A Batch Verification Scheme by Using the Matrix-Detection Algorithm

Yi-Li Huang, Chu-Hsing Lin, Fang-Yie Leu
Department of Computer Science, TungHai University, Taiwan
{yifung, chlin, leufy}@thu.edu.tw

Abstract

When a government office O delivers a batch of documents to a remote government office R, for security consideration, R has to verify whether the documents received are those originally sent by O or not. To do so, we need a security mechanism to perform the verification. That is, before sending the documents, the mailroom of O encrypts them with a private key, i.e., a digital signature. On receiving the documents, the mailroom of R decrypts them as a verification test with the corresponding public key. However, verifying the signatures of the documents received one by one is a crucial and inefficient work. In fact, if we can treat the documents received as a whole, and verify their signatures simultaneously, the verification efficiency will be higher. Therefore, a batch verification approach, a method simultaneously verifying a batch of signatures as a whole, was then proposed. In literatures, some batch verification schemes cannot efficiently and effectively identity bad signatures existing in a set of given signatures. The Small Exponent Test, a popular batch verification method, has its own problems, e.g., after a test, bad signatures still exist with some escape probability. In this paper, we propose a batch verification approach, called Matrix-Detection Algorithm (MDA for short), with which when the number of bad signatures in a batch of signatures is less than four, the batch cannot pass the MDA verification test. Analytical results show that the MDA is more secure and efficient than the SET.

Keywords: Homeland security/defense, Batch verification, Small exponent test, Matrix-Detection algorithm, Escape probability.

1 Introduction

Currently, many government offices or private companies electronically exchange their documents through the Internet to speed up the document delivery [1]. Basically, if the two offices/companies have strong business relationship between them, it is very often that many documents will be sent from one office/company to another at the same time [1-2]. For security consideration, it would be better if these documents are first encrypted by the document creators beforehand to avoid the exposure of the document contents. On the other hand, during the document delivery, if we can provide a verification mechanism to further protect the delivery of the documents [3-5], either encrypted or unencrypted, the security level of the document exchange system will be higher, particularly for those documents needed to be securely delivered. For example, a land office after finishing land sale/purchase transactions sends the information of the transactions as a backup copy to the local government, or a travel agency delivers the entrance application forms to the immigration office of a country for foreigners wishing to enter the country before the foreigners start for the country.

That means the keys used to encrypt/decrypt delivered documents are different from those employed by document creators to protect their document contents. Before sending a document to its destination, the sender will encrypt the document again by using the private keys for verification. Then, the receiving side needs to crucially decrypt them one by one. This is an inefficient work [3][6]. Therefore, a security scheme, called the digital-signature batch verification system (the batch verification system for short), is then proposed.

The batch verification scheme, first proposed by Naccache [3] in 1994, treats a batch of documents/signatures as a whole, and verifies them simultaneously. In real applications, the efficiency of a batch verification scheme is higher than that of a conventional verification method when verifying a large number of signatures signed individually [7]. However, batch verification schemes might be vulnerable to forged signature attacks, also called bad-signature attacks or dirty-signature attacks, i.e., due to exploiting the vulnerabilities of the batch verification schemes, a batch of signatures that contains bad signatures generated by hackers may pass the verification test.

Lim and Lee [8], Boyd and Pavlovski [9], and Hwang et al. [10] introduced several attack methods for existing batch verification schemes. Bellare et al. [7] and Hwang et al. [10] proposed verification methods to enhance the security of batch verification schemes. Nevertheless, with either Bellare's or Hwang's method, bad signatures can still pass the verification test with some escape probabilities.

In this paper, we propose a novel batch verification method, called Matrix-Detection Algorithm (MDA for short), with which when a batch of signatures that contains less than four bad signatures cannot pass the MDA verification test. Let P_{max} be the maximum escape probability of the MDA verification test, then P_{max} decreases

*Corresponding author: Fang-Yie Leu; E-mail: leufy@thu.edu.tw as the number of digital signatures or the number of bad signatures increases where the escape probability is the probability that bad signatures can pass a batch verification test. Analytical results show that the MDA is more secure and efficient than a state-of-the-art verification method, the Small Exponent Test (SET for short). In the following, we use message and documents interchangeably since we treat a document delivered as a message.

The rest of this paper is organized as follows. Section 2 describes the background and related work of the study. Section 3 introduces the proposed method. Analytical results are presented and discussed in Section 4. Section 5 concludes this paper and outlines our future research.

2 Background and Related Work

A batch verification scheme often consists of the *signing phase* and *verification phase*. Generally, there are two types of currently existing signature schemes, the DSA-type signature [11] and the RSA signature [12]. The former is a specific type of the ElGamal signature [13], the security of which heavily relies on the difficulty of solving a discrete logarithm problem, whereas the security of the latter was based on the hardness of solving the factorial problem.

2.1 DSA-type Batch Verification

The DSA-type batch verification scheme first proposed by Naccache [3] in 1994 is defined as follows.

Parameters:

p: A large prime.

q: A prime factor of $(p-1)$.

g: An element of order q in GF(p) where GF stands for Galois Field [3].

x: The signer's private key.

y: The signer's public key.

In the signing phase, n signers use two equations, $r_i = (g^{k_i} \bmod p)$ and $s_i = (k_i^{-1}(m_i + xr_i)) \bmod q$, to generate n signatures SIG = $\{sig_i = (r_i, s_i)$: for $i = 1, 2, \ldots, n\}$ where k_i and m_i are signer i's random number and message, respectively. After that, SIG is sent to the verifier, which in the verification phase employs the batch verification equation

$$\prod_{i=1}^{n} r_i \overset{?}{\equiv} g^{\sum_{i=1}^{n} m_i s_i^{-1} \bmod q} \, y^{\sum_{i=1}^{n} r_i s_i^{-1} \bmod q} \pmod{p} \tag{1}$$

to simultaneously verify the signatures contained in the received SIG. If both sides of the equation are identical, the n signatures pass the verification, i.e., authenticated.

2.2 RSA Batch Verification

A RSA batch verification scheme proposed by Harn [12] in 1998 is defined as follows.

Parameters:

N: The modulo of RSA.

e: The signer's public key.

d: The signer's private key.

In the signing phase, a set of n signatures SIG = $\{sig_i = s_i$: $i = 1, 2, \ldots, n\}$ is generated by using the equation $s_i = m_i^d \bmod N$ where m_i is the message to be signed by signer i.

On receiving the SIG, the verifier verifies the n signatures simultaneously by using the following equation.

$$\left(\prod_{i=1}^{n} s_i\right)^{e} \overset{?}{\equiv} \prod_{i=1}^{n} m_i \bmod N \tag{2}$$

If both sides of the equation are identical, the n signatures are authenticated.

2.3 The Small Exponent Test (SET)

The SET [7] is an innovative method used to verify whether bad signatures exist in a batch of signatures or not. When the SET is invoked by a DSA-type batch verification scheme, the signing phase is the same as that without employing the SET. But in the verification phase, the verifier first chooses n random numbers $b_1, b_2, \ldots, b_n$, each of which is l bits in length, and then authenticates the signatures by using the following equation.

$$\prod_{i=1}^{n} r_i^{b_i} \overset{?}{\equiv} g^{\sum_{i=1}^{n} m_i s_i^{-1} b_i \bmod q} \, y^{\sum_{i=1}^{n} r_i s_i^{-1} b_i \bmod q} \pmod{p} \tag{3}$$

If both sides of the equation are identical, the n signatures are authenticated.

When the SET is invoked by a RSA-type batch verification scheme, the signing phase is also the same as that without employing the SET. In the verification phase, the verifier also chooses n *l-bit* random numbers $b_1, b_2, \ldots, b_n$, and verifies the signatures by using the following equation.

$$\left(\prod_{i=1}^{n} s_i^{b_i}\right)^{e} \overset{?}{\equiv} \prod_{i=1}^{n} m_i^{b_i} \bmod N \tag{4}$$

If both sides of the equation are identical, the n signatures are authenticated.

3 The Proposed Method

In this study, we collect a set of signature samples from the given set of n signatures. In the verification phase, the samples are verified to determine whether these n signatures are authenticated or not.

3.1 Notations and Definitions

The notations used by the MDA, including an initial matrix, a checking matrix, an index set, an *n-batch*, a signature sample and a decision algorithm, are as follows.

Definition 1. An initial matrix $G = [g_{ij}] = [c_1, c_2, \ldots, c_n]$ is a binary $k \times n$ matrix in which $g_{ij} = 0$ or 1, $1 \leq i \leq k$, $1 \leq j \leq n$, $2^{k-1} \leq n \leq 2^k - 1$, and

$$k = \begin{cases} (\log_2 n) + 1, & \text{if n is a power of 2} \\ \lceil \log_2 n \rceil, & \text{otherwise} \end{cases} \qquad (5)$$

Furthermore, $c_j = [g_{1j}, g_{2j}, \ldots, g_{kj}]^T$ is the jth column of G, which consists of k binary digits, and the decimal value of c_j is j, i.e., $(c_j)_{10} = j$, for $j = 1, 2, \ldots, n$.

Definition 2. A *checking matrix* $H = [h_{ij}]$ is a binary $k \times n$ matrix obtained by randomly ordering the columns of $G_{k \times n}$.

Example 1:

$$G_{3\times5} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \text{ and}$$

$$H_{3\times5} = \left[h_{ij}\right]_{3\times5} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Definition 3. An *index set* $T_i = \{j: h_{ij} = 1, 1 \leq j \leq n\}$ derived from H is the set of column indices of i's row entries, the entry value of which is 1. We call T_i the i-th index set of H, $1 \leq i \leq k$.

The index sets derived from $H_{3\times5}$ shown in Example 1 are $T_1 = \{1,3\}$, $T_2 = \{4,5\}$, and $T_3 = \{1,2,4\}$.

Definition 4. An *n-batch* $S = \{(m_i, sig_i): i = 1, 2, \ldots, n\}$ is a set of n (message, digital signature) pairs where m_i is message i and sig_i is the corresponding signature. In a DSA-type system, $sig_j = (r_j, s_j)$ [3], and in a RSA-type system, $sig_j = s_j$ [12].

Definition 5. Given an *n-batch* S, a signature *sample* S_i is a set of signatures selected from S based on index set T_i where $S_i = \{(m_j, sig_j): j \in T_i, 1 \leq j \leq n\}$, $1 \leq i \leq k$.

In Example 1, the signature *samples* corresponding to the index sets of T_1, T_2, and T_3 shown above are $S_1 = \{(m_1, sig_1), (m_3, sig_3)\}$, $S_2 = \{(m_4, sig_4), (m_5, sig_5)\}$, and $S_3 = \{(m_1, sig_1), (m_2, sig_2), (m_4, sig_4)\}$, respectively.

Definition 6. A *decision algorithm* on an *n-batch* S, denoted by $DA(S)$, is a Boolean-valued function which is *True* if S passes the corresponding verification test. Otherwise, $DA(S) = False$.

Definition 7. Two signatures (m_a, sig_a) and (m_b, sig_b) are *distinguishable* by a signature sample S_i if either $\{(m_a, sig_a) \in S_i, (m_b, sig_b) \notin S_i\}$ or $\{(m_a, sig_a) \notin S_i, (m_b, sig_b) \in S_i\}$.

3.2 The Conventional Verification Scheme

Given a DSA-type n-batch S, if S passes the DSA-type batch verification test shown in Equation (1), we call that S passes the conventional verification scheme. Further, the left-hand side of Equation (1) can be rewritten as

$$\begin{aligned} \prod_{i=1}^{n} r_i \mod p &\equiv g^{\log_g(\prod_{i=1}^{n} r_i \mod p)} \\ &\equiv g^{(\log_g r_1 + \log_g r_2 + \ldots + \log_g r_n) \mod p} \\ &\equiv g^{\sum_{i=1}^{n} \log_g r_i \mod p} \equiv g^{\sum_{i=1}^{n} k_i' \mod p} \end{aligned}$$

where $k_i' = \log_g r_i$, and the right-hand side can be rewritten to

$$\begin{aligned} & g^{\sum_{i=1}^{n} m_i s_i^{-1} \mod q} y^{\sum_{i=1}^{n} r_i s_i^{-1} \mod q} \mod p \\ &\equiv g^{\left[\sum_{i=1}^{n} m_i s_i^{-1} \mod q + (\sum_{i=1}^{n} r_i s_i^{-1} \mod q) \log_g y\right] \mod p} \\ &\equiv g^{\left[(\sum_{i=1}^{n} m_i s_i^{-1} + \sum_{i=1}^{n} x r_i s_i^{-1}) \mod q\right] \mod p} \\ &\equiv g^{\left[\sum_{i=1}^{n} s_i^{-1}(m_i + x r_i) \mod q\right] \mod p} \equiv g^{\left[\sum_{i=1}^{n} k_i'' \mod q\right] \mod p} \end{aligned}$$

That is,

$$g^{\sum_{i=1}^{n} k_i' \mod p} \equiv g^{\left[\sum_{i=1}^{n} k_i'' \mod q\right] \mod p}, k_i' = \log_g r_i,$$
$$k_i'' = s_i^{-1}(m_i + x r_i), x = \log_g y \qquad (6)$$

Note that k_i' is not necessary equal to k_i'' [7]. A hacker only needs to fabricate a set of signatures $S' = \{(m_{j_t}, sig_{j_t}) : 1 \leq j_t \leq n$ and $t = 1, 2, \ldots, r\}$ as a subset of $S = \{(m_j, sig_j): j = 1, 2, \ldots, n\}$, $r \leq n$, making

$$\sum_{i=1}^{n} k_i' \equiv (\sum_{i=1}^{n} k_i'' \mod q) \pmod{p}, k_i' = \log_g r_i,$$
$$k_i'' = s_i^{-1}(m_i + x r_i), x = \log_g y \qquad (7)$$

then, S would pass the verification test. Namely, a hacker can penetrate the verification system by generating S' without solving the discrete logarithm problem since the decision algorithm for a DSA-type batch verification on S, i.e., $DA(S)$, only checks to see whether Equation (7) holds or not. If $DA(S) = True$, then S passes the verification test.

The decision algorithm on the i-th signature sample $S_i = \{(m_j, sig_j): j \in T_i, 1 \leq j \leq n\}$, called a signature sampling test on S_i (a sampling test for short) and denoted by $DA(S_i)$, checks to see whether Equation (8) holds or not where $sig_j = (r_j, s_j)$.

$$\sum_{j \in T_i} k_j' \overset{?}{\equiv} (\sum_{j \in T_i} k_j'' \mod q) \pmod{p},$$
$$k_i' = \log_g r_i, k_i'' = s_i^{-1}(m_i + x r_i), x = \log_g y \qquad (8)$$

If $DA(S_i) = True$, then S_i passes the test.

Similarly, the decision algorithm designed for verifying a RSA-type *n-batch* $S = \{(m_i, sig_i): i = 1, 2, \ldots, n\}$, i.e., $DA(S)$, only checks to see whether Equation (2) holds or not. If $DA(S) = True$, then S passes the test, and we call that S passes the conventional verification scheme. A hacker can generate S' to penetrate the verification system without solving the factorial problem. The decision algorithm on the i-th signature sample $S_i = \{(m_j, sig_j): j \in T_i, 1 \leq j \leq n\}$, i.e., a sampling test on S_i and also denoted by $DA(S_i)$, verifies whether Equation (9) holds or not where $sig_j = s_j$.

$$\left(\prod_{j\in T_i} s_j\right)^{e} \overset{?}{\equiv} \prod_{j\in T_i} m_j \mod N. \qquad (9)$$

If $DA(S_i) = True$, then S_i passes the test.

3.3 Batch Verification by Using a Checking Matrix

To test an n-batch S to see whether S is authenticated or not, we first generate a $k \times n$ checking matrix $H_{k\times n}$, and derive its index sets and k signature samples. By testing the k samples, bad signatures can be detected efficiently.

Theorem 1 shows that two different signatures in a given n-batch S are distinguishable by at least one signature sample of S.

Theorem 1: Two signatures (m_a, sig_a) and (m_b, sig_b) in an n-batch $S = \{(m_i, sig_i): i = 1, 2, \ldots, n\}$, a ≠ b, are *distinguishable* by at least one *signature sample*, e.g., S_q, $q \in \{1, 2, 3, \ldots, k\}$, where k is calculated based on Equation (5).

Proof: In the checking matrix $H_{k\times n} = [h_{ij}] = [c_1, c_2, \ldots, c_n]$, column c_i is corresponding to signature sig_i, $i = 1, 2, \ldots, n$. Since the binary digit permutation of $H_{k\times n}$ on all columns are distinct, two arbitrary columns, e.g., c_a and c_b, $1 \leq a, b \leq n$, a ≠ b, then $c_a \neq c_b$. Therefore, $\exists q$, $1 \leq q \leq k$ such that $h_{qa} \neq h_{qb}$, implying that either $a \in T_q, b \notin T_q$ or $a \notin T_q, b \in T_q$, i.e., $((m_a, sig_a) \in S_q$ and $(m_b, sig_b) \notin S_q)$ or $((m_a, sig_a) \notin S_q$ and $(m_b, sig_b) \in S_q)$. In other words, (m_a, sig_a) and (m_b, sig_b) can be distinguished by signature sample S_q. Q.E.D.#

The following theorem will show that when there is only one bad signature in S, the bad signature will be discovered by MDA (See Algorithm 1).

Theorem 2: If only one bad signature exists in an n-batch S, then the signature will be discovered by the MDA.

Proof: For a DSA-type (a RSA-type) batch verification scheme, $DA(S)$ checks to see whether Equation (7) (Equation [2]) holds or not. Let (m_j, sig_j) be the bad signature for some j, $1 \leq j \leq n$. Then Equation (7) will become $\sum_{i=1,i\neq j}^{n} k_i' + k_j' \equiv \left[\left(\sum_{i=1,i\neq j}^{n} k_i'' + k_j''\right) \bmod q\right] \bmod p$, which can further derive Equation (10).

$$k_j' \equiv (k_j'' \bmod q) \pmod{p} \qquad (10)$$

Equation (10) will not hold because that (m_j, sig_j) is a bad signature. Similarly, Equation (2) will reduced to

$$(s_j)^{e} \equiv (m_j) \bmod N \qquad (11)$$

```
Algorithm 1: MDA algorithm /* MDA verification test by using a checking matrix $H_{k\times n}$ */
  Input:   An n-batch $S$.
  Output:  $S$ is authenticated or rejected.
  Flag=True;
  Compute $\sigma = DA(S)$;
  If $\sigma$ = False, then {Flag=Fail; go to L;}
  Generate a $k \times n$ checking matrix $H_{k\times n}$ from a given $k \times n$ initial matrix $G_{k\times n}$;
  Compute the index sets $T_i$ and the signature samples $S_i$,
      for all $i$, $1 \leq i \leq k$;
  For $i$ = 1,2,...,$k$
      {Compute $\sigma_i = DA(S_i)$;
      If $\sigma_i$ = False, then {Flag=Fail; go to L;}}
  EndFor;
L:  If Flag=True, the batch of $n$ signatures is
      authenticated, else reject it;
```

Equation (11) will not hold because that (m_j, sig_j) is a bad signature. Then $DA(S)$ is false, implying that the bad signature (m_j, sig_j) can be discovered by MDA. Q.E.D.#

If S has only two bad signatures, we will prove that the MDA can discover them.

Theorem 3: If an n-batch S has only two DSA-type bad signatures, S cannot pass the MDA verification test.

Proof: Proof is by contradiction. Let (m_a, sig_a) and (m_b, sig_b) be the two bad signatures in S. If they do not pass the conventional verification scheme, $DA(S)$ is false. Otherwise, according to Equation (7), the equation $k_a' + k_b' = ((k_a'' + k_b'')$ mod q) (mod p) holds where $sig_a = (s_a, r_a)$ and $sig_b = (s_b, r_b)$. We claim that S cannot pass the MDA verification test since (m_a, sig_a) and (m_b, sig_b) need to pass all the k sampling tests, i.e., Equation (8), in the situation where $((m_a, sig_a) \in S_q$ and $(m_b, sig_b) \in S_q)$ or $((m_a, sig_a) \notin S_q$ and $(m_b, sig_b) \notin S_q)$ holds for all $q = 1, 2, \ldots, k$. However, according to Theorem 1, the two bad signatures must be distinguishable, i.e., the situation does not hold. Hence, the two bad signatures can be discovered by the MDA. Q.E.D.#

Theorem 4: If an n-batch S has only two RSA-type bad signatures, S cannot pass the MDA verification test.

Proof: If two bad signatures (m_a, sig_a) and (m_b, sig_b) in S do not pass the conventional verification scheme, then $DA(S)$ is false. Otherwise, according to Equation (2), the equation $(s_a s_b)^e \equiv m_a m_b \pmod{N}$ holds where $sig_a = s_a$ and $sig_b = s_b$. We claim that no such an n-batch exists since when S is detected by the MDA, (m_a, s_a) and (m_b, s_b) need to pass all the k sampling tests $\{\left(\prod_{j\in T_i} s_j\right)^e \equiv \prod_{j\in T_i} m_j$ mod N: for i = 1, 2, …, $k\}$ (see Equation [9]) in the situation where the two bad signatures coexist or do not exist in each of the k sampling tests. Similarly, by Theorem 1, the situation does not hold. Hence, the two bad signatures will be discovered. Q.E.D.#

Theorem 5 will show that when S contains three bad signatures, it cannot pass the MDA test.

Theorem 5. If an n-batch S has only three bad signatures (m_a, sig_a), (m_b, sig_b), and (m_c, sig_c), then S cannot pass the MDA verification test.

Proof: If the three DSA-type of bad signatures pass the conventional verification scheme, then $DA(S)$ holds and Equation (7) becomes:

$$k_a' + k_b' + k_c' \equiv ((k_a'' + k_b'' + \mathrm{k}_c'') \bmod \mathrm{q}\ )\ (\mathrm{mod\ p}) \quad (12)$$

We claim that such S cannot pass the MDA verification test. Since either one or two of the three bad signatures will be discovered by the k sampling tests. Now, if one of the three bad signatures is detected by one of the k sampling tests, by Theorem 1, it will be discovered. If two of the three bad signatures, e.g., (m_a, sig_a) and (m_c, sig_c), are detected by one of the k sampling tests, then these two signatures may be discovered. For the latter, Equation (7) holds and becomes:

$$k_a' + k_c' \equiv ((k_a'' + \mathrm{k}_c'') \bmod \mathrm{q}\ )\ (\mathrm{mod\ p}) \quad (13)$$

When both Equations (12) and (13) hold, indicating that (m_b, sig_b) is a pure signature, rather than a bad signature. This contradicts the assumption that (m_b, sig_b) is a bad signature, implying that the two equations do not hold, simultaneously. Hence, S cannot pass the MDA verification test. Similarly, if the signatures are RSA-type, then S cannot pass the MDA verification test. Q.E.D.#

A specific type of attacks is as follows.

Example 2. Given an 7-batch S and a 3 × 7 checking matrix

$$H_{3\times 7} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} = [c_1c_2c_3c_4c_5c_6c_7].$$

Assume that there are 4 bad signatures in S, and they are the fourth to the seventh signatures. The corresponding columns in $H_{3\times 7}$ are c_4, c_5, c_6 and c_7. If S is a DSA-type 7-batch and S passes the MDA verification test, then the 4 bad signatures can result in

$$m_4' = m_4 + h\times s_4,\ m_5' = m_5 - h\times s_5,\ m_6' = m_6 - h\times s_6,\ m_7' = m_7 + h\times s_7,\ h\in R.$$

According to Equation (7), s_4^{-1} $(m_4' + xr_4) = s_4'$ $(m_4 + xr_4) + h = k_4'' + h$. Similarly, s_5^{-1} $(m_5' + xr_5) = k_5'' - h$, s_6^{-1} $(m_6' + xr_6) = k_6'' - h$, s_7^{-1} $(m_7' + xr_7) = k_7'' + h$.

Then $DA(S)$ is true, and $DA(S_i)$ is true, for all i, $1 \le i \le 3$. Also, $\{(m_4', sig_4'), (m_7', sig_7')\}$ generates a dirty value +h, and $\{(m_5', sig_5'), (m_6', sig_6')\}$ generates another dirty value – h. Hence, the sum of these dirty values is zero. Thus, $\{(m_4', sig_4'), (m_5', sig_5'), (m_6', sig_6'), (m_7', sig_7')\}$ passes the MDA verification test and forms a self-compensating escape set (SCE-set for short) which is a subset of S that contains bad signatures but passes the MDA verification test by generating dirty values which mutually compensate for each other, thus generating a zero as the result. In Example 2, $\{(m_4', sig_4'), (m_7', sig_7')\}$ and $\{(m_5', sig_5'), (m_6', sig_6')\}$ form two mutually compensating escape subsets (MCE-s-sets for short).

For RSA-type of signatures, the 4 bad signatures can, like those in Example 2, generate

$$m_4' = h\ m_4, m_5' = \frac{m_5}{h}, m_6' = \frac{m_6}{h}, m_7' = h\ m_7,\quad h\in R$$

According to Equation (2),

$$\prod_{j=1}^{4} m_j' = \ m_4' \cdot m_5' \cdot m_6' \cdot m_7' = (hm_4)\cdot(\frac{m_5}{h})\cdot(\frac{m_6}{h})\cdot(hm_7) = m_4 \cdot m_5 \cdot m_6 \cdot m_7 = \prod_{j=1}^{4} m_j\ ,$$

implying that $DA(S)$ is true and $DA(S_i)$ is true, for all i, $1 \le i \le 3$. Also, $\{(m_4', sig_4'), (m_7', sig_7')\}$ generates dirty value h^2, and $\{(m_5', sig_5'), (m_6', sig_6')\}$ generates dirty value $\frac{1}{h^2}$. Hence, the product of dirty values is 1. Thus, $\{(m_4', sig_4'), (m_5', sig_5'), (m_6', sig_6'), (m_7', sig_7'),\}$ passes the MDA verification test and forms a SCE-set. Furthermore, $\{(m_4', sig_4'), (m_7', sig_7'),\}$ and $\{(m_5', sig_5'), (m_6', sig_6'),\}$ form two MCE-s-sets.

Theorem 6 will illustrate that given $S = \{(m_i, sig_i): i = 1, 2, \ldots, n\}$, when the number of the bad signatures r is larger than 4, the escape probability of the MDA on S, denoted by p, can be derived.

Theorem 6. If $S = \{(m_i, sig_i): i = 1, 2, \ldots, n\}$ has r (≥ 4) bad signatures, and S passes the conventional batch verification scheme, then the r signatures can be decomposed into q SCE-sets with $n_1 + n_2 + \ldots + n_q = r$ where n_j is the number of bad signatures in the jth SCE-set, $1 \le q \le \left\lfloor \frac{r}{4} \right\rfloor$, $4 \le n_j \le r$, $1 \le j \le q$. Then $p = \frac{\prod_{j=1}^{q} t_{r,j}}{n(n-1)(n-2)\ldots(n-r+1)}$ in which $t_{r,j}$ is the number of possible arrangements of the n_j bad signatures.

Proof: Let $S' = \{(m_j', sig_j') : j = 1, 2, \ldots, r\}$ be the collection of r bad signatures in S. If S passes the conventional batch verification scheme, then Equation (7) holds for DSA-type and Equation (2) holds for RSA-type. Equation (7) can be rewritten as

$$\sum_{\text{all pure signatures}} K'+\sum_{\text{all bad signatures}} K'$$

$$\equiv ((\sum_{\text{all pure signatures}} k''+\sum_{\text{all bad signatures}} k'') \bmod q) \bmod p^{,}$$

However $\sum_{\text{all pure signatures}} k' \equiv (\sum_{\text{all pure signatures}} k'' \bmod q) \bmod p_{,}$

then it becomes

$$\sum_{\text{all bad signatures}} k' \equiv (\sum_{\text{all bad signatures}} k'' \bmod q) \bmod p_{,}$$

and can be rewritten as $\sum_{j=1}^{r} k'_j \equiv (\sum_{j=1}^{r} k''_j \bmod q)(\bmod p)$,

$$\begin{aligned} &k'_j=\log_g r'_j, k''_j = s'^{-1}_j(m'_j+xr'_j) \text{ and} \\ &x=\log_g y, \text{ for j} = 1,2,\ldots,r. \end{aligned} \tag{14}$$

Similarly, Equation (2) can be rewritten to

$$\left(\prod_{j=1}^{r} s'_j\right)^e \equiv \prod_{j=1}^{r} m'_j (\bmod N) \tag{15}$$

Equations (14) and (15) show that the r bad signatures are tightly bounded and need to pass the k sampling tests, $2^{k-1} \le n \le 2^k - 1$, i.e., for DSA-type S, Equation (8) holds for $i = 1, 2, \ldots, k$, and for RSA-type, Equation (9) holds for $i = 1, 2, \ldots, k$. Theorems 2-5 show that a set of bad signatures S' in S cannot pass the MDA verification test when $|S'| \le 3$.Therefore, the necessary condition for S to pass the MDA verification test is $|S'| \ge 4$. Namely, S' can be decomposed into SCE-sets, e.g., S"s, with $DA(S")$ is true for all S", and S can pass the k *sampling* tests, implying the number of elements in each SCE-set is greater than or equal to 4. In other words, S' can be partitioned into q SCE-sets such that $n_1 + n_2 + \ldots + n_q = r$, $4 \le n_j \le r$, $1 \le j \le q$, $1 \le q \le \left\lfloor \frac{r}{4} \right\rfloor$. Then, the escape probabilities of n_1, n_2, n_3, … and n_q bad signatures are $\frac{t_{r,1}}{P^n_{n_1}}$, $\frac{t_{r,2}}{P^{n-n_1}_{n_2}}$, $\frac{t_{r,3}}{P^{n-n_1-n_2}_{n_3}}$, ……, and $\frac{t_{r,q}}{P^{n-n_1-n_2-n_3-\ldots-n_{q-1}}_{n_q}}$, respectively. Hence $P=\frac{t_{r,1}}{P^n_{n_1}}\cdot\frac{t_{r,2}}{P^{n-n_1}_{n_2}}\cdot\frac{t_{r,3}}{P^{n-n_1-n_2}_{n_3}}\cdots\frac{t_{r,q}}{P^{n-n_1-n_2-n_3-\ldots-n_{q-1}}_{n_q}} = \frac{\prod_{j=1}^{q} t_{r,j}}{n(n-1)(n-2)\ldots(n-r+1)}$. Q.E.D.#

By Theorem 7, $t = \prod_{j=1}^{q} t_{r,j}$ is the number of the possible distributions of the r bad signatures passing the MDA verification test. If $n_{j_1} = n_{j_2}$ and $j_1 < j_2$ then

$$t_{r,j_1} > t_{r,j_2}. \tag{16}$$

When $j_1 < j_2$, n_{j_1} bad signatures are arranged among the S before n_{j_2} bad signatures. Hence, the j_1th SCE-set has many more possible positions to place its n_{j_1} bad signatures than n_{j_2} SCE-set does. Let $P_{\max}$ is the maximum escape probability of an n-batch S, and let $P_{r,\max}$ be the maximum escape probability of $r(1 \le r \le n)$ bad signatures in S. The following theorem shows that $P_{\max}$ of the MDA occurs at $r = 4$, i.e., $P_{\max} = P_{4,\max}$.

Theorem 7. If an n-batch $S = \{(m_i, sig_i)$: $i = 1, 2, \ldots, n\}$ has r (≥ 4) bad signatures, then $P_{\max}$ of the MDA is $P_{4,\max}=\frac{t_{4,1}}{n(n-1)(n-2)(n-3)}$.

Proof: By Theorems 2-5, $P_{r,\max} = 0$ when $r \le 3$. So only $r \ge 4$ are considered. When $r = 4$, $n_1 = 4$ is the only SCE-set in S. By Theorem 6, $P_{4,\max}=\frac{t_{4,1}}{n(n-1)(n-2)(n-3)}$.

Similarly, When $r = 5$, $n_1 = 5$ is the only SCE-set in S and $P_{5,\max}=\frac{t_{5,1}}{n(n-1)......(n-4)}$.

When $r = 6$, $n_1 = 6$ is the only SCE-set in S and $P_{6,\max}=\frac{t_{6,1}}{n(n-1)......(n-5)}$.

When $r = 7$, $n_1 = 7$ is the only SCE-set in S and $P_{7,\max}=\frac{t_{7,1}}{n(n-1)......(n-6)}$.

When $r = 8$, $n_1 = 4$ and $n_2 = 4$ are the SCE-sets in S and $P_{8,\max}=\frac{t_{8,1}t_{8,2}}{n(n-1)(n-1)\ldots(n-7)}$, ….

Then, $P_{r,\max}$ can be derived where

$$P_{r,\max}=\frac{\prod_{j=1}^{q} t_{r,j}}{n(n-1)\ \ldots\ (n-r+1)}, 4 \le \text{r} \le \text{n}.$$

Furthermore, $\frac{P_{4,\max}}{P_{5,\max}}=\frac{t_{4,1}(n-4)}{t_{5,1}}>1$, $\frac{P_{4,\max}}{P_{6,\max}}=\frac{t_{4,1}(n-4)(n-5)}{t_{6,1}}>1$, $\frac{P_{4,\max}}{P_{7,\max}}=\frac{t_{4,1}(n-4)(n-5)(n-6)}{t_{7,1}}>1$ and $\frac{P_{4,\max}}{P_{8,\max}}=\frac{(n-4)(n-5)(n-6)(n-7)}{t_{8,2}}>1$, for $t_{4,1} = t_{8,1} > t_{8,2}$. We can further derive that $\frac{P_{4,\max}}{P_{r,\max}}=\frac{t_{4,1}(n-4)(n-5)\ldots(n-r+1)}{\prod_{j=1}^{q} t_{r,j}}\ge 1$, for all r and $4 \le r \le n$. Hence $P_{\max} = P_{4,\max}$. QED#

4 Security and Efficiency Analyses

In the following, we analyze the security and efficiency of the MDA and compare them with those of the SET.

4.1 Security

[8-10] claimed that an n-batch S with $r(r \geq 1)$ bad signatures might pass the verification of a conventional DSA-type or RSA-type batch verification scheme. The SET was then proposed to test the bad signatures. However, the escape probability that bad signatures can pass the SET is $p = \frac{1}{2^l}$ [7] where l is the length of a random number.

When $1 \leq r \leq 3$, the bad signatures may be or may not be discovered by the SET, however, they can be discovered by the MDA verification tests. For $r \geq 4$, the bad signatures may pass the SET and MDA verification tests. By Theorems 6-7, the maximum escape probability of the MDA verification test occurs at $r = 4$, $P_{4,\max} \geq P_{r,\max}$, as $r \geq 4$, $P_{r,\max} = 0$, as $r \leq 3$ and $P_{\max} = P_{4,\max} = \frac{t_{4,1}}{n(n-1)(n-2)(n-3)}$, showing that when $r \geq 4$, the more the bad signatures in an n-batch, the lower the escape probability of the MDA verification test. Also, a larger n will result in a lower escape probability.

The security levels of the SET and the MDA verification tests on different cases are listed in Table 1, in which level A, the highest, is defined as the level on which all the bad signatures will be discovered. In levels B, C and D, some bad signatures may not be discovered in the situation where level C is more secure than level D. When $l = k + 4$, the MDA and the SET are of the same security level, i.e., level B. Now we can conclude that the MDA is more secure than the SET.

Standing on users' viewpoint, a batch verification scheme is suitable for use in a close environment, i.e., the sending side S and receiving side R are strongly related to each other in business or responsibilities. In a RSA-type (DSA-type) batch verification scheme, only people work for both sides, i.e., S and R, can determine the parameters, N, e and d (p, q, g, x and y) which are used to generate a signature s_i m_i^d mod N for document/message mi's. Without these parameters, hackers cannot generate signatures for messages/documents. Even though hackers generate a faked massage and its signature, they cannot pass the MDA test. Namely, we can conclude that a batch of documents passing the RSA-type MDA test means the (m_i, r_i) is accurate, $1 \leq i \leq n$, and (m_i, r_i) is safely delivered from S to R. A batch of signatures that passes the DSA-type MDA test implies the similar conclusion.

Table 1 The Security Levels of the SET and MDA Verification Tests

	$r \leq 3$	$r \geq 4$		
		$l < (k+4)$	$l \approx (k+4)$	$l > (k+4)$
SET	D	D	B	C
MDA	A	C	B	D

Note. l: the length of a random variable.

4.2 Efficiency

When a checking matrix H has k rows, the possible numbers of columns of H range from 2^{k-1} to $2^k - 1$, i.e., $2^{k-1} \leq n \leq 2^k - 1$. Theorem 9 will show that the number of 1's in H, denoted by Z, ranges between $(k - 1) \times 2^{k-2} + 1$ and $k \times 2^{k-1}$ where Z is also the times that signatures will be tested.

Theorem 8. Given a checking matrix $H_{k \times n}$ where $2^{k-1} \leq n \leq 2^k - 1$, then, $(k - 1) \times 2^{k-2} + 1 \leq Z \leq k \times 2^{k-1}$.

Proof: When $n = 2^k - 1$, each row of H has at most 2^{k-1} entries, the values of which are 1. If H has k rows, the number of 1's in a $k \times (2^k - 1)$ checking matrix H is at most $k \times 2^{k-1}$. Let H' be a $(k - 1) \times (2^{k-1} - 1)$ checking matrix. The number of 1's in H' is $(k - 1) \times 2^{(k-1)-1} = (k - 1) \times 2^{k-2}$. Furthermore, a $k \times 2^{k-1}$ checking matrix can be obtained by adding a nonzero column to a $(k - 1) \times (2^{k-1} - 1)$ checking matrix. Hence the number of 1's contained in a $k \times 2^{k-1}$ checking matrix = (the number of 1's contained in the $(k - 1) \times (2^{k-1} - 1)$ checking matrix) + (the number of 1's in the added nonzero column) $\geq (k - 1) \times 2^{k-2} + 1$. Thus, the number of 1's in a $k \times n$ checking matrix will range from $(k - 1) \times 2^{k-2} + 1$ to $k \times 2^{k-1}$, i.e., $(k - 1) \times 2^{k-2} + 1 \leq Z \leq k \times 2^{k-1}$. Q.E.D.#

Given an n-batch S with $2^{k-1} \leq n \leq 2^k - 1$, by Theorem 8, each row of a $k \times (2^k - 1)$ checking matrix H has 2^{k-1} 1's, and the MDA needs k (= $\log_2 (n + 1)$) times of test, one row at a time. Each time 2^{k-1} signatures are verified, i.e., a total of $k \cdot 2^{k-1} = \frac{n+1}{2} \cdot \log_2(n+1) \approx \frac{n}{2} \cdot \log_2 n$ signatures are verified.

Table 2 summarizes the time complexities of DSA-type and RSA-type when the SET and MDA are individually employed. The time complexities of the DSA-type batch verification with the SET and the RSA batch verification with the SET are shown in [7]. Let T_{mul} (T_{exp}) be the cost of a modular multiplication (an exponentiation) operation. Our test environment includes x86 PC, P4 2G CPU, 1G DDR RAM, Multi-precision Integer and Rational Arithmetic C/C++ Library (MIRACL), MSVC Compiler, and MS Windows XP OS. According to our experimental results $T_{exp} \approx 30\ T_{mul}$, rather than $T_{exp} \approx 240\ T_{mul}$ presented in [14]. We adopted our results to perform the following experiment. The computation times required by the SET and the MDA are listed in Table 3, from which we can conclude that the MDA is more efficiently than the SET when $l \geq 30$ and $n \geq 512$.

Table 2 Time Complexities of DSA-type and RSA-type When the SET and MDA are Individually Employed

Methods	Time Complexity
DSA-type with the MDA	$\left[\left(\frac{3}{2}n-2\right)\log_2 n+3n-2\right] T_{mul} + (2\log_2 n + 2)\ T_{exp}$
DSA-type with the SET [7]	$(l+\frac{nl}{2}+4n+1)\ T_{mul} + 2\ T_{exp}$
RSA-type with the MDA	$[(n-2)\log_2 n + 2n - 2)]\ T_{mul} + (\log_2 n + 1)\ T_{exp}$
RSA-type with the SET [7]	$(2l + nl)\ T_{mul} + T_{exp}$

Note. n: number of signatures given; l: number of bits of a random number used by the SET; T_{mul}: computation time of a modular multiplication; T_{exp}: computation time of a modular exponentiation.

5 Conclusions and Future Work

The possible applications of the MDA include (1) to effectively protect the homeland of a country, when many foreigners, e.g., a tour team, would like to enter the country, they often submit their entrance applications to the immigration office of the country via a foreign travel agency; (2) A credit card company sends the transactions that a bank's customers submitted to purchase something. The receiving bank can verify the transactions as a whole with the MDA; (3) other examples can be delivering secret documents between two military units, between two government offices, etc.

Compared with the MDA, the SET has two weaknesses. The first is that the SET cannot correctly discover one, two and three bad signatures in an n-batch S so that attackers may penetrate the SET's signature verification system. The second is that the SET ignores the fact that the more the bad signatures in S, the lower the escape probability. The escape probability p of the SET is $p = \frac{1}{2^l}$ [7], which is independent from the number of bad signatures in S where l is the secure parameter used in the SET.

However, Theorems 2-5 show that one, two, and three bad signatures in S can be discovered by the MDA. Theorem 7 illustrates that when the number of digital signatures increases, the escape probability p of the MDA is lower and the maximum escape probability of the MDA occurs at r = 4. Theorem 6 also depicts that when the number of bad signatures r or the number of signatures n in S increases, p is lower, i.e., the MDA is more secure than the SET. Also, by Theorem 8 and according to Tables 2 and 3, the MDA's time complexity is lower than that of the SET.

In the future, we will try to develop a method to discover all bad signatures contained in an n-batch. Furthermore, if the number of bad signatures in S is rare, we would like to find an efficient method which can not only discover bad signatures, but also point out which signatures are bad [15-16]. We would also like to derive the reliability model for the MDA so that users can predict the reliability of the algorithm before using it. Those constitute our future research.

References

[1] Yu-Li Huang and Fang-Yie Leu, *Constructing a Secure Point-to-Point Wireless Environment by Integrating Diffie-Hellman PKDS RSA and Stream Ciphering for Users Known to Each Other*, *J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol.2, No.3, 2011, pp.96-107.

[2] Fuw-Yi Yang, Zhen-Wei Liu and Su-Hui Chiu, *Mobile Banking Payment System*, *J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol.2, No.3, 2011, pp.85-95.

Table 3 The Times Consumed When the SET and MDA are Individually Employed

Method Param.	DSA-type with the MDA	DSA-type with the SET	RSA-type with the MDA	RSA-type with the SET
l = 30, n = 512, T_{exp}= 30 T_{mul}	9,028 T_{mul}	9,819 T_{mul}	5,400 T_{mul}	15,990 T_{mul}
l = 30, n = 1,024, T_{exp} = 30 T_{mul}	19,070 T_{mul}	19,547 T_{mul}	11,572 T_{mul}	30,810 T_{mul}
l = 60, n = 512, T_{exp} = 30 T_{mul}	9,028 T_{mul}	17,529 T_{mul}	5,400 T_{mul}	30,870 T_{mul}
l = 60, n = 1,024, T_{exp} = 30 T_{mul}	19,070 T_{mul}	34,937 T_{mu}	11,572 T_{mul}	61,590 T_{mul}

[3] David Naccache, Serge Vaudenay, Dan Raphaeli and École Normale Supérieure, *Can DSA Be Improved: Complexity Trade-Offs with the Digital Signature Standard, Proc. of EUROCRYPT 1994*, Perugia, Italy, May, 1995, pp.77-85.

[4] Nasrollah Pakniat and Ziba Eslami, *A Proxy E-Raffle Protocol Based on Proxy Signatures, J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol.2, No.3, 2011, pp.74-84.

[5] Sumit Kumar Pandey and Rana Barua, *Efficient Construction of Identity Based Signcryption Schemes from Identity Based Encryption and Signature Schemes, J. Internet Services and Information Security*, Vol.1, No.2-3, 2011, pp.161-180.

[6] Turgay Korkmaz and Suleyman Tek, *Analyzing Response Time of Batch Signing, J. Internet Services and Information Security*, Vol.1, No.1, 2011, pp.70-85.

[7] Mihir Bellare, Juan A. Garay and Tal Rabin, *Fast Batch Verification for Modular Exponentiation and Digital Signatures, Proc. of EUROCRYPT 1998*, Espoo, Finland, May, 1998, pp.236-250.

[8] Chae Hoon Lim and Pil Joong Lee, *Security of Interactive DSA Batch Verification, Electronics Letters*, Vol.30, No.19, 1994, pp.1592-1593.

[9] Colin Boyd and Chris Pavlovski, *Attacking and Repairing Batch Verification Schemes, Proc. of ASIACRYPT 2000*, Kyoto, Japan, December, 2000, pp.58-71.

[10] Min-Shiang Hwang, Cheng-Chi Lee and Yuan-Liang Tang, *Two Simple Batch Verifying Multiple Digital Signatures, Proc. of International Conference on Information and Communications Security*, Xian, China, November, 2001, pp.233-237.

[11] National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, 2009, http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

[12] Lein Harn, *Batch Verifying Multiple RSA-type Digital Signatures, Electronics Letters*, Vol.34, No.12, 1998, pp.1219-1220.

[13] Tacer Elgamal, *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transactions on Information Theory*, Vol.31, No.4, 1985, pp.10-18.

[14] Neal Koblitz, Alfred Menezes and Scott Vanstone, *The State of Elliptic Curve Cryptography, Design, Codes and Cryptography*, Vol.19, No.2-3, 2000, pp. 173-193.

[15] Jarosław Pastuszak, Josef Pieprzyk and Jennifer Seberry, *Codes Identifying Bad Signatures in Batches, Proc. of INDOCRYPT 2000*, Calcutta, India, December, 2000, pp.143-154.

[16] Jaroslaw Pastuszak, Dariusz Michatek, Josef Pieprzyk and Jennifer Seberry, *Identification of Bad Signatures in Batches, Proc. of the International Workshop on Practice and Theory in Public Key Cryptography*, Victoria, Australia, January, 2000, pp.28-45.

Biographies

Yi-Li Huang received his master degree from National Central University of Physics, Taiwan, in 1983. His research interests include security of network and wireless communication, solar active-tracking system, pseudo random number generator design and file protection theory. He is currently a senior instructor of Tunghai University, Taiwan, and director of information security and grey theory laboratory of the University.

Chu-Hsing Lin received his PhD degree in computer sciences from National Tsing Hua University, Taiwan. Now he is a faculty at the Computer Science Department, Tunghai University. Professor Lin has ever been the Director of Computer Center from 1995 to 1999, and the Chair of the CS Department from 2004 to 2007. He has also been one of the Board Directors of the Chinese Information Security Association (CCISA) from 2001 till now. Dr. Lin has published over 150 papers in academic journals and international conferences. He was granted over twenty research projects from government departments and private companies in recent years. In 2006 and 2008, he was awarded the Outstanding Instructor Award of Master & PhD Thesis, repectively, by the IICM (Institute of Information & Computing Machinery). His current research interests include multimedia information security, wireless ad hoc networks, embedded systems applications.

Fang-Yie Leu received his BS, master and PhD degrees all from National Taiwan University of Science and Technology, Taiwan, in 1983, 1986 and 1991, respectively, and another master degree from Knowledge System Institute, USA, in 1990. His research interests include wireless communication, network security, Grid applications and Chinese natural language processing. He was invited to be a guest editor of *Mobile Information Systems Journal, Journal of Ambient Intelligence and Humanized Computing and Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. He is currently a workshop organizer of CWECS and MCNCS workshops, one of the program committee members of at least 10 international conferences, a full professor of TungHai University, Taiwan, and a director of database and network security laboratory of the University. He is also a member of IEEE Computer Society.