Batch Identification Game Model for Invalid Signatures in Wireless Mobile Networks

Jing Chen, Kun He, Quan Yuan, Guoliang Xue, *Fellow, IEEE*, Ruiying Du, and Lina Wang

Abstract—Secure access is one of the fundamental problems in wireless mobile networks. Digital signature is a widely used technique to protect messages' authenticity and nodes' identities. From the practical perspective, to ensure the quality of services in wireless mobile networks, ideally the process of signature verification should introduce minimum delay. Batch cryptography technique is a powerful tool to reduce verification time. However, most of the existing works focus on designing batch verification algorithms for wireless mobile networks without sufficiently considering the impact of invalid signatures, which can lead to verification failures and performance degradation. In this paper, we propose a Batch Identification Game Model (BIGM) in wireless mobile networks, enabling nodes to find invalid signatures with reasonable delay no matter whether the game scenario is complete information or incomplete information. Specifically, we analyze and prove the existence of Nash Equilibriums (NEs) in both scenarios, to select the dominant algorithm for identifying invalid signatures. To optimize the identification algorithm selection, we propose a self-adaptive auto-match protocol which estimates the strategies and states of attackers based on historical information. Comprehensive simulation results in terms of NE reasonability, algorithm selection accuracy, and identification delay are provided to demonstrate that BIGM can identify invalid signatures more efficiently than existing algorithms.

Index Terms—Batch identification, game theory, wireless mobile networks

1 Introduction

In the past few years, Wireless Mobile Networks (WMNs) have been dramatically developed due to the proliferation of inexpensive, widely available wireless mobile devices [1], [2], [3]. With the increasing number of mobile applications, such as social media networks, GPS, yelp, etc., people's life has been inseparable from mobile devices which can access the Internet at anytime and anywhere [4]. However, due to the openness characteristic of wireless channels, it becomes easier for malicious nodes to interfere the access process by tampering or forging request messages [5], [6]. To protect the security of access, one effective approach is to sign each outgoing message with a digital signature, and let the destination verify each received signature [7], [8]. Generally, signature verification induces extra delay and computational cost. The traditional way that verifying messages signatures individually could induce tremendous delay and severely affect the Quality of Service (QoS), especially when network traffic is heavy and a large number of signatures need to be verified [9].

To reduce verification delay and ensure QoS, researchers proposed the batch cryptographic technique which is a promising new direction in computer and communication security [10]. The concept of batch cryptography was introduced by Fiat [11] in 1990 for an RSA-type signature, and the first efficient batch verifier was proposed by Naccache et al. [12] in 1994 for DSA-type signatures. Currently, researchers focus on two directions to apply the batch cryptography concept in WMNs: *batch verification* and *batch identification*.

Batch verification deals with n (message, signature) pairs as a batch at a time [13]. As a result, compared with the traditional way, the validity of a batch can be checked more efficiently, and the verification delay can be remarkably reduced. In detail, batch verification methods return true if all of the n signatures are valid, and false when there is any invalid one. In 2008, considering that the verification of massive messages may induce huge time cost in mobile networks, Yu et al. [14] proposed an efficient identity-based batch verification scheme to reduce the delay in network coding. Zhang et al. [15] discussed a batch signature verification scheme for the communications between mobile nodes and the infrastructure to lower the total verification time. Horng et al. [16] presented a group signature and batch verification method for secure pseudonymous authentication in VANET. Unfortunately, even though those schemes could protect the authenticity of messages, their performance can be severely affected if there are invalid signatures existing in the verified batch. Adversaries can negate the advantages of batch verification by polluting signatures within a batch. It is unrealistic to completely prevent all adversaries from generating false messages with invalid signatures. Thus, to guarantee the performance of batch verification, we should identify invalid signatures in a batch rapidly.

Batch identification is a technique to find the bad signatures within a batch when the batch verification fails. Due to the inefficiency of individual identification, divide-and-conquer

- *J. Chen, K. He, R. Du, and L. Wang are with the State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan, Hubei 430072, China.*
  E-mail: {chenjing, kunhe, duraying, lnwang}@whu.edu.cn.
- *Q. Yuan is with the Department of Math and Computer Science, University of Texas-Permian Basin, Odessa, TX 79762.*
  E-mail: yuan_q@utpb.edu.
- *G. Xue is with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281.*
  E-mail: xue@asu.edu.

Manuscript received 6 Jan. 2016; revised 14 Aug. 2016; accepted 26 Aug. 2016. Date of publication 31 Aug. 2016; date of current version 3 May 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2016.2604820

techniques have been proposed to improve the performance of batch identification [17], [18]. Those methods can significantly reduce the identification time at different levels. Existing batch identification algorithms have been developed into two main branches: *special* and *generic*. The special methods are designed for certain batch signature types such as RSA-type, DSA-type, and pairing-type. Lee et al. [19] proposed a method to identify bad signatures in RSA-type batches. Later, Law and Matt [20] presented a quick binary and exponentiation method to find invalid signatures. Stanek [21] showed that the method in [19] was flawed, and proposed an improved protocol to resist attacks. Matt [22] discussed a solution in pairing-based signature scheme, which can identify nontrivial numbers of invalid signatures in batches. Though these works are state-of-the-art, it is challenging to apply them with various batch verification algorithms.

On the other side, the generic batch identification methods utilize the group testing technique [23] to find invalid signatures with the minimal number of tests, which can be applied with any signature types. Pastuszak et al. [17] designed a divide-and-conquer verifier, which splits an batch instance into sub ones, and applied the generic test to each sub-batch recursively, until all bad signatures are identified. Zaverucha et al. [23] presented and compared some group testing algorithms for finding invalid signatures. Zhang et al. [24] adopted the group testing technique to find invalid signatures in a batch in mobile networks. Lee et al. [25] proposed a secure batch verification with group testing to improve the real-time performance of mobile networks. Note that those generic methods are usually suitable for a specific attack situation in terms of the number of invalid signatures. Their performance may severely degrade if the number of invalid signatures varies, when the attack strategy is changed by the movement of adversaries or the alteration of false message attacking frequency. To select the most suitable batch verification algorithm, Akinyele et al. [26] first proposed an automated tool. However, they did not sufficiently consider the automatic selection of batch identification. Therefore, designing a generic and auto-match batch identification solution towards the heterogeneous and dynamic attack scenario becomes significant.

In this paper, we propose a Batch Identification Game Model (BIGM) in WMNs, enabling nodes to identify invalid signatures with a reasonable delay under heterogeneous and dynamic attacks. To enhance the flexibility of our model, we subdivide BIGM into Batch Identification Game Model with Complete information (C-BIGM) and Batch Identification Game Model with Incomplete information (I-BIGM) to protect regular nodes from the attacks of invalid signatures in different scenarios. In our model, a mobile node may be a regular one or a malicious one, and the game occurs between a regular node and its malicious neighbors. The regular node, as a verifier, aims at finding the invalid signatures to eliminate the impact of malicious nodes. The malicious neighbors, as attackers, intend to interpose batch verification process by broadcasting false messages signed by invalid signatures with different frequencies. Our main contributions are summarized as follows:

- To evaluate the effectiveness of batch identification, we recast three generic batch identification algorithms based on the group testing techniques used in [24]. We analyze and compare the time complexities of these algorithms with experiments. We observe that none of them has universal advantages in all situations. Therefore, we need an auto-match scheme to choose the batch identification algorithm adaptively, when the attack strategy changes.
- We design a game model to find the dominant batch identification algorithm against various attack strategies. We analyze and prove the existence of Nash equilibriums in the complete information scenario and the incomplete information scenario, respectively. Note that the members in strategy set are alternative, as long as these generic batch identification algorithms have their own advantages under different attack strategies. Thus, our game model provides a paradigm to select the dominant one to identify invalid signatures. Furthermore, BIGM is a generic solution, which can be equipped with any batch signature scheme.
- Considering that the dominant strategy may not always be the optimal choice, we propose a self-adaptive auto-match protocol to improve the selection accuracy of the batch identification algorithm based on history information. From the analysis and simulations, we find that it can effectively strengthen the prediction accuracy of nodes' states and the sensitivity of attack perception, and reduce the average identification delay of the whole network.

The reminder of the paper is organized as follows. In Section 2, we state the problem. In Section 3, three generic batch identification algorithms are recast and analyzed. In Section 4, we discuss the batch identification game models with complete information and incomplete information. In Section 5, a self-adaptive auto-match batch identification protocol is presented as an important part of our game model. In Section 6, we evaluate the performance of our scheme. Finally, section 7 concludes the paper.

Compared with our conference version [27], we make significant improvements in four aspects as follows. First, we complement each batch identification algorithm with the analysis of time complexity. Second, we extend our game model to support both the complete information and the incomplete information scenarios at the same time. Third, we propose a self-adaptive auto-match protocol to improve the prediction accuracy of nodes' states. Finally, we redesign the simulations to analyze the reasonability of NE, the selection accuracy of algorithms, and the identification delay, respectively.

2 Problem Statement

2.1 Network Model

We consider that the network has two layers as shown in Fig. 1. The bottom layer consists of mobile nodes accessing the network via GSM, 3G, etc. Each node has its own public/private keys, which are used to sign the outgoing messages and to verify the signatures of the received messages. The top layer is composed of an authority center and base stations. The authority center manages the key operations of all regular nodes which can be authenticated and authorized by offline or other methods, including generation, distribution, storage, update, and destruction. If mobile nodes directly

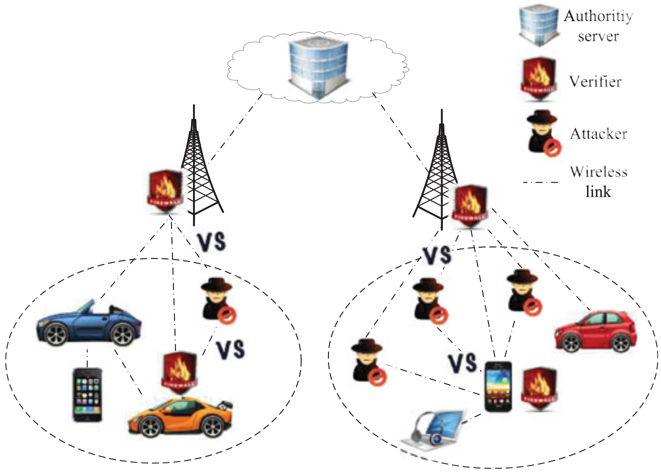


Fig. 1. The network model.

communicate with each other by wifi, bluetooth, etc., they should mutually verify the validity of the other party. If base stations forward messages, they have to verify the validity of requests. Hence, both base stations and mobile nodes can be attack targets. They should protect their own security, and identify invalid signatures in false messages by themselves.

2.2 Attack Model

We assume that the network consists of regular nodes (called *verifiers*), and malicious nodes (called *attackers*), which are the two players in the game. For a verifier, its attackers intend to interpose its batch verification process by broadcasting false messages with invalid signatures, while the verifier needs to identify the invalid signatures quickly to resist the attack. Note that the verifier is one player and all its malicious neighbors act as another player. In this paper, the verifier can be a base station or a mobile node. In addition, in our attack model, the capability of attackers is described as follows:

- Attackers cannot interfere or control the distribution process of key pairs, while those private keys can be distributed by a secure channel or offline methods. They launch attacks in the way of broadcasting false messages with invalid signatures, to disturb the batch verification process, and to consume the verifier's resources.
- Each attacker can choose and change the number of false messages at will. The verifier selects its strategy according to the sum of invalid signatures from its malicious neighbors.
- Attackers are divided into different types based on their preferences. For example, some attackers consider the risk of being traced, but others do not have such concerns.
- Attackers can acquire the public information of the verifier, such as the public key and the cryptographic algorithm.

2.3 Design Goals and Notations

The main idea of our game model is to help regular nodes to select the suitable batch identification algorithm no matter what the attack strategy is. We refine it as follows:

TABLE 1
Description of Major Notations

Notation	Description
n	The number of signatures which are needed to be verified in each round
d	The upper bound number of invalid signatures in a batch
$\log x$	$\log_2 x$
$\lceil x \rceil$ ($\lfloor x \rfloor$)	The smallest (largest) integer not less (greater) than x
$M_A(d,n)$	The worst-case number of tests required by algorithm A
C_{BV}^n	The cost of batch verification for n signatures. It equals to the number of tests required
Q	The communication benefit in ideal network environment
$\alpha_i(j)$	The cost of verifier i when it chooses the batch identification algorithm j
$\sigma_A(l)$	The attack cost which equals to the number of invalid messages. Attack strategy $l \in \{H, L\}$ while H indicates high attack frequency and L indicates low attack frequency.
$\alpha(j,k)$	The cost of batch identification algorithm. It is determined by verification strategy $j \in \{CBI, MRI, II\}$ and attack strategy $k \in \{H, L\}$, and it equals to the number of tests required.
S_a^h	The attackers' strategy in round h.

- BIGM has strong flexibility to handle various scenarios, such as complete information or incomplete information, static or dynamic, and homogeneous or heterogeneous.
- BIGM is a distributed scheme which means that it can work well even if the authority center is off-line. Each regular node assesses current attack strategy it faces and determines the defence strategy according to the history information collected by itself.
- BIGM has the self-evolution ability to continually optimize the selection accuracy of batch identification algorithm from two aspects. One is that it can select more reasonable strategy, not just depending on Nash Equilibrium. The other is that it can dynamically adjust the estimation results and improve the accuracy as time goes.

The main notations are summarized in Table 1.

3 Generic Batch Identification Algorithms

Generic batch identification algorithms for a bad batch usually adopt the group testing technique. In this section, we describe and analyze the idea of three generic algorithms based on the representative group testing techniques, including individual identification, generalized binary splitting, and Li's scheme [24], to identify d invalid signatures in a batch of n messages.

3.1 Individual Identification

One simple solution to identify all invalid signatures in a bad batch, is to verify each signature individually. Note that signatures are not aggregated with others until all invalid signatures have been found. Many schemes, which mainly focus on the batch verification process, adopt this algorithm.

That is, once the batch verification fails, *Individual Identification* (II) is employed to find all the invalid signatures. Obviously, the time complexity of II is $O(n)$, where n is the number of signatures to verify, as shown in Table 1.

3.2 Condensed Binary Identification

Inspired by the basic binary identification algorithm in [23], we present an improved scheme called the *Condensed Binary Identification* (CBI) algorithm. In the basic binary identification, it first divides the n messages into two groups of equal size. Then, those two groups are verified using batch verification individually. If the batch verification succeeds, there is no invalid signature in that group. Otherwise, messages in that group will be further divided into two sub-groups, and each sub-group is verified individually. That process repeats until all of the messages pass the batch verification. CBI improves the basic binary identification by adjusting the group size for efficiency. Concerning the probability, the ideal situation is that, each sub-group of $\lceil n/d \rceil$ messages has one invalid signature, where $\lceil n/d \rceil$ denotes the smallest integer not less than n/d. If we can adjust the sub-group size based on the number of the remaining invalid signatures, it can reduce the number of reverifications in attacks. CBI is described as Algorithm 1, where z, θ and v are three intermediate variables.

Algorithm 1. Condensed Binary Identification Algorithm

```
while true do
    if $n \le 2d-2$ then
        Verify $n$ messages using II;
        return;
    else
        $z = n - d + 1$;
        $\theta = \lfloor \log(z/d) \rfloor$;
    end
    Verify the prevenient $2^\theta$ messages with batch verification;
    if verification succeeds then
        $n = n - 2^\theta$;
        continue;
    else
        identify an invalid signature by basic binary
        identification after verifying $v$ messages;
        $n = n - 1 - v$;
        $d = d - 1$;
        continue;
    end
end
```

Theorem 3.1. *Assuming $\theta = \lfloor \log(z/d) \rfloor$, and $z = 2^\theta d + 2^\theta k_1 + k_2$, where $k_1 \ge 0$, and $0 \le k_2 < 2^\theta$, we have the worst-case number of required verifications of CBI, $M_{CBI}(d,n)$ as,*

$$M_{CBI}(d,n) = \begin{cases} n & \text{for } n \le 2d-2 \\ \theta d + k_1 & \text{for } n \ge 2d-1 \end{cases} \quad (1)$$

the time complexity of CBI is $O(d \log(n/d))$.

Proof. According to CBI, if $n \le 2d-2$, it verifies n messages using II. Hence, $M_{CBI}(d,n) = n$. If $n \ge 2d-1$, there are two cases to consider. One is that the first 2^θ samples are valid, while the other is that the first 2^θ sample messages include invalid signatures. Therefore, $M_{CBI}(d,n) = \{1 + M_{CBI}(d, n-2^\theta), \theta + M_{CBI}(d-1, n-1)\}$.

Case 1: $M_{CBI}(d,n) = 1 + M_{CBI}(d, n-2^\theta)$. Due to $n' = n - 2^\theta$ and $d' = d$, we have $z' = n' - d' + 1 = n - 2^\theta - d + 1 = z - 2^\theta$. Then,

$$z' = \begin{cases} 2^\theta d + 2^\theta(k_1 - 1) + k_2 & \text{for } k_1 \ge 1 \\ 2^\theta d + (k_2 - 2^\theta) & \text{for } k_1 = 0. \end{cases}$$

By mathematical induction,

$$M_{CBI}(d, n - 2^\theta) = \begin{cases} (\theta+2)d + k_1 - 2 & \text{for } k_1 \ge 1 \\ (\theta+1)d + d - 2 & \text{for } k_1 = 0. \end{cases}$$

Case 2: $M_{CBI}(d,n) = \theta + M_{CBI}(d-1, n-1)$. Due to $n' = n - 1$ and $d' = d - 1$, we have $z' = n' - d' + 1 = z$

$$z' = \begin{cases} 2^\theta d + 2^\theta k_1 + k_2 & \text{for } k_1 \le d-2 \\ 2^{(\theta+1)} d - 2^{(\theta+1)} + k_2 & \text{for } k_1 > d-2. \end{cases}$$

Hence, by mathematical induction

$$M_{CBI}(d-1, n-1) = \begin{cases} (\theta+2)(d-1) + k_1 + 1 & \text{for } k_1 \le d-2 \\ (\theta+3)(d-1) & \text{for } k_1 > d-2. \end{cases}$$

Consequently, for both cases under $n \ge 2d-1$, $k_1 = 0$ and $k_1 > d-2$ are mutually exclusive, we have

$$M_{CBI}(d,n) = (\theta+2)d + k_1 - 1, \textit{for } n \ge 2d-1, 1 \le k_1 \le d-2.$$

Because $\theta = \lfloor \log(z/d) \rfloor$, the time complexity of CBI is $O(d \log(n/d))$. □

3.3 Multiple Rounds Identification

In *Multiple Rounds Identification* (MRI) algorithm, we identify the invalid signatures in an iterative way which has m ($2 \le m \le n$) rounds, as described in Algorithm 2. In the first round, the n pending messages are divided into δ_1 groups, and each group has γ_1 messages except the last group. Then, each group is verified respectively. The groups identified with invalid signatures are aggregated as a new pending message batch. In the second round, that new message batch is divided into δ_2 groups of γ_2 messages. In general, in round i, $2 < i < m$, messages from the contaminated groups of round $i-1$ are pooled, and arbitrarily divided into δ_i groups of γ_i size except the last group whose size may be smaller than γ_i. A batch verification test is performed on each group. Note that γ_m is set to be 1. Thus every invalid signature is identified at round m.

Theorem 3.2. *Let $M^m_{MRI}(d,n)$ denote the number of verifications required by MRI algorithm in the worst case, we have $M^m_{MRI}(d,n) \le ed\ln(n/d)$, where e is the base of natural logarithm. The time complexity of MRI is $O(\log(n/d))$.*

Proof. To simplify the proof, we assume that n is divisible by γ_1. There are three cases in our proof.

Case 1: $m = 1$. This case corresponds to the II algorithm. Thus $M^1_{MRI}(d,n) = n$.

Case 2: $m = 2$. In this case, we have

$$M^2_{MRI}(d,n) = \delta_1 + \delta_2 \le \frac{n}{\gamma_1} + d\gamma_1.$$

TABLE 2
Batch Identification Algorithms Comparison

Algorithm	Complexity
Individual Identification (II)	$O(n)$
Condensed Binary Identification (CBI)	$O(d\log(n/d))$
Multiple Rounds Identification (MRI)	$O(\log(n/d))$

Since $\gamma_1 = \left\lceil (\frac{n}{d})^{\frac{m-1}{m}} \right\rceil$, we have $\delta_1 \le \sqrt{nd}$ and $M^2_{MRI}(d,n) \le 2\sqrt{nd}$.

Case 3: m is general in this case. Obviously,

$$M^m_{MRI}(d,n) = \sum_{i=1}^{m} \delta_i \le \frac{n}{\gamma_1} + \frac{d\gamma_1}{\gamma_2} + \cdots + \frac{d\gamma_{m-2}}{\gamma_{m-1}} + d\gamma_{m-1}.$$

Due to $\gamma_i = \lceil (\frac{n}{d})^{\frac{m-i}{m}} \rceil$, $\gamma_i \ge (\frac{n}{d})^{\frac{m-i}{m}}$, where $1 \le i \le m-1$. That gives $\delta_i \le d(\frac{n}{d})^{(\frac{1}{m})}$, and $M^m_{MRI}(d,n) \le md(\frac{n}{d})^{(\frac{1}{m})}$.

The first derivative of the above upper bound with respect to a continuous m is

$$d(\frac{n}{d})^{\frac{1}{m}}\left(\frac{m - \ln\frac{n}{d}}{m}\right),$$

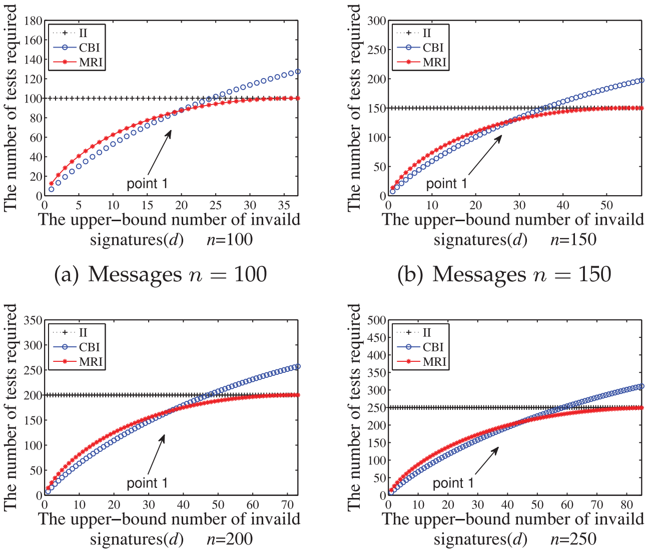
which has a unique root $m = \ln(\frac{n}{d})$. Obviously, $m = \ln(\frac{n}{d})$ is the unique maximum of the upper bound. Hence, we have

$$M^m_{MRI}(d,n) \le md\left(\frac{n}{d}\right)^{(\frac{1}{m})} \le ed\ln\left(\frac{n}{d}\right).$$

To execute the algorithm, one needs to compute the optimal m and γ_i for $i = 1, \ldots, m$. Each γ_i can be computed within constant time. With approximating the optimal m by the ceiling or floor function of $\log(\frac{n}{d})$, MRI algorithm's complexity is $O(\log(n/d))$. □

3.4 Performance Comparison

Based on the above analysis, we summarize the time complexity of these algorithms in Table 2. In addition, in Fig. 2, we investigate the relationship between the number of invalid signatures and the number of required batch verification tests, when the number of messages n varies. Figs. 2a, 2b, 2c, and 2d present the situation where n is equal to 100, 150, 200, and 250, respectively. The result shows that given a specific n, the number of required batch verifications ascends as the number of invalid signatures upper-bound increases in CBI and MRI, but not in II. Also, CBI has a lower start point and a larger slope, while MRI has a higher start point, and its slope turns smaller. As a result, in Fig. 2, CBI and MRI eventually meet at a point, marked as point 1. For our game model design, we define that point 1 as *the demarcation point of attack strategy*, because the optimal batch identification algorithm is changed at that point. That is, if the number of invalid signatures is less than that of the demarcation point, given the message number n, it means that attackers adopt the *low-frequency attack*, denoted by strategy L. Otherwise, they employ the *high-frequency attack*, represented by strategy H. Each batch identification algorithm has its own advantage under a specific attack strategy. If we can automatically choose the batch identification algorithms based on the attack strategy, it can achieve better performance.



(a) Messages $n = 100$

(b) Messages $n = 150$

(c) Messages $n = 200$

(d) Messages $n = 250$

Fig. 2. The number of required batch verifications.

Algorithm 2. Multiple Rounds Identification Algorithm

```
Copy $n$ sample messages to test_batch;
while $i \le m$ do
    $\gamma_i = \left\lceil (n/d)^{\frac{m-i}{m}} \right\rceil$;
    $\delta_i = \lfloor n/\gamma_i \rfloor + 1$;
    Divide test_batch into $\delta_i$ groups of $\gamma_i$ messages (may be less than $\gamma_i$ in the last group);
    for $j = 0$ to $j < \delta_i$ do
        if Batch verification on group $j$ succeeds then
            Remove the contents of group $j$ from test_batch;
        end
        $j++$;
    end
    $i = i + 1$;
end
return test_batch;
```

4 Batch Identification Game Model

The algorithms in Section 3 assist verifiers to find invalid signatures when the batch verification fails. From Fig. 2, we see that it is challenging to find an optimal algorithm for a general purpose, since those algorithms have different capabilities under different attack strategies. Therefore, the first step of our scheme is to find the dominant strategy by the game model from the three algorithms in Section 3.

To satisfy the flexibility requirement of our design goal, we consider two general scenarios in wireless mobile networks, respectively. In the first scenario, the verifier knows the strategy set and payoff function of attackers. Towards this scenario, we propose the *Batch Identification Game Model with Complete information* (C-BIGM). For the second one, we consider a more pervasive situation where the verifier cannot acquire exact information of attackers, and attackers may adopt different strategies at a certain probability. Towards this scenario, we propose the *Batch Identification Game Model with Incomplete information* (I-BIGM). Obviously, C-BIGM and I-BIGM are the specific instances of BIGM.

4.1 Game Model Definition

We consider the problem between a verifier and its attackers as a dynamic game, where attackers select the attack strategy first, and the verifier picks the batch identification algorithm accordingly. The definition of BIGM is represented by a triple (PL, S, U), where PL is the player set, S denotes the strategy set of players, and U stands for the payoff function set. The detailed description is as follows.

4.1.1 Players

The player set is represented by $PL = \{PL_i\}_{i=1}^{l}$, where i is the index number of a player, and l is the total number of players. Obviously, the set PL includes two players $(l = 2)$. One is the verifier, and the other is the attackers, which are the verifier's malicious neighbors.

4.1.2 Strategy Set

The strategy set of players is $S = \{S_a, S_v\}$. Different players in the game may have different strategies. For attackers, the adopted strategies fall into two types, high-frequency attack H and low-frequency attack L, in terms of the total number of invalid signatures. Hence, the strategy set of attackers is denoted as $S_a = \{H, L\}$. Note that the attack strategy is determined by the sum of invalid signatures of the verifier's malicious neighbors, while each malicious neighbor can randomly select its false message number. On the other side, the verifier's strategy set is $S_v = \{CBI, MRI, II\}$, which includes the three batch identification algorithms defined in Section 3.

4.1.3 Payoff Function

Each regular node acts as a verifier to protect its QoS. Let Q denote the communication benefit in an ideal mobile network environment. For the verifier V, the payoff function is $u_V = b_V - c_V$, where b_V is the communication benefit Q, and c_V indicates the total cost of batch verification and batch identification. The cost of batch verification for n messages, denoted as C_{BV}^{n}, is determined by the batch verification algorithm. The cost of batch identification algorithm is represented by $\alpha(j,k)$, which is determined by the identification strategy $j \in \{CBI, MRI, II\}$, and the attack strategy $k \in \{H, L\}$. To simplify notations, we use 1, 2, 3 to index the algorithm CBI, MRI, and II. Note that $\alpha(j,k)$ is determined by the number of required batch verification tests. With the above discussion, the payoff function of the verifier V can be defined as $u_V = Q - C_{BV}^{n} - \alpha(j,k)$.

Recall that the intention of attackers is to consume the verifier's resources by broadcasting false messages, and eventually to downgrade the QoS of the wireless mobile network. The payoff function of attackers A is $u_A = b_A - c_A$, where b_A is the loss of QoS, which is affected by the verification cost of the verifier. Therefore $b_A = C_{BV}^{n} + \alpha(j,k)$. c_A indicates the attack cost, which is determined by the number of the broadcasted false messages with invalid signatures, denoted by $\sigma(k)$ $(k \in \{H, L\})$. Therefore, the payoff function is $u_A = C_{BV}^{n} + \alpha(j,k) - \sigma(k)$.

4.2 Batch Identification Game Model with Complete Information

In this game scenario, attackers launch Denial of Service (DoS) attack by sending different quantities of invalid signatures, and the verifier selects batch identification algorithm after the failure of batch verification. Because the verifier can observe its neighboring attackers' behaviors, and make decision afterwards, obviously it is a two-stage sequential game.

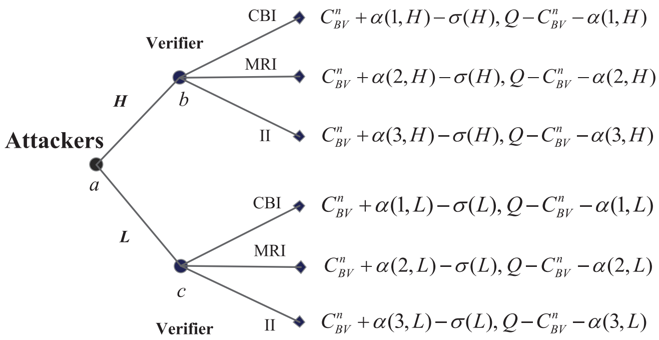

CBI
$C_{BV}^{n}+\alpha(1,H)-\sigma(H), Q-C_{BV}^{n}-\alpha(1,H)$
Verifier
MRI
$C_{BV}^{n}+\alpha(2,H)-\sigma(H), Q-C_{BV}^{n}-\alpha(2,H)$
H
b
Attackers
II
$C_{BV}^{n}+\alpha(3,H)-\sigma(H), Q-C_{BV}^{n}-\alpha(3,H)$
a
CBI
$C_{BV}^{n}+\alpha(1,L)-\sigma(L), Q-C_{BV}^{n}-\alpha(1,L)$
L
MRI
$C_{BV}^{n}+\alpha(2,L)-\sigma(L), Q-C_{BV}^{n}-\alpha(2,L)$
c
Verifier
II
$C_{BV}^{n}+\alpha(3,L)-\sigma(L), Q-C_{BV}^{n}-\alpha(3,L)$

Fig. 3. Two-stage complete information game tree of C-BIGM.

According to the analysis in Section 3, we have $\sigma(L) < \sigma(H)$. The performance difference of batch identification algorithms depends on the total quantity of messages, and the upper bound of invalid signatures number. From Fig. 2, we have $\alpha(1,L) < \alpha(2,L) < \alpha(3,L)$ when attackers adopt strategy L, and $\alpha(2,H) < \alpha(3,H) < \alpha(1,H)$ when attackers employ strategy H. Taking all those factors into consideration, from Fig. 2, we can achieve that

$$\begin{aligned}\alpha(1,L) &< \alpha(2,L) < \alpha(2,H)\\ &< \alpha(3,L) = \alpha(3,H) < \alpha(1,H).\end{aligned} \tag{2}$$

In addition, the verifier can acquire the total number of messages by sniffing, but it can hardly estimate the accurate upper bound of the invalid signatures number. The goal of C-BIGM is to find the most suitable batch identification algorithm for the verifier in this scenario. We analyze C-BIGM with a two-stage game tree in Fig. 3.

Theorem 4.1. *The C-BIGM has a pure strategy Nash equilibrium.*

Proof. Since C-BIGM is a two-stage sequential game, we analyze it in each stage. Fig. 3 shows that attackers select attack strategy in stage 1, and the verifier chooses the batch identification algorithm in stage 2 when batch verification fails.

First of all, let us consider the choice of the verifier in stage 2 after attackers have employed a specific attack strategy. If the attack strategy is L, which means that attackers send a relatively small number of false messages, the game moves to branch c of game tree in Fig. 3. As the previous analysis, for the verifier, we can get $\alpha(1,L) < \alpha(2,L) < \alpha(3,L)$. Hence, the CBI algorithm can bring the largest benefit to the verifier, and it is the best choice for the verifier under attack strategy L. If the attack strategy is H, which means that a significant number of false messages are sent by attackers, the game moves to branch b of game tree. Due to $\alpha(2,H) < \alpha(3,H) < \alpha(1,H)$, accordingly, the MRI algorithm is the dominant option for the verifier in this case. In a word, if attackers select strategy L, the verifier will choose CBI. Otherwise, the verifier will adopt MRI for batch idenfitication. As a result, we can simplify the game tree as Fig. 4.

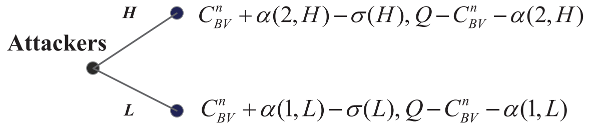

Attackers
H
$C^n_{BV}+\alpha(2,H)-\sigma(H), Q-C^n_{BV}-\alpha(2,H)$
L
$C^n_{BV}+\alpha(1,L)-\sigma(L), Q-C^n_{BV}-\alpha(1,L)$

Fig. 4. The simplified game tree of C-BIGM.

Second, we analyze the choice of attackers in stage 1 from Fig. 4. From the simplified game tree, attackers only need to compare the benefit of strategy H with that of strategy L. From Section 4.1.3, we can find that $\alpha(j,k)$ is determined by the number of required batch verification tests, and $\sigma(k)$ is determined by the number of the broadcasted false messages with invalid signatures. The delay of one batch verification for n signatures is in the range of $0.87n + 6.42$ ms to $6.6n + 6.42$ ms according to different algorithms, while the delay for forging one false message is only about 0.2 ms [16]. Moreover, from Fig. 2, we observe that the number of required batch verification tests increases greater than the upper-bound number of invalid signatures. For instance, in Fig. 2a which has the smallest n and the required test number in Figs. 2a, 2b, 2c, and 2d, if the attack strategy is changed from L to H (e.g., 10 to 30), the number of tests required varies from 63 to 98. In other figures, the gap will become larger. Therefore, we can get

$$\alpha(2,H) - \alpha(2,L) > \sigma(H) - \sigma(L). \qquad (3)$$

From Equation (2), we can get

$$\alpha(2,H) > \alpha(2,L) > \alpha(1,L). \qquad (4)$$

Hence,

$$\alpha(2,H) - \alpha(1,L) > \alpha(2,H) - \alpha(2,L) > \sigma(H) - \sigma(L). \qquad (5)$$

Because $C^n_{BV} + \alpha(2,H) - \sigma(H) > C^n_{BV} + \alpha(1,L) - \sigma(L)$, the benefit of strategy H is higher than that of strategy L for attackers. As a result, attackers will adopt strategy H. Considering the analysis of stage 2 in game tree, we know that the verifier will choose MRI algorithm. Therefore, C-BIGM has a pure Nash equilibrium (H, MRI). □

Notice that we only consider the number of verifications required, and the invalid signatures as the cost, if more factors are considered, such as the consumption difference between sending and receiving packets, the choice of attackers may be changed, and the Nash equilibrium is not (H, MRI) any more. However, from the previous analysis, we can conclude that the pure Nash equilibrium still exists no matter which strategy attackers select.

4.3 Batch Identification Game Model with Incomplete Information

In this section, we analyze the other instance I-BIGM for a more ubiquitous situation. Generally, attackers hiding in the darkness can monitor and collect the verifier's information as common knowledge. However, it is hard for the verifier to acquire the complete information of attackers in advance. Recall that there are different types of attackers with various preferences in wireless mobile networks, and each attacker sends different number of false messages. The verifier does not exactly know which strategies are adopted by which type of attackers. I-BIGM is designed toward that scenario.

TABLE 3
Game Array between the Verifier and Its Hot-Headed Attackers

A \ V	CBI(1)	MRI(2)	II(3)
H	$C^n_{BV}+\alpha(1,H)-\sigma(H)$, $Q-C^n_{BV}-\alpha(1,H)$	$C^n_{BV}+\alpha(2,H)-\sigma(H)$, $Q-C^n_{BV}-\alpha(2,H)$	$C^n_{BV}+\alpha(3,H)-\sigma(H)$, $Q-C^n_{BV}-\alpha(3,H)$
L	$C^n_{BV}+\alpha(1,L)-\sigma(L)$, $Q-C^n_{BV}-\alpha(1,L)$	$C^n_{BV}+\alpha(2,L)-\sigma(L)$, $Q-C^n_{BV}-\alpha(2,L)$	$C^n_{BV}+\alpha(3,L)-\sigma(L)$, $Q-C^n_{BV}-\alpha(3,L)$

TABLE 4
Game Array between the Verifier and Its Cautious Attackers

A \ V	CBI(1)	MRI(2)	II(3)
H	$C^n_{BV}+\alpha(1,H)$ $-\sigma(H)-\beta(H)$, $Q-C^n_{BV}-\alpha(1,H)$	$C^n_{BV}+\alpha(2,H)$ $-\sigma(H)-\beta(H)$, $Q-C^n_{BV}-\alpha(2,H)$	$C^n_{BV}+\alpha(3,H)$ $-\sigma(H)-\beta(H)$, $Q-C^n_{BV}-\alpha(3,H)$
L	$C^n_{BV}+\alpha(1,L)$ $-\sigma(L)-\beta(L)$, $Q-C^n_{BV}-\alpha(1,L)$	$C^n_{BV}+\alpha(2,L)$ $-\sigma(L)-\beta(L)$, $Q-C^n_{BV}-\alpha(2,L)$	$C^n_{BV}+\alpha(3,L)$ $-\sigma(L)-\beta(L)$, $Q-C^n_{BV}-\alpha(3,L)$

Specifically, we divide attackers into two types based on their preferences. One type is hot-headed, who does not consider the possibility of traceback by the verifier, while the other is more cautious, who does some extra work such as ultilizing zombie [28] to confuse the verifier in order to protect its identity. We denote the cost of extra protections for anti-tracking as $\beta(k)$ $(k \in \{H, L\})$, and it grows as the number of false messages increases.

For hot-headed attackers, the benefit of strategy H is greater than that of strategy L. Hence, they are inclined to employ strategy H. We can get $\alpha(x,H) - \sigma(H) > \alpha(x,L) - \sigma(L)$, where x is the index of the batch identification algorithm. For cautious attackers, the extra confusion work will protect itself, and that work is more effective in strategy L. Hence, we can achieve that $\alpha(x,H) - \sigma(H) - \beta(H) < \alpha(x,L) - \sigma(L) - \beta(L)$. Since the cost of the verifier is a common knowledge, the verifier only has one type. The payoff array of hot-headed attackers is as Table 3 and that of cautious attackers is shown in Table 4.

Because both types of attackers may exist simultaneously, we use P to denote the ratio between hot-headed attackers to all attackers, and correspondingly, the ratio between cautious attackers to all attackers is $1 - P$. To analyze the game with incomplete information, we design the game tree of our model as shown in Fig. 5. From Fig. 5, we find that I-BIGM follows Theorem 4.2.

Theorem 4.2. *I-BIGM has at least one Nash equilibrium.*

Proof. Let us analyze the two different cases of our game model respectively. In the discussion below, $E_{u_V}(j)$ represents the benefit of the verifier V when it chooses the batch identification algorithm j, and $E_{u_A}(k)$ indicates the benefit of attackers A when they choose the attack strategy k.

In practice, each attacker only cares about its own utility. The preference of hot-headed attackers is strategy H

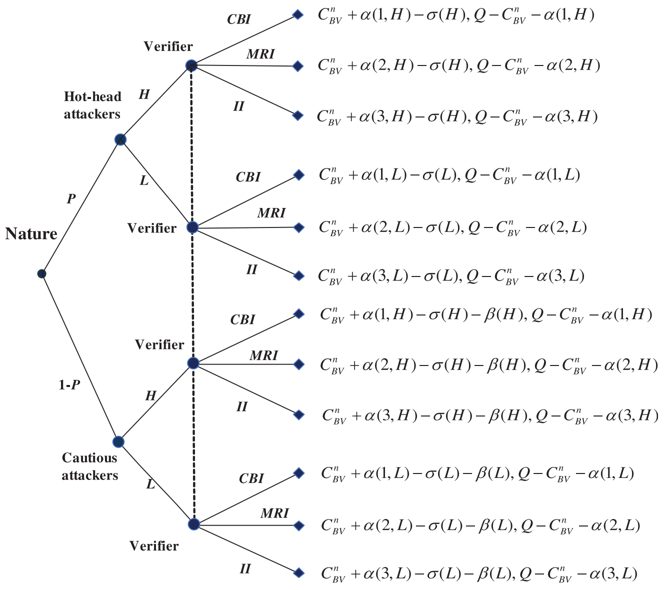

Nature
P
1-P
Hot-head attackers
Cautious attackers
H
L
Verifier
CBI
MRI
II
$C^n_{BV}+\alpha(1,H)-\sigma(H), Q-C^n_{BV}-\alpha(1,H)$
$C^n_{BV}+\alpha(2,H)-\sigma(H), Q-C^n_{BV}-\alpha(2,H)$
$C^n_{BV}+\alpha(3,H)-\sigma(H), Q-C^n_{BV}-\alpha(3,H)$
$C^n_{BV}+\alpha(1,L)-\sigma(L), Q-C^n_{BV}-\alpha(1,L)$
$C^n_{BV}+\alpha(2,L)-\sigma(L), Q-C^n_{BV}-\alpha(2,L)$
$C^n_{BV}+\alpha(3,L)-\sigma(L), Q-C^n_{BV}-\alpha(3,L)$
$C^n_{BV}+\alpha(1,H)-\sigma(H)-\beta(H), Q-C^n_{BV}-\alpha(1,H)$
$C^n_{BV}+\alpha(2,H)-\sigma(H)-\beta(H), Q-C^n_{BV}-\alpha(2,H)$
$C^n_{BV}+\alpha(3,H)-\sigma(H)-\beta(H), Q-C^n_{BV}-\alpha(3,H)$
$C^n_{BV}+\alpha(1,L)-\sigma(L)-\beta(L), Q-C^n_{BV}-\alpha(1,L)$
$C^n_{BV}+\alpha(2,L)-\sigma(L)-\beta(L), Q-C^n_{BV}-\alpha(2,L)$
$C^n_{BV}+\alpha(3,L)-\sigma(L)-\beta(L), Q-C^n_{BV}-\alpha(3,L)$

Fig. 5. The game tree of I-BIGM after Harsanyi transformation.

and that of cautious attackers is strategy L. However, in a mobile wireless network, a verifier may suffer from several cautious attackers simultaneously. As a result, the verifier may still encounter H attack strategy even if all attackers are cautious. Hence, we analyse two cases as follows.

Case 1: Two types of attackers adopt H attack strategy.

In this situation, the benefit of the verifier is as follows for algorithms CBI, MRI, and II

$$\begin{aligned} E_{u_V}(CBI) &= P(Q - C^n_{BV} - \alpha(1,H)) \\ &\quad + (1-P)(Q - C^n_{BV} - \alpha(1,H)) \\ &= Q - C^n_{BV} - \alpha(1,H) \\ E_{u_V}(MRI) &= P(A - C^n_{BV} - \alpha(2,H)) \\ &\quad + (1-P)(Q - C^n_{BV} - \alpha(2,H)) \\ &= Q - C^n_{BV} - \alpha(2,H) \\ E_{u_V}(II) &= P(Q - C^n_{BV} - \alpha(3,H)) \\ &\quad + (1-P)(Q - C^n_{BV} - \alpha(3,H)) \\ &= Q - C^n_{BV} - \alpha(3,H). \end{aligned}$$

From the analysis in Section 3, we know that $E_{u_V}(MRI)$ is the largest, and MRI is the dominant algorithm among those three algorithms in this situation.

Hence, *(Attacker: hot-headed attacker, H, cautious attacker, H, P; Verifier: MRI)* is a candidate Nash Equilibrium.

Case 2: Two types of attackers adopt different strategies.

In this case, the benefit of the verifier is displayed as follows for algorithms CBI, MRI, and II

$$\begin{aligned} E_{u_V}(CBI) &= P(Q - C^n_{BV} - \alpha(1,H)) \\ &\quad + (1-P)(Q - C^n_{BV} - \alpha(1,L)) \\ E_{u_V}(MRI) &= P(A - C^n_{BV} - \alpha(2,H)) \\ &\quad + (1-P)(Q - C^n_{BV} - \alpha(2,L)) \\ E_{u_V}(II) &= P(Q - C^n_{BV} - \alpha(3,H)) \\ &\quad + (1-P)(Q - C^n_{BV} - \alpha(3,L)). \end{aligned}$$

TABLE 5
The Conditions of Different Nash Equilibriums

Nash Equilibrium	Condition
(Attacker: hot-headed, H, cautious, L, P; Verifier: CBI)	$0 < P < \frac{\alpha(2,L)-\alpha(1,L)}{\alpha(2,L)-\alpha(1,L)+\alpha(1,H)-\alpha(2,H)}$, and $0 < P < \frac{\alpha(3,L)-\alpha(1,L)}{\alpha(1,H)-\alpha(1,L)}$
(Attacker: hot-headed, H, cautious, L, P; Verifier: MRI)	$\frac{\alpha(2,L)-\alpha(1,L)}{\alpha(2,L)-\alpha(1,L)+\alpha(1,H)-\alpha(2,H)} < P < 1$, and $\frac{\alpha(3,L)-\alpha(2,L)}{\alpha(2,H)-\alpha(2,L)} < P < 1$
(Attacker: hot-headed, H, cautious, L, P; Verifier: II)	$\frac{\alpha(3,L)-\alpha(1,L)}{\alpha(1,H)-\alpha(1,L)} < P < 1$, and $0 < P < \frac{\alpha(3,L)-\alpha(2,L)}{\alpha(2,H)-\alpha(2,L)}$

First of all, let us analyze the situation that CBI is the dominant algorithm. Then, we have $E_{u_V}(CBI) > E_{u_V}(MRI)$ and $E_{u_V}(CBI) > E_{u_V}(II)$.

We get,

$$0 < P < \frac{\alpha(2,L) - \alpha(1,L)}{\alpha(2,L) - \alpha(1,L) + \alpha(1,H) - \alpha(2,H)} \tag{6}$$

$$0 < P < \frac{\alpha(3,L) - \alpha(1,L)}{\alpha(1,H) - \alpha(1,L)}. \tag{7}$$

For attackers, due to $\alpha(x,H) - \sigma(H) > \alpha(x,L) - \sigma(L)$, the hot-headed attackers select strategy H. On the other hand, since $\alpha(x,H) - \sigma(H) - \beta(H) < \alpha(x,L) - \sigma(L) - \beta(L)$, the cautious attackers pick strategy L. Hence, *(Attacker: hot-headed, H, cautious, L, P; Verifier: CBI)* is a candidate Nash Equilibrium as long as P satisfies Equations (6) and (7).

Similarly, *(Attacker: hot-headed, H, cautious, L, P; Verifier: MRI)* and *(Attacker: hot-headed, H, cautious, L, P; Verifier: II)* also are candidate Nash Equilibriums when P has the proper value. The relationship between Nash Equilibrium and the value range of P is summarized in Table 5.

Note that the cautious attackers gain more benefit in the Nash Equilibriums of Case 2 than that of Case 1. Thus, in practice, the cautious attackers are more inclined to adopt low frequency attack strategy. In another word, if P is in the appropriate range, the Nash Equilibrium of Case 2 is the prime choice of the verifier. However, from Table 5, we find that the Nash Equilibrium may not exist if the ratio of hot-headed attackers P is in some special range. When that situation happens, the Nash Equilibrium in Case 1 will be used.

Hence, I-BIGM at least has one Nash Equilibrium. □

With the game model, the verifier can pick the dominant algorithm to identify invalid signatures. For example, assuming the ratio of hot-headed attackers P is 50 percent, $\frac{\alpha(2,L)-\alpha(1,L)}{\alpha(2,L)-\alpha(1,L)+\alpha(1,H)-\alpha(2,H)}$ is 0.4, $\frac{\alpha(3,L)-\alpha(1,L)}{\alpha(1,H)-\alpha(1,L)}$ is 0.45 and $\frac{\alpha(3,L)-\alpha(2,L)}{\alpha(2,H)-\alpha(2,L)}$ is 0.55, the Nash Equilibrium are *(Attacker: hot-headed, H, cautious, L, $P \in (0, 0.4)$; Verifier: CBI)*, *(Attacker: hot-headed, H, cautious, L, $P \in (0.55, 1)$; Verifier: MRI)* and *(Attacker: hot-headed, H, cautious, L, $P \in (0.4, 0.45)$; Verifier: II)*. The verifier would first check whether it can achieve a Nash Equilibrium from the results of Case 2. If yes, it picks the strategy as the dominant choice accordingly. Otherwise, it will use the Nash Equilibrium of Case 1. Obviously, when $P \in (0.45, 0.55)$, there is no Nash Equlibrium in Case 2. Hence, the verifier will find

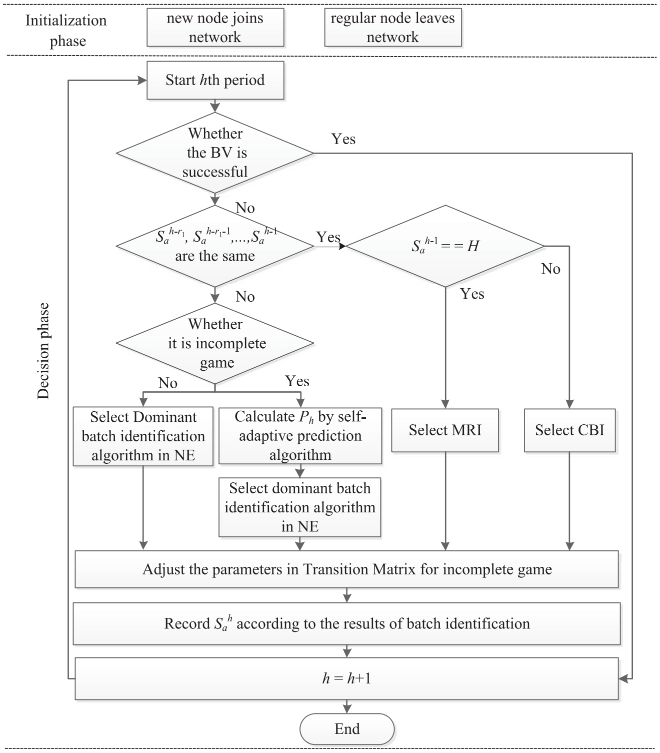

Initialization phase
new node joins network
regular node leaves network
Start hth period
Whether the BV is successful
Yes
No
$S_a^{h-r_1}, S_a^{h-r_1-1}, \ldots, S_a^{h-1}$ are the same
Yes
$S_a^{h-1} == H$
No
Yes
No
Whether it is incomplete game
No
Yes
Select Dominant batch identification algorithm in NE
Calculate P_h by self-adaptive prediction algorithm
Select MRI
Select CBI
Select dominant batch identification algorithm in NE
Adjust the parameters in Transition Matrix for incomplete game
Record S_a^h according to the results of batch identification
$h = h+1$
End
Decision phase

Fig. 6. Self-adaptive auto-match protocol process.

the strategy from the Nash Equilibrium in Case 1, and adopt MRI as the dominant algorithm.

5 Self-Adaptive Auto-Match Protocol for Algorithm Selection

With the previous discussion, we find that both game scenarios with complete information and incomplete information at least exist one Nash Equilibrium, which provides the dominant choice for the verifier. However, there still exist two challenges to be solved. The first one is that the dominant strategy may not always be the optimal choice. It works well when the verifier cannot acquire sufficient accessorial data, such as the recent number of invalid signatures. Whereas, if the verifier can observe that the number of invalid signatures has remained low during some periods, MRI algorithm, which might be the choice from Nash Equilibrium, obviously is not the optimal one. Therefore, the Nash Equilibrium is only useful when the verifier does not have any accessorial data, or attack strategy is changed frequently. The other challenge is that the estimation process of Nash Equilibrium in the game with incomplete information heavily relies on the ratio of hot-headed attackers which can be acquired only after the batch identification process is finished.

In this section, to overcome these two challenges, we formally propose a self-adaptive auto-match protocol for the selection of batch identification algorithms based on history information, which can automatically choose the suitable batch identification algorithm considering both of the analysis results of game theory and the recent behaviors of attackers. Our protocol can be implemented within two phases: (a) *Initialization phase*, where the verifier configures system parameters to resist attacks; (b) *Decision phase*, where the verifier estimates the situation, and selects the suitable batch identification algorithm. The process of protocol is illustrated as Fig. 6.

5.1 Initialization Phase

The target of this phase is to negotiate and update some shared security information for batch verification, which includes several public parameters, such as signature algorithm, hash function, public key, etc., as well as a few private parameters, such as private key and identity information. The initialization operation is triggered in two case. One is when a new regular node joins the WMN, the authority server distributes the related security information to it offline. The other is when a regular node leaves the WMN, the authority server updates and broadcasts that information to other regular nodes with signatures. Note that in this paper, we focus on identifying invalid signatures. The schemes of key distribution and management are out of our scope.

5.2 Decision Phase

After initialization phase, mobile nodes can be protected from the attacks of malicious nodes by batch verification, such as forging and tampering signatures. Due to the randomness of network attacks, batch verification must be conducted periodically. Hence, we divide time into multiple periods, and the verifier must automatically decide which batch identification algorithm is applied in each period. In this section, we focus on the first challenge how to leverage the Nash Equilibrium of game model more reasonably. The main idea is to estimate the current attack strategy based on history information. As we know, cyber attacks often concentrate outburst and malicious nodes often maintain the attack state for some time. It means that the attack strategy may not change very frequently. Thus, if the strategy of attackers remains stable in past several periods, we consider that the current strategy will be changed with a small probability. In fact, under this assumption, there indeed exist a few counter-examples which lead to unsuitable selection of batch identification algorithm. Fortunately, our protocol is keen to detect the change of attackers and respond the strategy fluctuations rapidly, when the attack strategy is uncertain. In the decision phase, first of all, the verifier tests the messages with a batch verification algorithm. If it succeeds, then nothing is done until the next period comes. Otherwise, the verifier must find the invalid signatures using a batch identification algorithm. Second, if the attack strategy is not changed in the past r_1 periods, where r_1 is configurable, the verifier can directly make a decision to select the proper algorithm according to the observed behavior history of attackers. Otherwise, the game model can provide the dominant choice for batch identification in the complete information or the incomplete information scenarios. Thus, the game model play an important role when the attack strategy is unstable and cannot be estimated by the verifier. In addition, due to the random variation of mobile nodes' states, we exploit a self-adaptive prediction algorithm described in Section 5.3 to calculate P_h which denotes the ratio between hot-headed attackers to all attackers in the hth period of our protocol. Third, no matter which batch identification algorithm is chosen, the verifier needs to collect the related parameters, such as the number of invalid signatures, the ratios of different attackers, and analyze the current attack strategy. Such information will be stored as history data for

$$M_p = \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{ccc} N & O & C \\ \left[\begin{array}{c} p_{NN} \\ p_{ON} \\ p_{CN} \end{array}\right. & \begin{array}{c} p_{NO} \\ p_{OO} \\ p_{CO} \end{array} & \left.\begin{array}{c} p_{NC} \\ p_{OC} \\ p_{CC} \end{array}\right] \end{array}\begin{array}{c} \\ N \\ O \\ C \end{array}$$

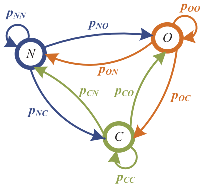


Fig. 7. The transition matrix.

next period. The pseudo-code is shown in Algorithm 3, where S_a^h indicates the attackers' strategy in period h.

Algorithm 3. Decision Phase

```
while true do
    if (Batch verification succeeds) then
        $h = h + 1$;
        continue;
    else
        if ($S_a^{h-r_1}, S_a^{h-r_1+1}, \ldots, S_a^{h-1}$) are the same then
            if ($S_a^{h-1} == H$) then
                Select MRI algorithm;
            else
                Select CBI algorithm;
            end
        else
            if Game with incomplete information then
                Calculate $P_h$ by self-adaptive prediction
                algorithm;
                Select the dominant batch identification
                algorithm with I-BIGM;
            else
                Select the dominant batch identification
                algorithm with C-BIGM;
            end
        end
    end
    Adjust the parameters in Transition Matrix;
    Record $S_a^h$ according to the results of batch identification;
    $h = h + 1$;
    if (Process should be ended) then
        Break;
    end
end
```

5.3 Self-Adaptive Prediction Algorithm

As above mentioned, the second challenge is to estimate the value of P_h. In wireless mobile networks, each node may be a potential attacker. Even if an attacker is identified by some mobile nodes, it may continue disturbing the batch verification process as a new forged identity. In our model, mobile nodes have three states: Non-threatening(N), hOt-headed (O), and Cautious(C). In detail, the non-threatening nodes include not only regular ones but also the ones identified as malicious nodes. In another word, nodes in the non-threatening state cannot launch effective attacks no matter whether they are legal. Because the transition from one state to another is a random process characterized as memoryless, these states satisfy Markov property [29]. Thus, we construct the transition matrix and transition graph in Fig. 7, while p_{ij} $(i, j \in \{N, O, C\})$ indicates the probability of transition from state i to state j in one period.

TABLE 6
The Simulation Parameters

Simulation Parameter	Value
Simulation platform	NS2
Area Size	500 m × 500 m
Node Number	100
Node Communication Range	50 m
Wireless Protocol	802.11a
Simulation Time	100 s
Batch verification algorithm	b-SPECS+ [16]
Number of messages n in a batch for verification	100

For example, p_{NO} represents the probability that a node in non-threatening state will turn to the hot-headed state. To predict more accurately, each probability value in transition matrix evolves in each period as Equation (8), while p_{ij}^h represents p_{ij} in period h, μ_{ij}^{φ} and μ_i^{φ} denote the number of transition nodes from state i to state j and the number of nodes in state i in period φ, respectively. Note that the periodicity of attacks may change, we use a configurable parameter r_2 to adjust the amount of history information.

$$p_{ij}^h = \frac{\sum_{\varphi=h-1-r_2}^{h-1} \mu_{ij}^{\varphi}}{\sum_{\varphi=h-1-r_2}^{h-1} \mu_i^{\varphi}}. \tag{8}$$

In period h, P_h, the ratio of hot-headed state nodes in all attackers, can be calculated by following equation which is jointly determined by the amount of hot-headed state and cautious state nodes

$$P^h = \frac{\sum_{i=N,O,C} \mu_i^{h-1} p_{iO}^{h-1}}{\sum_{i=N,O,C} \mu_i^{h-1} p_{iO}^{h-1} + \sum_{i=N,O,C} \mu_i^{h-1} p_{iC}^{h-1}}. \tag{9}$$

6 Performance Evalution

6.1 Simulation Configuration

In our simulations, we define and evaluate the time cost of cryptographic operations required in the batch identification. We adopt the MIRACL cryptographic library in [30], and run the simulation on an Intel Pentium IV 3.0 GHZ machine. Also, as we mentioned in Section 5.2, the batch identification is conducted every 5 seconds as one period. The other simulation parameters are presented in Table 6.

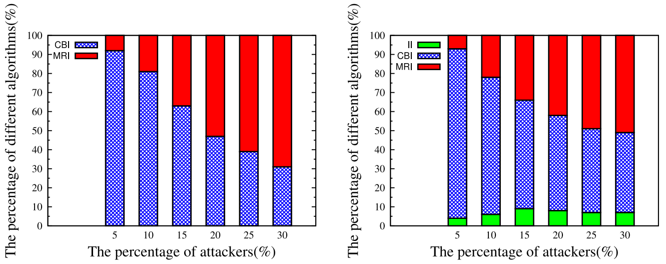
Note that our scheme is independent from batch verification algorithms. Therefore, we can equip BIGM with various batch verification algorithms [15], [16], [25], [31], even though the delay may be different, the relationship of the compared algorithms should be the same. We choose b-SPECS+ [16] as the batch verification algorithm, which calculates the processing time for a curve with embedding degree $k = 6$ and 80-bit p.

6.2 Result Analysis

In this section, we analyze the performance from three aspects.

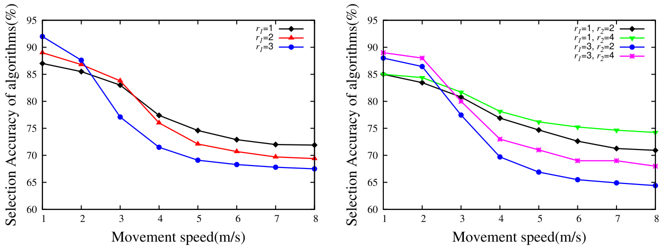
6.2.1 *The Reasonability of NE*

The percentage of a algorithm indicates the ratio between the invoking number of the algorithm to that of all algorithms. For example, assuming that ten verifiers exist in the



(a) C-BIGM (b) I-BIGM

Fig. 8. Percentage of algorithms versus percentage of attackers.



(a) C-BIGM (b) I-BIGM

Fig. 9. Selection accuracy versus movement speed.

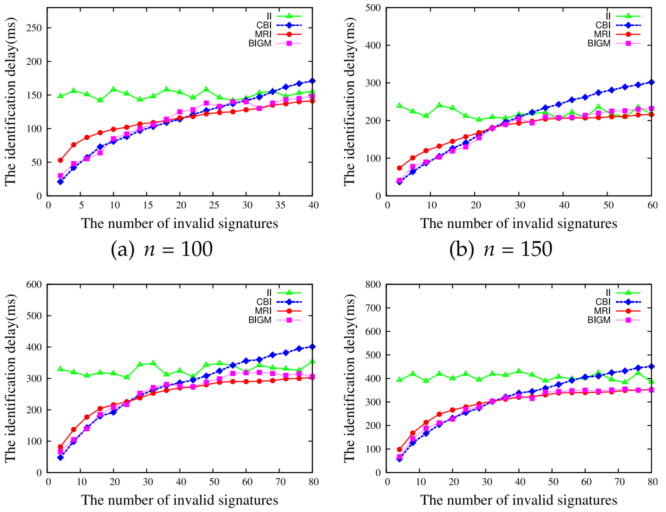
wireless mobile network, where four verifiers choose CBI as batch identification algorithm, the percentage of CBI is 40 percent. Figs. 8a and 8b show the changes of this parameter, where the game with complete information and incomplete information, individually. Form these figures, we can find that all algorithms, including II, CBI, and MRI, are used in the incomplete information scenario, but only CBI and MRI are adopted in the complete information scenario. In detail, the percentage of MRI rises from 8.1 to 69.3 percent in C-BIGM, and it increases from 7.3 to 48.6 percent in I-BIGM. While the percentage of CBI declines from 91.9 to 30.7 percent in C-BIGM, and it decreases from 89.2 to 45.4 percent in I-BIGM. In addition, in the latter scenario, the percentage of II grows up when the percentage of attackers increases at a low level, and it fluctuates in a certain range between 5 to 9 percent when attackers become more and more.

Reviewing the game model, we can find that these phenomenons fit to the NE results. In C-BIGM, the probability of H attack strategy increases accordingly with the growth of the number of attackers. Thus, more and more verifiers choose MRI as the defense strategy. The similar transition happens in I-BIGM for the same reason, while the existence of cautious attackers leads to lower probability of H attack strategy and less percentage of MRI than that in C-BIGM.

6.2.2 The Selection Accuracy of Algorithms

The *selection accuracy* of algorithms is the percentage of correct choices which lead to less identification number, such as CBI for L attack strategy and MRI for H attack strategy. Figs. 9a and 9b present the selection accuracy of algorithms with different strategy transition rates, where the parameter $r_1 = 1, 2, 3$ in complete information scenario and the parameter $r_2 = 2, 4$ in incomplete information scenario, respectively. The attack strategy is determined by all attackers around the verifier, thus, it is hard to control the transitions of attack strategies precisely. Instead, we observe the results according to the variation of nodes' movement speed. In this simulation, the selection accuracy of algorithms declines at different levels while the movement speed of mobile nodes rises. In C-BIGM, the curve with smallest r_1 has the lowest selection accuracy 87.6 percent at the beginning and the highest selection accuracy 71.9 percent at the end, while the one with largest r_1 has the highest selection accuracy 92.6 percent at the beginning and the lowest selection accuracy 67.5 percent at the end. In I-BIGM, the curves have the similar tendency while the larger r_2 introduces the higher selection accuracy. Thus, the curve with $r_1 = 3$ and $r_2 = 4$ has the highest accuracy 89.8 percent when the nodes' speed is 1 m/s.

From above analysis, we can achieve three conclusions as follows. First of all, our game model is more suitable with the low strategy transition rate. Second, if the strategy transition rate grows up, smaller r_1 is a better choice. At last, the selection accuracy of algorithm increases steadily while r_2 rises, no matter how the strategy transition rate changes.



(a) n = 100 (b) n = 150

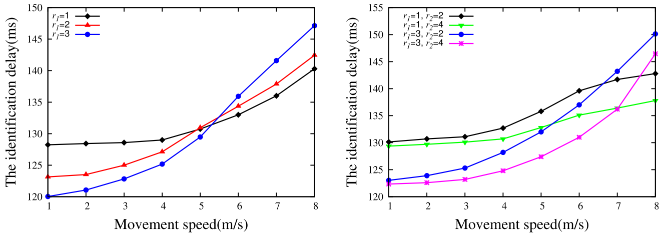
(c) n = 200 (d) n = 250

Fig. 10. Identification delay versus the number of invalid signatures.

6.2.3 The Identification Delay

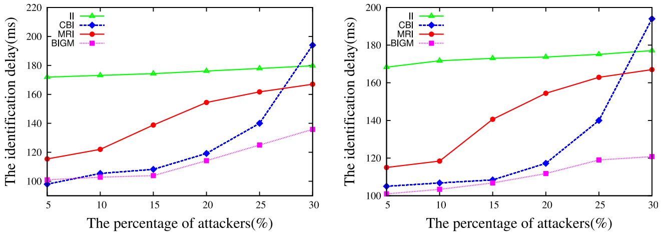
In this part, we evaluate the influence to identification delay from four aspects: *the number of invalid signatures*, *strategy transition rate*, *the percentage of attackers*, and *the ratio of hot-headed attackers*. For concise presenting, we use the *identification delay* to indicate the average delay of all verifiers in simulations.

The Number of Invalid Signatures. Figs. 10a, 10b, 10c, and 10d show the identification delay with different number of invalid signatures, where the number of sampled messages n = 100, 150, 200, 250, respectively. From these figures, we can find that the identification delay of II fluctuates in a certain range, while the other algorithms' identification delay increases. The performance of CBI algorithm is most impacted by the number of invalid signatures. CBI has the best performance when invalid signatures are few, while it has the worst performance when the number of invalid signatures goes up. The performance of MRI is more stable, and it has the lowest delay when more invalid signatures exist in the wireless mobile network. Relatively, BIGM has more stable performance compared with CBI and MRI. BIGM has an approximate performance with CBI, when the



(a) C-BIGM (b) I-BIGM

Fig. 11. Identification delay versus movement speed.



(a) C-BIGM (b) I-BIGM

Fig. 12. Identification delay versus percentages of attackers.

number of invalid signatures is small, and its performance approaches that of MRI, when the number of invalid signatures increases. It means that the identification delay of BIGM is always close to the minimum value.

The Strategy Transition Rate. From Figs. 11a and 11b, we can find the identification delay ascends with the increase of mobile nodes' speed in both games with complete information and incomplete information. In C-BIGM, when r_1 rises from 1 to 3, the curves of identification delay become steeper which means the lower head and the higher tail. The $r_1 = 1$ curve is the steepest one, where the delay varies from 128.23 to 140.29 ms with the mobile nodes' speed rising. In I-BIGM, besides the affection of r_1, the identification delay can be reduced by increasing the value of r_2. In addition, the influence of r_2 rises when the mobile nodes' speed increases. As a result, these simulations illustrate that the smaller r_1 is more suitable for frequently strategy transition and the larger r_2 will decrease the identification delay. The reason is that the smaller r_1 means that our protocol is more sensitive to strategy transitions, and the larger r_2 means that our protocol can estimate the state transitions of mobile nodes more accurately.

The Percentage of Attackers. Figs. 12a and 12b show the identification delay of different algorithms with various percentages of attackers. The identification delays of CBI, MRI, and BIGM rise obviously when attackers become intensive, while that of II fluctuates in a certain range. It is due to the constant property of II in test number. The results indicate that BIGM has the best performance no matter what the density of attackers is. Furthermore, the larger the density of attackers is, the more the advantages BIGM has, compared with other algorithms.

The Ratio of Hot-Headed Attackers. As previous mentioned, attackers can be subdivided into two classes: *hot-headed* and *cautious*. Figs. 13a, 13b, and 13c display the relationship between the identification delay and the elapsed time, where the percentages of attackers are 10, 20, and 30 percent, respectively. Furthermore, the ratios of hot-headed attackers are 30 percent in Fig. 13a, 50 percent in Fig. 13b, and 70 percent in Fig. 13c.

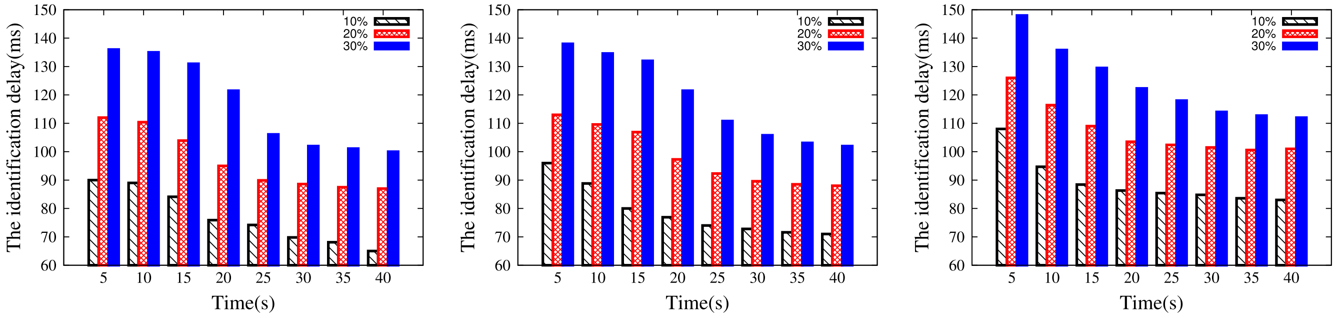
These figures illustrate that it takes several periods for BIGM to reach the Nash Equilibrium, since it is hard for verifiers to acquire the precise attack information at once. In Fig. 13, the identification delay decreases at beginning, and finally turns stable at some level. Besides, as the percentage of attackers and the ratio of hot-headed attackers increase, the time to reach the Nash Equilibrium becomes larger, and the stable value of identification delay also rises.

7 Conclusion

For selecting suitable batch identification algorithm with high efficiency, we propose a Batch Identification Game Model, named BIGM, which consists of three components. First, we analyze the performance of three generic batch identification algorithms as the defence strategies of our game model, and discuss their advantages under different attack strategies. Then, we give the definition of BIGM, and prove the Nash Equilibriums in the games with complete information and incomplete information. Finally, we design a self-adaptive auto-match protocol to improve the practicability of our game model, considering the transition possibility of attack strategy and nodes' states. From the simulations, we find that our protocol can choose more reasonable batch identification algorithm to reduce delay and ensure network QoS, under the heterogeneous and dynamic attack scenario in WMNs.

Acknowledgments

This research was supported in part by US National Science Foundation grants 1421685, 1457262, the Key Laboratory of Aerospace Information Security and Trusted Computing,



(a) 30% hot-headed attackers (b) 50% hot-headed attackers (c) 70% hot-headed attackers

Fig. 13. Identification delay versus ratio of hot-headed attackers.

Ministry of Education, the National Natural Science Foundation of China under Grant No. 61272451, 61572380, U1536204, the Major State Basic Research Development Program of China under Grant No. 2014CB340600, and the National High Technology Research and Development Program (863 Program) of China under Grant No. 2014BAH41B00. The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through the research group project No. RGP-318. The information reported here does not reflect the position or the policy of the funding agencies. The corresponding author is Kun He.

References

[1] J. Chen, K. He, R. Du, Y. Xiang, and Q. Yuan, "Dominating set and network coding-based routing in wireless mesh networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 423–433, Feb. 2015.

[2] L. Xiao, Y. Chen, W. S. Lin, and K. J. R. Liu, "Indirect reciprocity security game for large-scale wireless networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 4, pp. 1368–1380, Aug. 2012.

[3] J. Chen, K. He, Q. Yuan, R. Du, and J. Wu, "Distributed greedy coding-aware deterministic routing for multi-flow in wireless networks," *Comput. Netw.*, vol. 105, pp. 194–206, 2016.

[4] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, "Mobility-aware caching and computation offloading in 5G ultradense cellular networks," *Sensors*, vol. 16, no. 7, pp. 974–986, 2016.

[5] Y. Liu, D. Bild, R. Dick, Z. Mao, and D. Wallach, "The Mason test: A defense against Sybil attacks in wireless networks without trusted authorities," *IEEE Trans. Mobile Comput.*, vol. 14, no. 11, pp. 2376–2391, Nov. 2015.

[6] M. Chen, Y. Qian, S. Mao, W. Tang, and X. Yang, "Software-defined mobile networks security," *Mobile Netw. Appl.*, 2016. [Online]. Available: http://doi:10.1007/s11036-015-0665-5.

[7] B. Alomair and R. Poovendran, "Efficient authentication for mobile and pervasive computing," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 496–481, Mar. 2014.

[8] Q. Wang, M. He, M. Du, S. S. M. Chow, R. W. F. Lai, and Q. Zou, "Searchable encryption over feature-rich data," *IEEE Trans. Depend. Secure Comput.*, 2016. [Online]. Available: http://doi.org/10.1109/TDSC.2016.2593444

[9] L. Y. Yeh, Y. L. Huang, A. Joseph, S. Shieh, and W. Tsaur, "A batch-authenticated and key agreement framework for P2P-based online social networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 4, pp. 1907–1924, May 2012.

[10] K. He, J. Chen, R. Du, Q. Wu, G. Xue, and X. Zhang, "DeyPoS: Deduplicatable dynamic proof of storage for multi-user environments," *IEEE Trans. Comput.*, 2016. [Online]. Available: http://doi.org/10.1109/TC.2016.2560812

[11] A. Fiat, "Batch RSA," in *Proc. Advances CRYPTO*, 1989, pp. 175–185.

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

[13] J. Cheon, J. Coron, J. Kim, and M. Lee, "Batch fully homomorphic encryption over the integers," in *Proc. EUROCRYPT*, 2013, pp. 315–335.

[14] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proc. IEEE INFOCOM*, 2008, pp. 1409–1417.

[15] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. S. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, 2008, pp. 246–250.

[16] S. Horng, S. Tzeng, Y. Pan, and P. Fan, "b-SPECS+: Batch verification for secure pseudonymous authentication in VANET," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 11, pp. 1860–1875, Nov. 2013.

[17] J. Pastuszak, D. Michalek, J. Pieprzyk, and J. Seberry, "Identification of bad signatures in batches," in *Proc. 3rd Int. Workshop Practice Theory Public Key Cryptography*, 2000, pp. 28–45.

[18] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing SIFT: Privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3411–3425, Jul. 2016.

[19] S. Lee, S. Cho, J. Choi, and Y. Cho, "Efficient identication of bad signatures in RSA-type batch signature," *IEICE Trans. Fundamentals Electron. Commun. Comput. Sci.*, vol. E89-A, no. 1, pp. 74–80, Jan. 2006.

[20] L. Law and B. Matt, "Finding invalid signatures in pairing-based bathes," in *Cryptography and Coding*. Berlin, Germany: Springer, 2007, pp. 34–53.

[21] M. Stanek, "Attacking LCCC batch verification of RSA signatures," *Int. J. Netw. Secur.*, vol. 6, no. 2, pp. 238–240, 2008.

[22] B. J. Matt, "Identification of multiple invalid signatures in pairing-based batched signatures," in *Proc. 12th Int. Conf. Practice Theory Public Key Cryptography*, 2009, pp. 337–356.

[23] G. M. Zaverucha and D. R. Stinson, "Group testing and batch verification," in *Proc. IEEE 4th Int. Conf. Inf. Theoretic Secur.*, 2009, pp. 140–157.

[24] C. Zhang, P. Ho, and J. Tapolcai, "On batch verification with group testing for vehicular communications," *Wireless Netw.*, vol. 17, no. 8, pp. 1851–1865, 2011.

[25] C. Lee and Y. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Netw.*, vol. 19, no. 6, pp. 1441–1449, 2013.

[26] J. A. Akinyele, M. Green, S. Hohenberger, and M. W. Pagano, "Machine-generated algorithms, proofs and software for the batch verification of digital signature schemes," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 474–487.

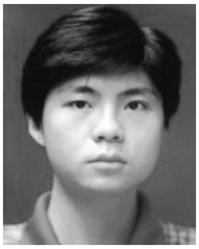
[27] J. Chen, Q. Yuan, G. L. Xue, and R. Y. Du, "Game-theory-based batch identification of invalid signatures in wireless mobile networks," in *Proc. IEEE INFOCOM*, 2015, pp. 262–270.

[28] Z. Lu, W. Wang, and C. Wang, "How can botnets cause storms? Understanding the evolution and impact of mobile botnets," in *Proc. IEEE INFOCOM*, 2014, pp. 1501–1509.

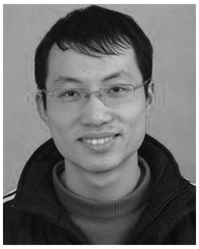
[29] Y. Ephraim and W. J. J. Roberts, "An EM algorithm for Markov modulated Markov processes," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 463–470, Feb. 2009.

[30] MIRACL Cryptographic Library: Multiprecision Integer and Rational Arithmetic C/C++ Library. [Online]. Available: https://www.certivox.com/miracl

[31] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments-security Services for Applications and Management Messages*, IEEE Std. 1609, 2006.



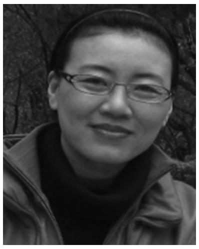
Jing Chen received the PhD degree in computer science from the Huazhong University of Science and Technology, Wuhan. He worked as an associate professor from 2010. His research interests in computer science are in the areas of network security and cloud security. He has published more than 60 research papers in many international journals and conferences, such as the *IEEE Transactions on Parallel and Distributed Systems*, INFOCOM, SECON, TrustCom, and NSS.



Kun He received the MS degree in computer science from Wuhan University, Wuhan, China, in 2011. He is working toward the PhD degree at Wuhan University. His research interests include network security, cloud security, and mobile computing.

Quan Yuan is an assistant professor in the Department of Math and Computer Science, University of Texas-Permian Basin, Texas. His research interests include network security, mobile computing, routing protocols, Peer-to-Peer computing, parallel and distributed systems, and computer networks.

Guoliang Xue received the BS degree in mathematics, the MS degree in operations research from Qufu Normal University, and the PhD degree in computer science from the University of Minnesota, in 1981, 1984, and 1991, respectively. He is a professor of computer science and engineering with Arizona State University. His research interests span the areas of Quality of Service provisioning, network security and privacy, crowdsourcing and network economics, smart city, and smart grids. He has published more than 280 papers in these areas, many of which in top conferences such as INFOCOM, MOBICOM, NDSS and top journals such as the *IEEE/ACM Transactions on Networking*, the *IEEE Journal on Selected Areas in Communications*, and the *IEEE Transactions on Mobile Computing*. He was a keynote speaker at IEEE LCN'2011 and ICNC'2014. He was a TPC co-chair of IEEE INFOCOM'2010 and a general co-chair of IEEE CNS'2014. He has served on the TPC of many conferences, including ACM CCS, ACM MOBIHOC, IEEE ICNP, IEEE INFOCOM, and IEEE GC/ICC. He served on the editorial boards of the *IEEE/ACM Transactions on Networking and* the *Computer Networks* Journal. He now serves as the area editor of the *IEEE Transactions on Wireless Communications*, overseeing 13 editors in the Wireless Networking area. He is a fellow of the IEEE, and the vice president of Conferences of the IEEE Communications Society.



Ruiying Du received the BS, MS, and PhD degrees in computer science from Wuhan University, Wuhan, China, in 1987, 1994, and 2008. She is a professor in the Computer School, Wuhan University. Her research interests include network security, wireless network, and mobile computing.

Lina Wang received the PhD degree in computer science from Northeastern University, Shenyang, China, in 1999. She is a professor in the Computer School, Wuhan University. She is the director in the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, China. Her research interests include network and system security.