2004*, pp. 277–290, 2004.

**Tianjie Cao** received the M.Sc. degree in Mathematics from Nankai University (P.R. China) in 1993. Currently he is Associate Professor of Computer Science at China University of Mining and Technology. He is also ph.D candidate at the State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences. His research interests include cryptographic protocols and network security. E-mail address: tjcao@is.iscas.ac.cn.

**Dongdai Lin** received the M.Sc. and Ph.D. degree in Cryptography at Institute of System Sciences, Chinese Academy of Sciences in 1990. Currently he is a research professor at the State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences. E-mail address: ddlin@is.iscas.ac.cn.

**Rui Xue** received the Ph.D. degree in Mathematics from Beijing Normal University (P.R. China) in 1999. He was a post-doctorial fellow at the Laboratory of Computer Science in the Institute of Software (1999-2001), Chinese Academy of Sciences (CAS). Currently he is a research professor at the State Key Laboratory of Information Security, Institute of Software, CAS. E-mail address: rxue@is.iscas.ac.cn