Research on Batch Verification Schemes for Identifying Illegal Signatures

Hsieh-Tsen Pan[1], Eko Fajar Cahyadi[1,2], Shu-Fen Chiou[3], and Min-Shiang Hwang[1,4]
(Corresponding author: Min-Shiang Hwang)
Department of Computer Science & Information Engineering, Asia University, Taichung, Taiwan[1]
Department of Telecommunication Engineering, Institut Teknologi Telkom Purwokerto, Purwokerto, Indonesia[2]
Department of Information Management, National Taichung University of Science and Technology, Taiwan[3]
Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan[4]
(Email: mshwang@asia.edu.tw)
(Received Mar. 3, 2019; revised and accepted Oct. 1, 2019)

Abstract

Invalid signatures produced by some adversaries may pose a severe challenge to the recipient. Furthermore, identifying an invalid signature in a bunch of messages could be a complex and challenging task to do. Batch verification is an idea to simultaneously verify multiple digital signatures in just one exponential operation time. By this scheme, we can make a quick response, and improve the verification time. In terms of identifying illegal signatures in the batch of messages, we have surveyed several well-known papers that proposed different approaches. In this paper, we define four criteria for evaluating these schemes. It is followed by a detailing review and computation comparisons from all documents. Finally, we provide two issues for future works.

Keywords: Batch Verification Schemes; Digital Signature; Identifying Illegal Signatures

1 Introduction

A digital signature is a method for signing a transmitted electronic document so that the other parties can verify its contents and the sender's identity. Each signer has a pair of keys: a private key and a public key. The private key is kept being secret, while the public key is made public. A signer creates a digital signature using their private key, while a recipient can verify the digital signature by the signer's public key. No one can forge the signer's digital signature as the private key is safely guarded [6, 11, 18].

If a signer wants to generate and send t signatures, then the verifier needs to check for t signatures. It is inefficient since they should spend t times to validate t digital signatures using the traditional cryptosystems. The batch verification schemes were proposed to verify these multiple digital signatures by the signer's public key, which needs only one verification instead of t verifications [5, 7, 9, 13, 15, 17]. However, if the batch verification fails, all the individual signatures must be verified separately, which would become inefficient.

In this survey, we provide a relational approach of several representative publications that have a common interest in batch verification schemes for illegal signatures identification [1, 12, 14, 16]. A detailed overview of those researches is focused on their strengths and weaknesses.

To achieve effectivity and efficiency evaluation of identifying illegal signatures in a batch verification scheme, we propose the following four criteria.

1) Unforgeability: No one can forge the legitimate multiple digital signatures. It's the basic requirement of the batch verification scheme.

2) Efficient traceability: It should be able to determine illegal signatures efficiently. When the multiple digital signatures are forged, the verifier can identify them with minimum computational and communication costs.

3) Applicability: A generic illegal signature locating algorithm could be used in any type of batch verification scheme. Since the effort is reinvested over a larger number of applications, the single generic illegal signature locating algorithm may be optimized, verified, and otherwise improved.

4) Error locating auditability: It does not misjudge the legal signature as an illegal signature.

For a better understanding, the rest of this paper is organized as follows. Section 2 describes various batch verification schemes for identifying illegal signatures. In Section 3, we give an analysis and a comparison among these schemes. In Section 4, two issues for future works are proposed. Finally, the conclusion is explained in Section 5.

2 Related Works

Several batch verification schemes for identifying illegal signatures have been proposed [1, 12, 14, 16].

2.1 A GCD-based Batch Verification Scheme for Identifying Illegal Signatures

In 2002, Hwang, Lee, and Lai proposed a batch verification scheme for identifying illegal signatures [12]. Their scheme is based on the Greatest Common Divisor (GCD). If a signer Alice wants to transmit the message M to receiver Bob, she must generate a digital signature S by using two assumptions: One is that $\prod_{i=1}^{t} h(M_i) < n$, and the other is $h(M_i)$, must be a prime, where $h(.)$ represents a public one-way hashing function, and $i = 1, 2, \cdots, t$. Once the signer sends of t message and signature pairs $((M_i, S_i), i = 1, 2, \cdots, t)$, the verifier will perform the following procedures to authenticate the illegal signature.

Step 1: The verifier computes $A = \gcd[((\prod_{i=1}^{t} S_i)^e \bmod n), \prod_{i=1}^{t} h(M_i)]$.

Step 2: The verifier computes $B = (\prod_{i=1}^{t} S_i)^e \bmod n/A$. If $B = 1$, these signatures are legal. Otherwise, one or more signatures are illegal.

Step 3: If B is a prime, the message (M'_j, S'_j) is illegal. Otherwise, check the following

$$B \bmod M'_j \stackrel{?}{=} 0, j = 1, 2, \cdots, t.$$

If the above equation holds, the message (M'_j, S'_j) is illegal.

If there is only one illegal signature $(S'_j, j \in (1, 2, \cdots, t))$ hiding among the t signatures, Hwang-Lee-Lai's scheme is efficient. However, it needs $t-1$ modulus remainder operations to identify these illegal signatures in Hwang-Lee-Lai's scheme.

For example, suppose the signer sends 5 messages $((M_1, S_1), (M_2, S_2), (M_3, S_3), (M_4, S_4), (M_5, S_5))$ to the verifier. If there is one illegal signature (S'_4) among the five signatures, it would be identified as follows:

$$\begin{aligned} A &= \gcd[((\prod_{i=1}^{5} S_i)^e \bmod n), \prod_{i=1}^{5} h(M_i)] \\ &= h(M_1)h(M_2)h(M_3)h(M_5). \\ B &= (S_1S_2S_3S'_4S_5)^e \bmod n/A \\ &= \frac{h(M_1)h(M_2)h(M_3)h(M'_4)h(M_5)}{h(M_1)h(M_2)h(M_3)h(M_5)} \\ &= h(M'_4) \end{aligned}$$

Thus, the illegal message is M'_4. However, if there are two illegal signatures, S'_2 and S'_4, among the five, it will not be directly identified because of,

$$\begin{aligned} A &= \gcd[((\prod_{i=1}^{5} S_i)^e \bmod n), \prod_{i=1}^{5} h(M_i)] \\ &= h(M_1)h(M_3)h(M_5). \\ B &= (S_1S'_2S_3S'_4S_5)^e \bmod n/A \\ &= \frac{h(M_1)h(M'_2)h(M_3)h(M'_4)h(M_5)}{h(M_1)h(M_3)h(M_5)} \\ &= h(M'_2)h(M'_4). \end{aligned}$$

The verifier cannot directly identify the illegal signatures from the multiplication of $h(M'_2)h(M'_4)$. The verifier needs to identify invalid signatures as follows:

$$\begin{aligned} B \bmod M'_1 &\neq 0 \\ B \bmod M'_2 &= 0 \\ B \bmod M'_3 &\neq 0 \\ B \bmod M'_4 &= 0 \\ B \bmod M'_5 &\neq 0. \end{aligned}$$

From the above equations, the verifier identifies two illegal messages: $(M'_2, S'_2), (M'_4, S'_4)$.

2.2 A 2D-based Batch Verification Scheme for Identifying Illegal Signatures

In 2010, a 2D-based batch verification scheme for identifying illegal signatures was proposed by Li, Hwang, and Chen [14]. When the verifier receives 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 is the smallest integer which satisfies $m \times n \geq t$, where t is random numbers. The verifier performs the following procedures to verify the illegal signature.

Step 1: The verifier constructs an $m \times n$ matrix (see Table 1).

Table 1: An $m \times m$ matrix

S(1,1)	S(1,2)	...	S(1,m-1)	S(1,m)
S(2,1)	S(2,2)	...	S(2,m-1)	S(2,m)
⋮	⋮	⋮	⋮	⋮
S(m-1,1)	S(m-1,2)	...	S(m-1,m-1)	S(m-1,m)
S(m,1)	S(m,2)	...	S(m,m-1)	S(m,m)

Step 2: The verifier randomly selects and fills these t digital signatures in the $m \times n$ matrix.

Step 3: The verifier performs the batch verify each of the rows. The details of row verifications are computed as follows:

$$(\prod_{i=1}^{m} S_{(r,i)})^e \stackrel{?}{=} \prod_{i=1}^{m} h(M_{(r,i)}) \bmod N, \ r = 1, 2, \cdots, m).$$

Step 4: The verifier performs the batch verify each of the columns. The details of column verifications are computed as follows:

$$(\prod_{i=1}^{m} S_{(i,c)})^e \stackrel{?}{=} \prod_{i=1}^{m} h(M_{(i,c)}) \bmod N,\ c = 1, 2, \cdots, m).$$

Step 5: If there are some signature-verification faults in the matrix, the verifier could find out where these signature-verification faults are located by finding the matrix positions of row and column overlaps.

For example, suppose the signer sends 16 messages $((M_1, S_1), (M_2, S_2), \cdots, (M_{16}, S_{16}))$ to the verifier. The verifier will generate 36 signatures by random selections and fills these signatures in the 4×4 matrix (see Table 2).

Table 2: A 4×4 matrix

S(1,1)	S(1,2)	S(1,3)	S(1,4)
S(2,1)	S(2,2)	S(2,3)	S(2,4)
S(3,1)	S(3,2)	S(3,3)	S(3,4)
S(4,1)	S(4,2)	S(4,3)	S(4,4)

Assume there is one illegal signature in the position $S(2, 3)$ of matrix. There would occur two verification failures in the second row and the third column, respectively (see Table 3).

Table 3: A 4×4 matrix with one illegal signature

S(1,1)	S(1,2)	S(1,3)	S(1,4)	Pass
S(2,1)	S(2,2)	$S^*(2,3)$	S(2,4)	Fail
S(3,1)	S(3,2)	S(3,3)	S(3,4)	Pass
S(4,1)	S(4,2)	S(4,3)	S(4,4)	Pass
Pass	Pass	Fail	Pass	

According to the overlap of verification failures of the second row and the third column, the illegal signature could be precisely identified in the position $S(2, 3)$ of the matrix.

2.3 A BT-based Batch Verification Scheme for Identifying Illegal Signatures

In 2013, a binary tree-based (BT-based) batch verification scheme for identifying illegal signatures was proposed by Atanasiu [1]. When the verifier receives the messages $(M_1, S_1), (M_2, S_2), \cdots, (M_t, S_t)$ from the signer, the verifier will re-order these signatures by a total order relation and perform the following procedures to verify the illegal signature.

Step 1: The verifier re-orders these signatures by a total order relation: $(M'_1, S'_1), (M'_2, S'_2), \cdots, (M'_t, S'_t)$. Here, $(M'_1, S'_1) < (M'_2, S'_2) < \cdots < (M'_t, S'_t)$ are by the following rule:

$$(M'_i, S'_i) < (M'_j, S'_j) \iff (S'_i < S'_j) \vee [(S'_i = S'_j) \wedge (M'_i < M'_j)].$$

Step 2: The verifier performs one time of the batch verify with all t signatures:

$$(\prod_{i=1}^{t} S'_i)^e \stackrel{?}{=} \prod_{i=1}^{t} h(M'_i).$$

Step 3: If the above equation holds, all t signatures are legal. Otherwise, the verifier divides t signatures into two parts: $[(M'_1, S'_1), (M'_2, S'_2), \cdots, (M'_{t/2}, S'_{\lceil \frac{t}{2} \rceil})]$ and $[(M'_{\lceil \frac{t}{2}+1 \rceil}, S'_{\lceil \frac{t}{2}+1 \rceil}), (M'_{\lceil \frac{t}{2}+2 \rceil}, S'_{\lceil \frac{t}{2}+2 \rceil}), \cdots, (M'_t, S'_t)]$. Next, the verifier repeatedly performs Steps 2 and 3 for all parts.

For example, suppose the signer sends 16 signatures $((M_1, S_1), (M_2, S_2), \cdots, (M_{16}, S_{16}))$ to the verifier. The verifier re-orders these signatures by a total order relation: $(M'_1, S'_1), (M'_2, S'_2), \cdots, (M'_{16}, S'_{16})$ such that $(M'_1, S'_1) < (M'_2, S'_2) < \cdots < (M'_{16}, S'_{16})$.

Assume there is one illegal signature: S'_{15}. The verifier performs the following procedures:

Step 1: The verifier performs one times of the batch verify with all 16 signatures:

$$(\prod_{i=1}^{16} S'_i)^e \stackrel{?}{=} \prod_{i=1}^{16} h(M'_i).$$

Since there is one illegal signature: S'_{15}, the above equation is not held. The verifier divides these 16 signatures into two parts: Part 1: $[(M'_1, S'_1), (M'_2, S'_2), \cdots, (M'_8, S'_8)]$ and Part 2: $[(M'_9, S'_9), (M'_{10}, S'_{10}), \cdots, (M'_{16}, S'_{16})]$.

Step 2: The verifier performs one time of batch verification with all signatures in Part 1:

$$(\prod_{i=1}^{8} S'_i)^e \stackrel{?}{=} \prod_{i=1}^{8} h(M'_i).$$

Since there are not illegal signatures in Part 1, the above equation is held.

Step 3: The verifier performs one time of batch verification with all signatures in Part 2:

$$(\prod_{i=9}^{16} S'_i)^e \stackrel{?}{=} \prod_{i=9}^{16} h(M'_i).$$

Since there is one illegal signature: S'_{15}, the above equation is not held. The verifier divides these 8 signatures into two parts: Part 3: $[(M'_9, S'_9), (M'_{10}, S'_{10}), \cdots, (M'_{12}, S'_{12})]$ and Part 4: $[(M'_{13}, S'_{13}), (M'_{14}, S'_{14}), \cdots, (M'_{16}, S'_{16})]$.

Step 4: The verifier performs one time of batch verification with all signatures in Part 3:

$$(\prod_{i=9}^{12} S_i')^e \stackrel{?}{=} \prod_{i=9}^{12} h(M_i').$$

Since there are not illegal signatures in Part 3, the above equation is held.

Step 5: The verifier performs one time of batch verification with all signatures in Part 4:

$$(\prod_{i=13}^{16} S_i')^e \stackrel{?}{=} \prod_{i=13}^{16} h(M_i').$$

Since there is one illegal signature: S_{15}', the above equation is not held. The verifier divides these 4 signatures into two parts: Part 5: $[(M_{13}', S_{13}'), (M_{14}', S_{14}')]$ and Part 6: $[(M_{15}', S_{15}'), (M_{16}', S_{16}')]$.

Step 6: The verifier performs one time of batch verification with all signatures in Part 5:

$$(S_{13}' S_{14}')^e \stackrel{?}{=} h(M_{13}' M_{14}').$$

Since there are not illegal signatures in Part 5, the above equation is held.

Step 7: The verifier performs one time of batch verification with all signatures in Part 6:

$$(S_{15}' S_{16}')^e \stackrel{?}{=} h(M_{15}' M_{16}').$$

Since there is one illegal signature: S_{15}', the above equation is not held. The verifier divides these 8 signatures into two parts: Part 7: $[(M_{15}', S_{15}')]$ and Part 8: $[(M_{16}', S_{16}')]$.

Step 8: The verifier performs one time of batch verification with all signatures in Part 7:

$$(S_{15}')^e \stackrel{?}{=} h(M_{15}').$$

Since there is one illegal signature: S_{15}', the above equation is not held, so the verifier knows the (M_{15}', S_{15}') is illegal.

Step 9: The verifier performs one time of batch verification with all signatures in Part 8:

$$(S_{16}')^e \stackrel{?}{=} h(M_{16}').$$

Since there are not illegal signatures in Part 8, the above equation is held.

2.4 An n-Dimension-based Batch Verification Scheme for Identifying Illegal Signatures

In 2015, an n-Dimension-based batch verification scheme for identifying illegal signatures was proposed by Ren *et al.* [16]. Their approach is an extended version of a 2D-based batch verification scheme for identifying invalid signatures [14]. For the sake of understanding their method, we introduce the 3D-based batch verification scheme for identifying illegal signatures ($n = 3$).

When the verifier receives the messages $(M_1, S_1), (M_2, S_2), \cdots, (M_t, S_t)$ from the signer, the verifier will generate an $m \times m \times m$ matrix in which m is the smallest integer which satisfies $m^3 \geq t$. The verifier performs the following procedures to verify the illegal signature.

Step 1: The verifier constructs an $m \times m \times m$ matrix.

Step 2: The verifier randomly selects and fills these t digital signatures in the $m \times m \times m$ matrix.

Step 3: The verifier performs the batch verification of each plane. The details of x-axis plane verifications are computed as follows:

$$(\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(x,i,j)})^e \stackrel{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(x,i,j)}),$$
$$x = 0, 1, \cdots, (m-1).$$

Step 4: The verifier performs the batch verification of each plane. The details of y-axis plane verifications are computed as follows:

$$(\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,y,j)})^e \stackrel{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,y,j)}),$$
$$y = 0, 1, \cdots, (m-1).$$

Step 5: The verifier performs the batch verification of each plane. The details of z-axis plane verifications are computed as follows:

$$(\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,j,z)})^e \stackrel{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,j,z)}),$$
$$z = 0, 1, \cdots, (m-1).$$

Step 6: If there are some signature-verification faults in the matrix, the verifier could find out where these signature-verification faults are located by finding the matrix positions of x, y, and z-axis plane overlap.

3 Comparisons

In this section, we compare these schemes introduced in Section 2 in terms of efficiency, the type of batch verification scheme (BVS) and misidentification (see Table 4).

3.1 Analysis of Hwang-Lee-Lai's Scheme

In Hwang-Lee-Lai's batch verification scheme for identifying illegal signatures [12], there are two assumptions:

Table 4: Comparisons among the batch verification schemes for identifying illegal signatures

Schemes	Computations with illegal signature	Computations with two or more illegal signatures	The Type of Batch Verification Scheme (BVS)	Misidentification
Hwang-Lee-Lai's Scheme [12]	$t/10E$	$t/10E + (t-1)MR$	RSA-Type BVS	No
Li-Hwang-Chen's Scheme [14]	$2\lceil\sqrt{t}\rceil$	$2\lceil\sqrt{t}\rceil$	Any Types	Yes
Atanasiu's Scheme [1]	$1+\log t$	$\frac{(1+2\log t)+(2\log t+2\log\lceil\frac{t}{2}\rceil)}{2}$	Any Types	No
Ren *et al.*'s scheme [16]	$n\lceil\sqrt[n]{t}\rceil$	$n\lceil\sqrt[n]{t}\rceil$	Any Types	Yes

t : The number of digital signatures
E : Exponential operations
MR : Modulus Remainder operations

One is that $\prod_{i=1}^{t} h(M_i) < n$, and the other is that $h(M_i)$, must be a prime where $i = 1, 2, \cdots, t$.

In a secure digital signature, the length of the signature is 1024 bits. Thus, for satisfying the assumption, $\prod_{i=1}^{t} h(M_i) < n$, the length of $h(M_i)$ is selected as 10 bits. This means the maximus of $h(M_i)$ is 1,021 which is the maximal prime of less than the $2^{10} = 1,024$. Therefore, the total computations for identifying illegal signatures in Hwang-Lee-Lai's scheme needs $t/10$ exponential operations and $t-1$ modulus remainder operations.

In Hwang-Lee-Lai's scheme, the verifier needs to verify by RSA-type batch verification scheme (BVS). Their method does not misjudge the legal signature as an illegal signature.

3.2 Analysis of Li-Hwang-Chen's Scheme

In Li-Hwang-Chen's batch verification scheme for identifying illegal signatures [14], the verifier needs to constructs an $m \times n$ matrix, where m and n are the smallest integer that satisfies $m \times n \geq t$.

In Li-Hwang-Chen's scheme, the verifier needs to verify each row and column by a general batch verification scheme [2–4, 7, 8, 10]. Any multiple digital signature schemes could be used in Li-Hwang-Chen's scheme.

For the sake of comparison, the batch verification scheme of their scheme is used in RSA signature. The computation of the RSA batch verification scheme is one exponential operation (one time verification). Therefore, the total computations for identifying illegal signatures in Li-Hwang-Chen's scheme needs $2\lceil\sqrt{t}\rceil$ exponential operations.

In Li-Hwang-Chen's scheme, the illegal signature could be precisely identified if it is only one illegal signature. However, it will have misidentification if two or more illegal signatures occur. For example, assume there are two illegal signatures in the positions $S(4,1)$ and $S(2,3)$ of matrix in Table 5. There would occur fourth verification failures in the second and the fourth rows and the first and third columns, respectively (see Table 6). However, obviously there are only two of them are real invalid signatures. There are two legal signatures, S(2,1) and S(4,3), misidentified as illegal signatures.

Table 5: A 4×4 matrix with two illegal signatures

S(1,1)	S(1,2)	S(1,3)	S(1,4)	Pass
S(2,1)	S(2,2)	$S^*(2,3)$	S(2,4)	Fail
S(3,1)	S(3,2)	S(3,3)	S(3,4)	Pass
$S^*(4,1)$	S(4,2)	S(4,3)	S(4,4)	Fail
Fail	Pass	Fail	Pass	

Table 6: Four verification failure in a 4×4 matrix with two illegal signatures

S(1,1)	S(1,2)	S(1,3)	S(1,4)	Pass
S(2,1)	S(2,2)	$S^*(2,3)$	S(2,4)	Fail
S(3,1)	S(3,2)	S(3,3)	S(3,4)	Pass
$S^*(4,1)$	S(4,2)	S(4,3)	S(4,4)	Fail
Fail	Pass	Fail	Pass	

3.3 Analysis of Atanasiu's Scheme

In Atanasiu's batch verification scheme for identifying illegal signatures [1], the verifier needs to verify each parts by a general batch verification scheme [2–4, 7, 8, 10]. Any multiple digital signature schemes could be used in his scheme.

For the sake of comparison, the batch verification scheme of their scheme is used in RSA signature. The computation of the RSA batch verification scheme is one exponential operation (one time verification). Therefore, the total computations for identifying illegal signatures in Atanasiu's scheme needs $1+2\log t$ exponential operations if it is only one illegal signature. For example, assume there is one illegal signature: S'_{15} in Figure 1. The verifier needs to perform the following batch verification: $\{P_0$, P_1, P_2, P_5, P_6, P_{13}, P_{14}, S'_{15}, and $S_{16}\}$. The total computations for identifying illegal signatures is $1 + 2\log t = 9$ exponential operations.

The best case of the total computations for identifying illegal signatures in Atanasiu's scheme needs $1 + 2\log t$

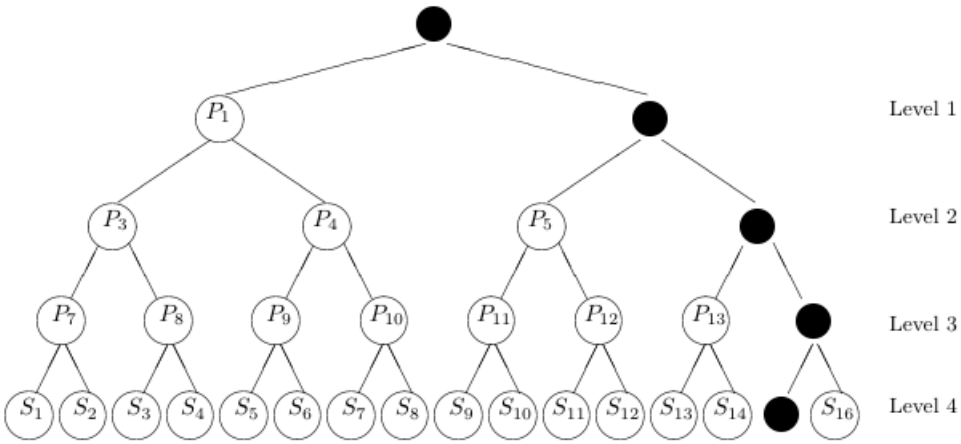

Figure 1: An example of an illegal signature: S'_{15}

exponential operations if it has two illegal signatures in the same part. For example, assume there are two illegal signatures: S'_{15} and S'_{16} in Figure 2. The verifier needs to perform the following batch verification: $\{P_0, P_1, P_2, P_5, P_6, P_{13}, P_{14}, S'_{15}$, and $S_{16}\}$. The total computations for identifying illegal signatures is $1 + 2\log t = 9$ exponential operations.

The worst case of the total computations for identifying illegal signatures in Atanasiu's scheme needs $1 + 2\log t + 2\log\lceil\frac{t}{2}\rceil$ exponential operations if it has two illegal signatures in the different parts. For example, assume there are two illegal signatures: S'_1 and S'_{15} in Figure 2. The verifier needs to perform the following batch verification: $\{(P_0, P_1, P_2, P_5, P_6, P_{13}, P_{14}, S'_{15}, S_{16})$ and $(P_3, P_4, P_7, P_8, S'_1, S_2\}$. The total computation for identifying illegal signatures is $1 + 2\log 16 + 2\log\lceil\frac{16}{2}\rceil = 15$ exponential operations.

The average computation for identifying two illegal signatures in Atanasiu's scheme needs $\frac{(1+2\log t)+(2\log t+2\log\lceil\frac{t}{2}\rceil)}{2}$ exponential operations. Obviously, the average computations for identifying two or more illegal signatures in Atanasiu's scheme needs more exponential operations than Hwang-Lee-Lai's and Li-Hwang-Chen's schemes.

The more the number of illegal signatures, the less efficient Atanasiu's scheme. For example, assume there are three illegal signatures: S'_1, S'_5, and S'_{15} in Figure 3. The verifier needs to perform the following batch verification: $\{(P_0, P_1, P_2, P_3, P_4, P_7, P_8, S'_1, S_2)$, (P_9, P_{10}, S'_5, S_6), and $P_5, P_6, P_{13}, P_{14}, S'_{15}, S_{16})\}$. The total computation for identifying illegal signatures is $9 + 4 + 6 = 19$ exponential operations.

The worst case of the total computations for identifying illegal signatures in Atanasiu's scheme needs $2t - 1$ exponential operations if it has $\frac{t}{2}$ illegal signatures in the different parts. For example, assume there are eight illegal signatures: S'_1, S'_3, S'_5, S'_7, S'_9, S'_{11}, S'_{13}, and S'_{15} in Figure 4. The verifier needs to perform the following batch verification: $\{(P_0, P_1, P_2, P_3, P_4, P_7, P_8, S'_1, S_2)$, (S'_3, S_4), (P_9, P_{10}, S'_5, S_6), (S'_7, S_8), $(P_5, P_6, P_{11}, P_{12}, S'_9, S_{10})$, (S'_{11}, S_{12}), $(P_{13}, P_{14}, S'_{13}, S_{14})$, and $(S'_{15}, S_{16})\}$. The total computation for identifying illegal signatures is $9 + 2 + 4 + 2 + 6 + 2 + 4 + 2 = 35$ exponential operations.

3.4 Analysis of Ren *et al.*'s Scheme

In Ren *et al.*'s batch verification scheme for identifying illegal signatures [16], the verifier needs to constructs an $\underbrace{m \times m \times \cdots \times m}_{n}$ n-Dimension matrix where m is the smallest integer which satisfies $m^3 \geq t$. This means the verifier could select $m = \lceil \sqrt[n]{t} \rceil$.

In Ren *et al.*'s scheme, the verifier needs to verify each plane. Any multiple digital signature schemes could be used in their scheme.

For the sake of comparison, the batch verification scheme of their scheme is used in RSA signature. The computation of the RSA batch verification scheme is one exponential operation (one time verification). Therefore, the total computation for identifying illegal signatures in Ren *et al.*'s scheme needs $n\lceil \sqrt[n]{t} \rceil$ exponential operations. For example, if there are 1000 signatures, the total computation for identifying illegal signatures in Ren *et al.*'s cheme needs $3\lceil \sqrt[3]{1000} \rceil = 30$ exponential operations.

In Ren *et al.*'s scheme, the illegal signature could be precisely identified if it is only one illegal signature. However, it will have misidentification if two or more illegal signatures occur.

4 Future Works

In this section, we propose two issues for future works.

1) An efficient batch verification scheme for identifying illegal signatures. The verifier could precisely identify these illegal signatures with low computation.

2) An application of a batch verification scheme for identifying illegal signatures. There are many appli-

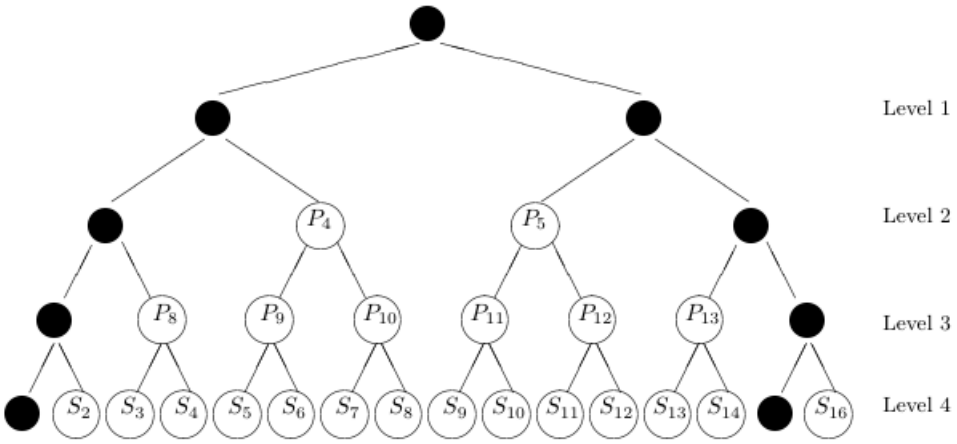

Figure 2: An example of two illegal signatures: S'_1 and S'_{15}

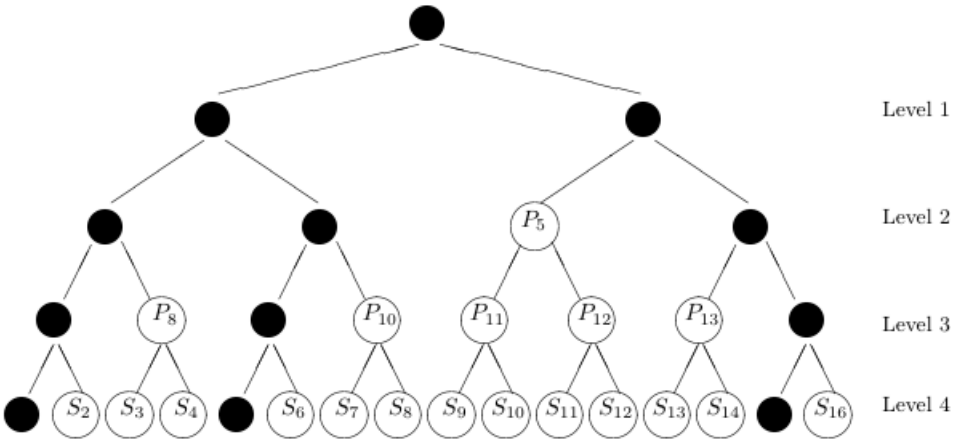

Figure 3: An example of three illegal signatures: S'_1, S'_5, and S'_{15}

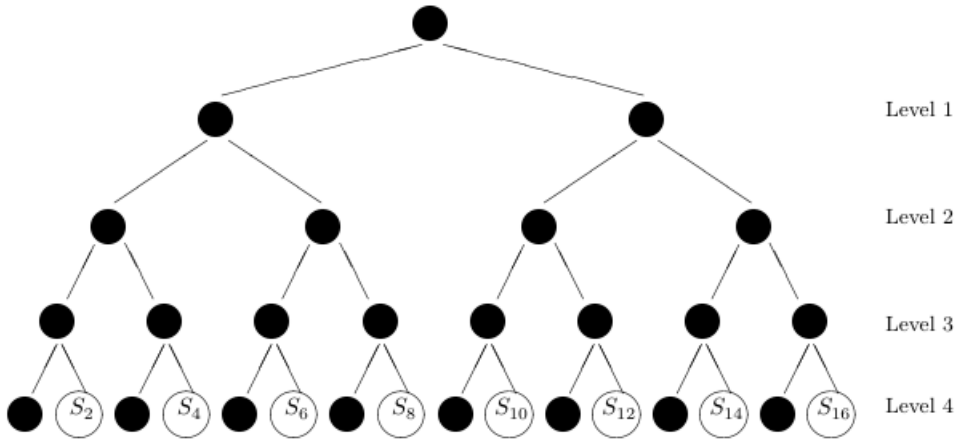

Figure 4: An example of eight illegal signatures: S'_1, S'_3, S'_5, S'_7, S'_9, S'_{11}, S'_{13}, and S'_{15}

cations in Internet of Thing (IoT). These applications with a lot of data need to verify the legal messages and signatures efficiently.

5 Conclusions

In this paper, we have reviewed some batch verification schemes for identifying illegal signatures, and proposed some criteria for evaluating these schemes. We also proposed two issues for future works.

Acknowledgments

This research was partially supported by the Ministry of Science and Technology, Taiwan (ROC), under contract no.: MOST 108-2410-H-468-023 and MOST 108-2622-8-468-001-TM1.

References

[1] A. Atanasiu, "A new batch verifying scheme for identifying illegal signatures," *Journal of Computer Science and Technology*, vol. 28, no. 1, pp. 1–8, 2013.

[2] F. Bao, C. C. Lee, M. S. Hwang, "Cryptanalysis and improvement on batch verifying multiple RSA digital signatures," *Applied Mathematics and Computation*, vol. 172, no. 2, pp. 1195–1200, 2006.

[3] T. Y. Chang, M. S. Hwang, W. P. Yang, K. C. Tsou, "A modified Ohta-Okamoto digital signature for batch verification and its multi-signature version," *International Journal of Engineering and Industries*, vol. 3, no. 3, pp. 75–83, 2012.

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

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

[6] M. S. Hwang, C. C. Chang, and K. F. Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.

[7] M. S. Hwang, T. Y. Chang, W. P. Yang, "The batch verifying multiple digital signature scheme: A modified version of Ohta-Okamoto digital signature," in *The 3rd International Conference on Next Generation Information Technology (ICNIT'12)*, pp. 732–735, 2012.

[8] 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. 1–7, 2005.

[9] M. S. Hwang, C. C. Lee, and Eric J. L. Lu, "Cryptanalysis of the batch verifying multiple DSA-type digital signatures," *Pakistan Journal of Applied Sciences*, vol. 1, no. 3, pp. 287–288, 2001.

[10] M. S. Hwang, C. C. Lee, and Y. L. Tang, "Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13–16, Xian, China, 2001.

[11] M. S. Hwang, C. C. Lee, and Y. C. Lai, "Traceability on RSA-based partially signature with low computation," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 465–468, 2003.

[12] M. S. Hwang, C. C. Lee, Y. C. Lai, "Detecting the illegal signature in multiple signatures," in *International Conference on Advanced Communications Technology (ICACT'02)*, pp. 881–882, 2002.

[13] 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.

[14] C. T. Li, M. S. Hwang, S. M. Chen, "A batch verifying and detecting the illegal signatures," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 12, pp. 5311–5320, 2010.

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

[16] Y. L. Ren, S. Wang, X. P. Zhang, and M. S. Hwang, "An efficient batch verifying scheme for detecting illegal signatures," *International Journal of Network Security*, vol. 17, no. 4, pp. 463–470, 2015.

[17] S. F. Tzeng, C. C. Lee, and M. S. Hwang, "A batch verification for multiple proxy signature," *Parallel Processing Letters*, vol. 21, no. 1, pp. 77–84, 2011.

[18] S. F. Tzeng, C. Y. Yang, and M. S. Hwang, "A new digital signature scheme based on factoring and discrete logarithms," *International Journal of Computer Mathematics*, vol. 81, no. 1, pp. 9–14, 2004.

Biography

Hsien-Tsen Pan received B.S. in Business Administration From Soochow University Taipei Taiwan in 1999; M.S in Information Engineering, Asia University Taichung Taiwan 2015; Doctoral Program of Information Engineering, Asia University Taichung Taiwan from 2015 till now. From 2011 to 2014, he was the manager in Enterprise Service Chunghwa Telecom South Branch Taichung Taiwan. From 2014 to 2017, he was the operation manager in Medium division Taiwan Ricoh Co., Ltd. Taichung Taiwan From 2017 Sep 20 he is the Apple MDM Server Service VP in Get Technology Co.Ltd. Taipei Taiwan.

Eko Fajar Cahyadi is a lecturer in the Department of Telecommunication Engineering, Institut Teknologi Telkom Purwokerto, Indonesia. He currently pursuing a Ph.D. degree in the Department of Computer Science

and Information Engineering at Asia University, Taiwan. He receives the B. Eng. and M. Sc. degree in electrical engineering from Institut Sains dan Teknologi Akprind Yogyakarta in 2009, and Institut Teknologi Bandung in 2013, respectively. His research interest includes wireless network security, optical fiber communication, and teletraffic engineering.

Shu-Fen Chiou received a B.B.A degree in Information Management from National Taichung Institute of Technology, Taichung, Taiwan, ROC, in 2004; She studied M.S. degree in Computer Science and Engineering from National Chung Hsing University for one year, and she started to pursue the Ph.D. degree. She received a Ph. D. from Computer Science and Engineering from National Chung Hsing University in 2012. She is currently an assistant professor of department of Information Management, National Taichung University of Science and Technology. Her current research interests include information security, network security, data hiding, text mining and big data analysis.

Min-Shiang Hwang received M.S. in industrial engineering from National Tsing Hua University, Taiwan in 1988; and Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan in 1995. He was a professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2009. He was also a visiting professor with University of California (UC), Riverside and UC. Davis (USA) during 2009-2010. He was a distinguished professor of Department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Exceilent Research Award of National Science Council (Taiwan). Dr. Hwang was a dean of College of Computer Science, Asia University (AU), Taichung, Taiwan. He is currently a chair professor with Department of Computer Science and Information Engineering, AU. His current research interests include information security, electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.