*International Journal of Innovative Computing, Information and Control*
Volume **x**, Number **0x**, x **2008**

ICIC International ©2010 ISSN 1349-4198
pp. **0–0**

# A BATCH VERIFYING AND DETECTING THE ILLEGAL SIGNATURES

Chun-Ta Li

Department of Information Management
Tainan University of Technology
529 Jhong Jheng Road, Yongkang, Tainan, Taiwan 710, R.O.C.
th0040@mail.tut.edu.tw

Min-Shiang Hwang[1]

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

Shih-Ming Chen

Department of Information Management
Chaoyang University of Technology
168 Jifong E. Road, Taichung County, Taiwan 413, R.O.C.

Abstract. *The concept of batch verifying multiple RSA digital signatures is to find a method that multiple digital signatures can be verified simultaneously in only one exponential operation time. In this article, we proposed a new batch verifying multiple RSA digital signatures scheme. The main contribution of the proposed scheme is that it can easily discover where the signature-verification fault is located without re-verifying all individual signatures separately.*

**Keywords:** Digital signature, information security, multiple signatures, PKI, RSA.

## 1. Introduction.

In 1978, Rivest, Shamir, and Adleman proposed a famous asymmetric cryptosystem named RSA [26] which included four main characteristics: user authentication, confidentiality, integrity, and non-repudiation. It can protect the transaction information which can be safely transmitted and avoid the problems of tampering or usurped the information over the network [12, 13, 14, 15]. In addition, it also solved the requirement of user authentication and communication security on networking environments [16, 17, 18, 19, 20, 21, 23, 24].

RSA utilized two different keys to perform encryption and decryption, the public key($e$) and the private key($d$), respectively. In RSA signature mechanism, both the signer and receiver have the private key and public key of itself own [6, 7, 11, 16, 17, 27, 28, 29]. First of all, the signer used personal private key to sign documents $M_i$(where $i = 1$ to $t$) and generated $t$ signatures when the signing process is completed. Then, the signer transmitted $t$ documents and signatures $S_i$(where $i = 1$ to $t$ and $S_i = M_i^d$) to receiver. After receiving these documents and signatures, the receiver used signer's public key to verify each of these $t$ signatures one by one and checks whether $M_i \stackrel{?}{=} S_i^e$ holds or not. During authentication and verification phase, it will reduce the computer host's processing ability because it needs to consume a large amount of exponential computation time. Therefore, the concept of batch verifying multiple RSA

[1]Responsible for correspondence: Prof. Min-Shiang Hwang.

digital signatures is to find a method that can efficiently improve the performance of verifying multiple RSA digital signatures.

In 1998, Harn proposed two batch verifying multiple DSA-type and RSA digital signature schemes [4, 5] which the multiple signatures can be signed by the same private key and these multiple signatures can be verified simultaneously in only one exponential operation time. So, it's more efficient than separate verified signature schemes which the signer must repeatedly verify each of these multiple signatures [1, 2, 10]. However, Harn's scheme has some weaknesses proposed by Hwang et al. [8, 9]. In 2002, Changchien and Hwang proposed a batch verifying scheme [3] to detect and identify forged multiple signatures [25]. In Harn's scheme, the verifier could not detect where the signature-verification fault was located if the batch verification fails. As a result, it is inefficient that the verifier must re-verify each of the signatures and then confirms where the signature-verification fault is located. Motivated by Harn's weaknesses, we proposed a matrix-based solution to quickly find out where the signature-verification faults are located without re-verifying each of the signatures.

In this article, the detailed explanation of Harn's scheme [5] and its weaknesses [8] will be found in Section 2. Section 3 will propose our new scheme to overcome Harn's weaknesses. Section 4 will analyze and show the performance results of three schemes including: RSA, Harn's scheme and ours. Finally, we make our conclusion in Section 5.

2. **Analysis of Harn's Scheme.** Before reviewing Harn's scheme, we reviewed the RSA digital signature scheme first of all. There are two prime numbers, $p$ and $q$ generated and a modulus $N$ which is obtained by multiplying $p$ and $q$ together. In addition, $e$ and $d$ are represented as the signer's public key and private key, respectively. If signer Alice wants to transmit the message $M$ to receiver Bob, she must generate a digital signature $S$ by using the formula $S = h(M)^d \bmod N$, where $h(\cdot)$ represents a public one-way hashing function. Next, when Bob receiving $(M, S)$ from Alice, he verifies the signature $S$ by using the formula $h(M) = S^e \bmod N$. As a result, suppose there are $t$ documents and $t$ digital signatures, it needs $t$ times signing process and verification process for signer and verifier, respectively. Therefore, it is inefficient if there are multiple digital signatures and Harn proposed an efficient batch verifying multiple RSA digital signatures scheme [5] which can be verified simultaneously in only one exponential operation time.

In Harn's scheme, suppose Alice wants to transmit the messages $M_1$, $M_2$, $\cdots$, $M_t$ and signatures $S_1, S_2, \cdots, S_t$ to Bob, all of these signatures $S_i = h(M_i)^d \bmod N$, $(i = 1, 2, \cdots, t)$ would be generated by using Alice's private key $d$. Then, Bob uses Alice's public key $e$ to verify messages $M_1$, $M_2$, $\cdots$, $M_t$ and its corresponding signatures $S_1, S_2, \cdots, S_t$ by using the formula shown as follows:

$$(\prod_{i=1}^{t} S_i)^e = \prod_{i=1}^{t} h(M_i) \bmod N. \tag{1}$$

If Equation (1) holds, it means signatures $S_1, S_2, \cdots, S_t$ are valid generated from Alice's messages $M_1, M_2, \cdots, M_t$. Furthermore, in Equation (1), these multiple signatures can be verified simultaneously in one exponential operation time.

2.1. **The Advantage of Harn's Scheme.** In conventional PKI mechanism, the signer signs messages $M_1, M_2, \cdots, M_t$ with personal private key and transmits corresponding messages $M_1$, $M_2$, $\cdots$, $M_t$ and signatures $S_1, S_2, \cdots, S_t$ to the receiver one at a time. Then the receiver verifies the digital signatures with the signer's public key one by one. However, in Harn's scheme, it can be verified in only one exponential operation time and the proposed scheme is efficient for batch signature verification.

2.2. **The Weakness of Harn's Scheme.** At present, there are two weaknesses in Harn's scheme. In 2000, Hwang et al. showed that the Harn's scheme could not find out two kinds of attacks. In the first attack, we exchanged two messages and signatures at random, Harn's scheme was unable to detect the illegal signatures. For example, we assume the signer Alice sends messages and signatures to the receiver Bob. The messages and signatures are generated as follows: $S_i‘ = h(M_{f(i)})^d \bmod N$. $S_i‘$ means a false signature which is generated from signature $S_i‘$ and $f(\cdot)$ is a one-to-one mapping function (i.e., $f(i) = j$,

$i = 1, 2, \cdots, t$, $j = 1, 2, \cdots, t$). Owing to the equation $(\prod_{i=1}^{t} S_i`)^e = \prod_{i=1}^{t} h(M_{f(i)}) \bmod N$, Alice could easily make the false signature and Bob will believe the signature is legally generated from Alice. The following is an example to illustrate the first attack in brief.

We assume Alice is a dishonest signer and sends three messages $(M_1, S_1`)$, $(M_2, S_2`)$, and $(M_3, S_3`)$ to receiver Bob. First, Alice could forge these three signatures $S_1` = h(M_2)^d \bmod N$, $S_2` = h(M_3)^d \bmod N$ and $S_3` = h(M_1)^d \bmod N$, respectively. Then, Bob will verify these fake signatures $M_1$, $M_2$ and $M_3$ by the equation $(\prod_{i=1}^{t} S_i`)^e = \prod_{i=1}^{t} h(M_{f(i)}) \bmod N$ which was introduced previously. Consequently, Bob still makes the batch verification valid because of the commutative principle of multiplication. But actually, Alice could claim that she didn't transmit the messages to Bob by this equation, $h(M_i) \neq (S_i`)^e \bmod N$. Briefly verification process is shown as follows:

$$\begin{aligned}
(S_1` \times S_2` \times S_3`)^e &= [h(M_1)^d \times h(M_2)^d \times h(M_3)^d]^e \bmod N \\
&= ([h(M_2) \times h(M_3) \times h(M_1)]^d)^e \bmod N \\
&= h(M_1) \times h(M_2) \times h(M_3) \bmod N.
\end{aligned}$$

In the second attack, we multiplied the signature factor by the random factor and made the multiplication factor be one. If this is the case, it will satisfy Equation (1) in Harn's scheme. First, we assume Alice sends the messages $(M_1, S_1`)$, $(M_2, S_2`)$, and $(M_3, S_3`)$ to receiver Bob, where $S_i` = a_i \times S_i$, $i = 1, 2, \cdots, t$ and $\prod_{i=1}^{t} a_i = 1$. According to the equation shown as follows: $(\prod_{i=1}^{t} S_i`)^e = \prod_{i=1}^{t} h(M_i) \bmod N$, Alice could maliciously make the false signature to Bob and he still believes the signature is generated from Alice. The following is a simple example to illustrate the second attack.

First, we assume that Alice is a dishonest signer who forges the messages $S_1`$, $S_2`$, and $S_3`$, where $S_1` = 1/2S_1$, $S_2` = 1/4S_2$ and $S_3` = 8S_3$. Then, Alice sends messages $(M_1, S_1`)$, $(M_2, S_2`)$ and $(M_3, S_3`)$ to Bob and he would verify the messages $M_1$, $M_2$ and $M_3$ by the equation shown as follows: $(\prod_{i=1}^{t} S_i`)^e = \prod_{i=1}^{t} h(M_i) \bmod N$. Similarly, these fake signatures will pass the verification and Alice could claim that she didn't transmit the messages to Bob by using this equation, $h(M_i) \neq (S_i`)^e \bmod N$. The completely verification phase is shown as follows:

$$\begin{aligned}
(S_1` \times S_2` \times S_3`)^e &= (1/2S_1 \times 1/4S_2 \times 8S_3)^e \bmod N \\
&= (S_1 \times S_2 \times S_3)^e \bmod N \\
&= [h(M_1)^d \times h(M_2)^d \times h(M_3)^d]^e \bmod N \\
&= ([h(M_1) \times h(M_2) \times h(M_3)]^d)^e \bmod N \\
&= h(M_1) \times h(M_2) \times h(M_3) \bmod N.
\end{aligned}$$

2.3. **The Restrictions of Harn's Scheme.** In PKI mechanism, the signer utilized his/her private key to sign the message, and then transmits the digital signature to the receiver. Next, the receiver verified the digital signature by using signer's public key. So, the verifier could detect the illegal signature if the verification fails. However, in Harn's scheme, the verifier could not detect where the signature-verification fault was located if the batch verification fails. Therefore, it is inefficient that the verifier must re-verify each of the signatures and then confirms where the signature-verification fault is located.

According to the weaknesses of Harn's scheme, in Section 3, we will propose an improved scheme to resist Hwang et al.'s two attacks [8, 9]. Furthermore, our scheme could also detect where the signature-verification fault is located when the verification fails without re-verifying each of these signatures.

3. **The Proposed Scheme.** Now we present our improved scheme. First of all, when the verifier received the messages $(M_1, S_1)$, $(M_2, S_2)$, $\cdots$, $(M_t, S_t)$ from the signer, the verifier will generate an $m \times n$ matrix(where $m \times n \geq t$) and $t$ random numbers $r_i$, $i = 1, 2, \cdots, t$, where $r_i \in \{1, 2, \cdots, t\}$ and $r_i$ is one-to-one mapping relation. Table 1 is an example of $m \times n$-matrix. Next, the verifier would randomly fill these $t$ messages in the $m \times n$-matrix position by using the following equation:

$$\begin{cases} S(m,n) = S(\lceil r_i/n \rceil, n), & if\ r_i \bmod n = 0; \\ S(m,n) = S(\lceil r_i/n \rceil, r_i\ mod\ n), & otherwise. \end{cases} \tag{2}$$

Table 1. An $m \times n$ matrix

| | | | | |
|---|---|---|---|---|
| $S(1,1)$ | $S(1,2)$ | $\cdots$ | $S(1,n-1)$ | $S(1,n)$ |
| $S(2,1)$ | $S(2,2)$ | $\cdots$ | $S(2,n-1)$ | $S(2,n)$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $S(m-1,1)$ | $S(m-1,2)$ | $\cdots$ | $S(m-1,n-1)$ | $S(m-1,n)$ |
| $S(m,1)$ | $S(m,2)$ | $\cdots$ | $S(m,n-1)$ | $S(m,n)$ |

After filling these messages $(M_1, S_1)$, $(M_2, S_2)$, $\cdots$, $(M_t, S_t)$ in the $m \times n$-matrix, the verifier could batch verify each of the rows and the columns, respectively. The complete batch verifying process is divided into two verifications: row verification and column verification. Briefly steps of row/column verification are shown as follows.

1. Row Verification:
   First row: $(\prod_{i=1}^{n} S_{(1,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(1,i)}) \bmod N$
   Second row: $(\prod_{i=1}^{n} S_{(2,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(2,i)}) \bmod N$
   $\vdots$
   $(m-1)^{th}$ row: $(\prod_{i=1}^{n} S_{(m-1,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(m-1,i)}) \bmod N$
   $m^{th}$ row: $(\prod_{i=1}^{n} S_{(m,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(m,i)}) \bmod N$
2. Column Verification:
   First column: $(\prod_{i=1}^{m} S_{(i,1)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,1)}) \bmod N$
   Second column: $(\prod_{i=1}^{m} S_{(i,2)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,2)}) \bmod N$
   $\vdots$
   $(n-1)^{th}$ column: $(\prod_{i=1}^{m} S_{(i,n-1)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,n-1)}) \bmod N$
   $n^{th}$ column: $(\prod_{i=1}^{m} S_{(i,n)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,n)}) \bmod N$

If there are some signature-verification faults in the matrix, we could find out where these signature-verification faults are located by finding the matrix positions of row and column overlaps. We give a simple example to briefly illustrate our scheme as follows.

Suppose Alice sends 25 messages to Bob, then Bob will generate 25 random numbers and a $5 \times 5$ matrix shown as Table 2. If $r_1 = 16$, the message $(M_1, S_1)$ would be filling in the position $S(4,1)$ of matrix by using Equation (2). Similarly, if $r_2 = 3$, the message $(M_2, S_2)$ would be filling in the position $S(1,3)$ of matrix by using Equation (2), and the rest can be deduced by the similar way. After filling 25 messages in the matrix, Bob could batch verify each of the rows and columns, respectively. As mentioned above, the batch verifying process is divided into row/column verifications and does the following.

- Row Verification:
  First row: $(\prod_{i=1}^{n} S_{(1,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(1,i)}) \bmod N$
  Second row: $(\prod_{i=1}^{n} S_{(2,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(2,i)}) \bmod N$
  Third row: $(\prod_{i=1}^{n} S_{(3,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(3,i)}) \bmod N$
  Fourth row: $(\prod_{i=1}^{n} S_{(4,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(4,i)}) \bmod N$
  Fifth row: $(\prod_{i=1}^{n} S_{(5,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(5,i)}) \bmod N$
- Column Verification:
  First column: $(\prod_{i=1}^{m} S_{(i,1)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,1)}) \bmod N$
  Second column: $(\prod_{i=1}^{m} S_{(i,2)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,2)}) \bmod N$
  Third column: $(\prod_{i=1}^{m} S_{(i,3)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,3)}) \bmod N$
  Fourth column: $(\prod_{i=1}^{m} S_{(i,4)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,4)}) \bmod N$
  Fifth column: $(\prod_{i=1}^{m} S_{(i,5)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,5)}) \bmod N$

After batch verifying each of the rows and the columns, Bob is now confirmed that the signature-verification fault is really occurring or not. Suppose there was one signature-verification fault in the

Table 2. A $5 \times 5$ matrix

| | | | | |
|---|---|---|---|---|
| $S(1,1)$ | $S(1,2)$ | $S(1,3)$ | $S(1,4)$ | $S(1,5)$ |
| $S(2,1)$ | $S(2,2)$ | $S(2,3)$ | $S(2,4)$ | $S(2,5)$ |
| $S(3,1)$ | $S(3,2)$ | $S(3,3)$ | $S(3,4)$ | $S(3,5)$ |
| $S(4,1)$ | $S(4,2)$ | $S(4,3)$ | $S(4,4)$ | $S(4,5)$ |
| $S(5,1)$ | $S(5,2)$ | $S(5,3)$ | $S(5,4)$ | $S(5,5)$ |

Table 3. The size of Row and Column values chosen with 25 messages in our scheme

| Sizes of Row | Sizes of Column | computation time |
|---|---|---|
| 2 | 13 | 1.752 |
| 3 | 9 | 1.372 |
| 4 | 7 | 1.292 |
| 5 | 5 | 1.092 |

position $S(3,3)$ of matrix, Bob could easily realize there was a signature-verification fault occurring and precisely detects where the signature-verification fault is located. From the row/column verifications of the mentioned above, there would occur two verification fails and these two fails would occur in the third row and the third column, respectively. According to the verification fails of the third row and the third column overlaps, the signature-verification fault could be precisely detected in the position $S(3,3)$ of matrix. Therefore, our scheme could resist the first attack proposed by Hwang et al. In addition, our scheme filled the messages in $m \times n$ matrix at random that it could prevent the second attack proposed by Hwang et al. [8, 9].

## 4. **Implementation and Result Analysis.**

### 4.1. **Experiment Results.**

In order to clarify the proposed scheme, in this section, we implement the following three schemes: RSA, Harn's scheme, and ours. In our proposed scheme, the size of row and column values chosen is shown in Table 3. In addition, the performance results of RSA, Harn's scheme, and ours are shown in Table 4 in terms of computation time. From Table 3 shows, the performance of our scheme is related to the numbers of row and column selection. When this situation is achieved that both of the numbers of row and column square root are equal to the messages numbers, the performance of our scheme would be the best.

From Table 4 shows, the performance of Harn's scheme is better when the batch verifying multiple signatures has succeeded. Contrary to the previous situation, if there are signature-verification faults in Harn's scheme, it must re-verify all of these multiple signatures like RSA. Thus, the performance of Harn's scheme will be worse in comparison with RSA and our schemes. Although the performance of our scheme isn't the best, yet the signature-verification faults occur, the performance of RSA and Harn's schemes will far behind our scheme in terms of required computational time.

In comparison with required computation-time of batch multiple signatures verification of RSA and ous shown in Figure 1, the performance of our scheme is not always suitable for the batch multiple signatures verification, particularly less messages amount. According to the experiment result shows, in Figure 1, the performance of RSA batch verification is close to our scheme when the messages amount is 4.

### 4.2. **Analysis of Illegal Signatures Detection.**

From the proposed scheme mentioned in the previous section, the performance of illegal signatures detection is regarded as the position it is located in. The best condition is only one illegal signature is detected and we could detect where the signature-verification fault is located immediately. If not (two illegal signatures and upward are detected), the detection results

Table 4. Experiment results of RSA, Harn and our schemes

| Method | Messages | Time | Sizes of Row | Sizes of Column |
|---|---|---|---|---|
| RSA | 25 | 2.884 | — | — |
| Harn | 25 | 0.34 | — | — |
| Ours | 25 | 1.092 | 5 | 5 |
| RSA | 100 | 12.178 | — | — |
| Harn | 100 | 1.031 | — | — |
| Ours | 100 | 2.283 | 10 | 10 |
| RSA | 256 | 30.344 | — | — |
| Harn | 256 | 2.304 | — | — |
| Ours | 256 | 3.535 | 16 | 16 |

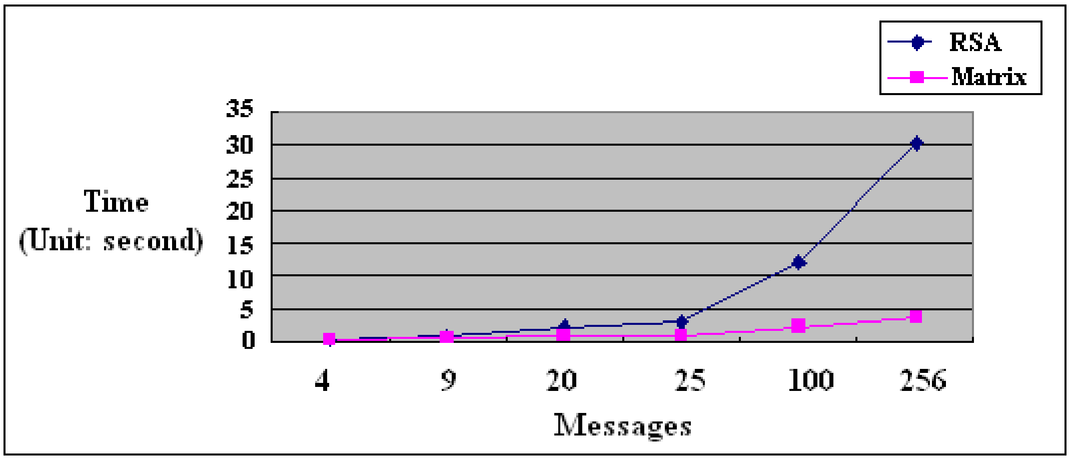


Figure 1. Comparison with RSA and our schemes

is regarded as the position it is located in. In the following, we demonstrated two conditions included: one illegal signature is detected and two illegal signatures are detected.

1. One Illegal Signature:
   From Figure 2 shows, the batch verification failure is occurring on the third row and third column. As a result, from the position of the overlapping of third row and third column, it confirms that the signature $S(3, 3)$ is illegal and the number of total verification times is 10(5 rows and 5 columns).
2. Two Illegal Signatures:
   At this situation, there are three consequences from the detection: two of the signature-verification faults are occurring on the same row or on the same column; two signature-verification faults are occurring on adjacent diagonal; two signature-verification faults are not occurring on the same row or not on the same column. Brief explanations are described as follows.
   - Two signature-verification faults are occurring on the same row or on the same column.
     In Figure 3, it could confirm that both the two signatures $S(3, 3)$ and $S(3, 4)$ are illegal immediately because the batch verification failures are occurring on the third row, third column and fourth column. So in this case, the number of total verification times is 10(5 rows and 5 columns).
   - Two signature-verification faults are occurring on adjacent diagonal.
     In Figure 4, if the failed batch verifications are occurring on the second row, third row, third column and fourth column. It must verify both the four signatures $S(2, 3), S(3, 3), S(2, 4)$ and

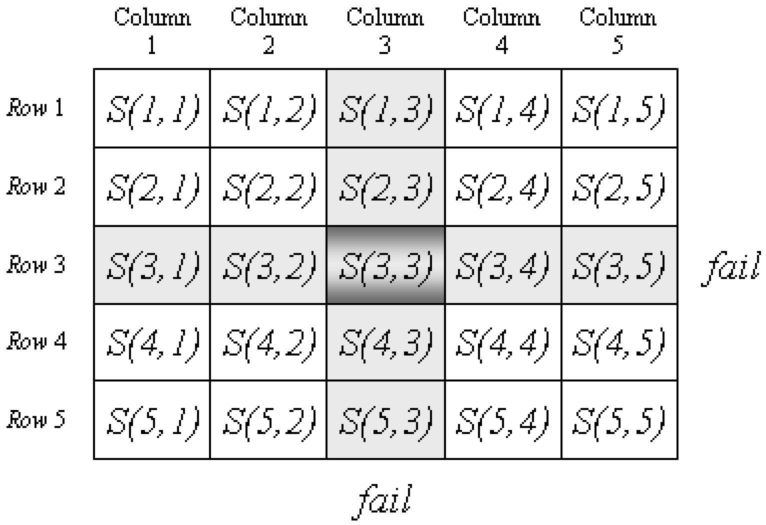


Figure 2. One illegal signature

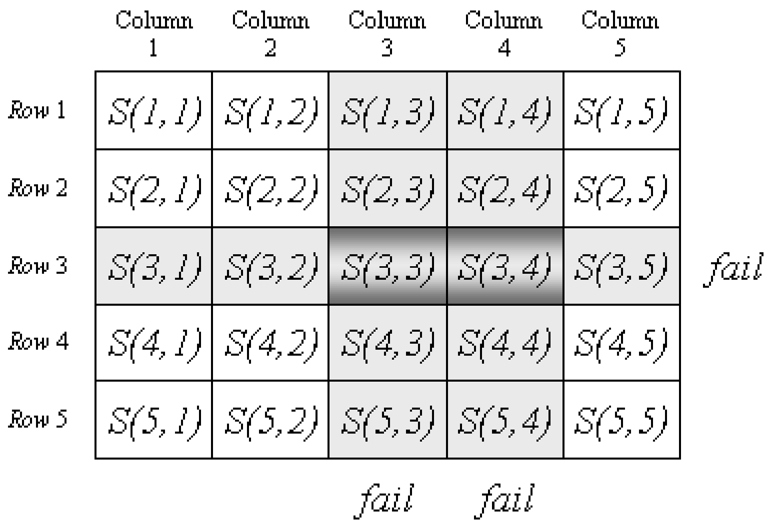


Figure 3. Two illegal signatures are occurring on the same row

$S(3,4)$ to confirm where the two faults are located. Finally, it confirms that the two signatures $S(3,3)$ and $S(2,4)$ are illegal and the number of total verification times is 14(5 rows, 5 columns and 4 signatures).

- Two signature-verification faults are not occurring on the same row or not on the same column. In Figure 5, the batch verification failures are occurring on the first row, third row, third column and fourth column. It must verify both the four signatures $S(1,3), S(3,3), S(3,4)$ and $S(1,4)$ to confirm where the two faults are located. Thus, it confirms that the two signatures $S(3,3)$ and $S(1,4)$ are illegal and the number of total verification times is 14 (5 rows, 5 columns, and 4 signatures).

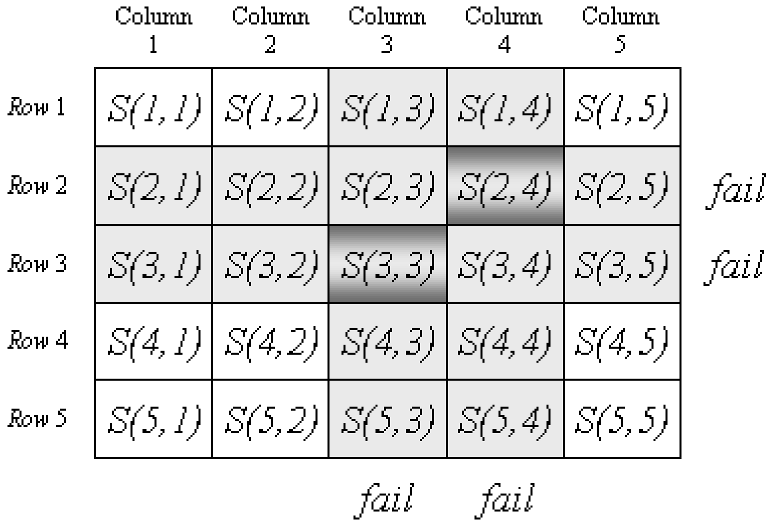


FIGURE 4. Two illegal signatures are occurring on adjacent diagonal

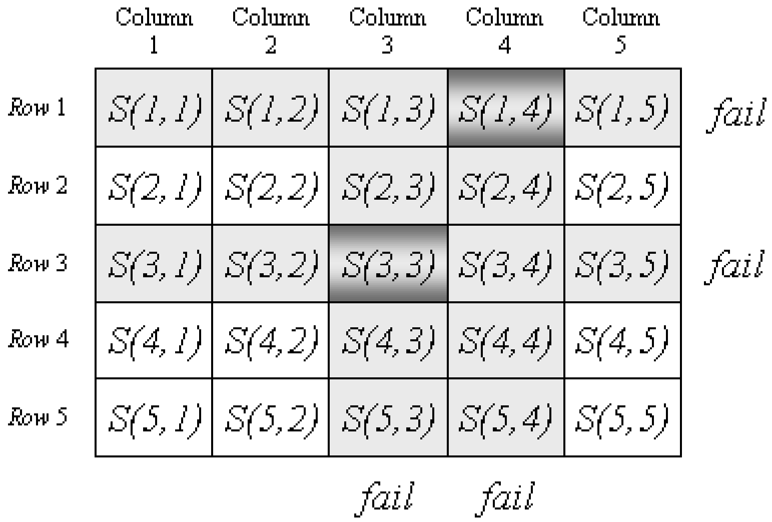


FIGURE 5. Two illegal signatures are not occurring on the same row or not occurring on the same column

In the above mentioned situations, the worst case of detecting the illegal signatures is that the two illegal signatures are occurring on adjacent diagonal and the two illegal signatures are not occurring on the same row or not on the same column. Both the number of total verification times of these two situations are 14. Although in this worst case, our scheme is still more efficient than Harn's scheme because their scheme must re-verify all individual signatures separately to confirm where the illegal signatures are located and the number of total verification times is 26(25 messages and 1 batch verification).

5. **Conclusions.** In this paper, we have introduced three authentication methods including RSA, Harn's, and our improved scheme. We also introduce the performance of these three schemes and some illegal signature detections in our scheme. According to the experiment result shows, the performance of Harn's

scheme is the best when there does not have any signature-verification faults and our scheme is not suitable for batch verification when the messages amount is small. However, there are two kinds of attacks that can cheat Harn's scheme and their scheme cannot find out where the signature-verification fault is located if the batch verification fails. Therefore, our improved scheme could resist the weakness of Harn's and quickly find out where the signature-verification fault is located.

## REFERENCES


[1] Chih-Ying Chen, Hsiu-Feng Lin, and Chin-Chen Chang. An Efficient Generalized Group-oriented Signature Scheme. *International Journal of Innovative Computing, Information and Control*, 4(6):1335–1346, 2008.

[2] Chia-Ho Chu, Hsiu-Feng Lin, Chin-Chen Chang, and Chih-Ying Chen. A Multi-policy Threshold Signature Scheme with Traceable Participant Cosigners. *International Journal of Innovative Computing, Information and Control*, 4(6):1347–1356, 2008.

[3] S. Wesley Changchien, Min-Shiang Hwang, and Kuo-Feng Hwang. A batch verifying and detecting multiple RSA digital signatures. *International Journal of Computational and Numerical Analysis and Applications*, 2(3):303–307, 2002.

[4] L. Harn. Batch verifying multiple DSA-type digital signatures. *Electronics Letters*, 34(9):870–871, 1998.

[5] L. Harn. Batch verifying multiple RSA digital signatures. *Electronics Letters*, 34(12):1219–1220, 1998.

[6] Hui-Feng Huang, Chao-Wen Chan, Chih-Hao Lin, and Hsin-Wei Wang. A low-computation conference key system for mobile communications. *International Journal of Innovative Computing, Information and Control*, 5(2):461–466, 2009.

[7] Min-Shiang Hwang and Cheng-Chi Lee. Research issues and challenges for multiple digital signatures. *International Journal of Network Security*, 1(1):1–7, 2005.

[8] Min-Shiang Hwang, Cheng-Chi Lee, and Eric Jui-Lin Lu. Cryptanalysis of the batch verifying multiple DSA-type digital signatures. *Pakistan Journal of Applied Sciences*, 1(3):287–288, 2001.

[9] Min-Shiang Hwang, Iuon-Chung Lin, and Kuo-Feng Hwang. Cryptanalysis of the batch verifying multiple RSA digital signatures. *Informatica*, 11(1):15–19, 2000.

[10] Min-Shiang Hwang, Shiang-Feng Tzeng and Shu-Fen Chiou. A Non-repudiable Multi-proxy Multi-signature Scheme. *ICIC Express Letters*, 3(3(A)):259–264, 2009.

[11] Maged Hamada Ibrahim. Resisting traitors in linkable democratic group signatures. *International Journal of Network Security*, 9(1):51–60, 2009.

[12] Chun-Ta Li, Min-Shiang Hwang, and Chi-Yu Liu. An electronic voting protocol with deniable authentication for mobile ad hoc networks. *Computer Communications*, 31(10):2534–2540, 2008.

[13] Chun-Ta Li, Min-Shiang Hwang, and Yen-Ping Chu. Further improvement on a novel privacy preserving authentication and access control scheme for pervasive computing environments. *Computer Communications*, 31(18):4255–4258, 2008.

[14] Chun-Ta Li, Min-Shiang Hwang, and Yen-Ping Chu. A secure and efficient communication scheme with authenticated key establishment and privacy preservation for vehicular ad hoc networks. *Computer Communications*, 31(12):2803–2814, 2008.

[15] Chun-Ta Li, Min-Shiang Hwang, and Yen-Ping Chu. Improving the security of a secure anonymous routing protocol with authenticated key exchange for ad hoc networks. *International Journal of Computer Systems Science and Engineering*, 23(3):227-234, 2008.

[16] Chun-Ta Li, Min-Shiang Hwang, and Yen-Ping Chu. An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks. *International Journal of Innovative Computing, Information and Control*, 5(8):2107–2124, 2009.

[17] Chun-Ta Li and Min-Shiang Hwang. An online biometrics-based secret sharing scheme for multiparty cryptosystem using smart cards. *International Journal of Innovative Computing, Information and Control*, article in press, 2009.

[18] Chun-Ta Li, C. H. Wei, and Y. H. Chin. A secure event update protocol for peer-to-peer massively multiplayer online games against masquerade attacks. *International Journal of Innovative Computing, Information and Control*, 5(12(A)):4715–4723, 2009.

[19] Chun-Ta Li. An efficient and secure communication scheme for trusted computing environments. *Journal of Computers*, 20(3):17–24, 2009.

[20] Chun-Ta Li and Yen-Ping Chu. Cryptanalysis of threshold password authentication against guessing attacks in ad hoc networks. *International Journal of Network Security*, 8(2):166–168, 2009.

[21] Chun-Ta Li and Min-Shiang Hwang. An efficient biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications*, 33(1):1–5, 2010.

[22] Chun-Ta Li, Cheng-Chi Lee and Lian-Jun Wang. A Two-Factor User Authentication Scheme Providing Mutual Authentication and Key Agreement over Insecure Channels. *Journal of Information Assurance and Security*, 5(1):201–208, 2010.

[23] Chun-Ta Li, Cheng-Chi Lee and Lian-Jun Wang. On the Security Enhancement of An Efficient and Secure Event Signature Protocol for P2P MMOGs. In *The 2010 International Conference on Computational Science and Its Applications*, pages 599-609, Lecture Notes in Computer Science, Vol. 6016, 2010.

[24] Chun-Ta Li, C. H. Wei, Cheng-Chi Lee, Y. H. Chin and Lian-Jun Wang. A Secure and Undeniable Billing Protocol among Charged Parties for Grid Computing Environments. *International Journal of Innovative Computing, Information and Control*, article in press, 2010.

[25] Rongxing Lu, Zhenfu Cao, and Jun Shao. On security of two nonrepudiable threshold multi-proxy multi-signature schemes with shared verification. *International Journal of Network Security*, 4(3):248–253, 2007.

[26] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[27] Martin Stanek. Attacking lccc batch verification of rsa signatures. *International Journal of Network Security*, 6(2):238–240, 2008.

[28] Tony Thomas and Arbind Kumar Lal. A zero-knowledge undeniable signature scheme in non-abelian group setting. *International Journal of Network Security*, 6(3):265–269, 2008.

[29] Baodian Wei, Fangguo Zhang, and Xiaofeng Chen. A new type of designated confirmer signatures for a group of individuals. *International Journal of Network Security*, 7(2):293–300, 2008.