Defending Against Man-In-The-Middle Attack in Repeated Games

Shuxin Li[1], **Xiaohong Li**[1], **Jianye Hao**[2*], **Bo An**[3], **Zhiyong Feng**[2],
Kangjie Chen[4] and **Chengwei Zhang**[1]
[1] School of Computer Science and Technology, Tianjin University, China
[2] School of Computer Software, Tianjin University, China
[3] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[4] Tianjin International Engineering Institute, Tianjin University, China
{lishuxin,xiaohongli,jianye.hao,zyfeng,kangjiechen,chenvy}@tju.edu.cn, boan@ntu.edu.sg

Abstract

The Man-in-the-Middle (MITM) attack has become widespread in networks nowadays. The MITM attack would cause serious information leakage and result in tremendous loss to users. Previous work applies game theory to analyze the MITM attack-defense problem and computes the optimal defense strategy to minimize the total loss. It assumes that all defenders are cooperative and the attacker know defenders' strategies beforehand. However, each individual defender is rational and may not have the incentive to cooperate. Furthermore, the attacker can hardly know defenders' strategies ahead of schedule in practice. To this end, we assume that all defenders are self-interested and model the MITM attack-defense scenario as a simultaneous-move game. Nash equilibrium is adopted as the solution concept which is proved to be always unique. Given the impracticability of computing Nash equilibrium directly, we propose practical adaptive algorithms for the defenders and the attacker to learn towards the unique Nash equilibrium through repeated interactions. Simulation results show that the algorithms are able to converge to Nash equilibrium strategy efficiently.

1 Introduction

Recent years have witnessed the development of the Internet and public concern of the potential leakage of sensitive information has been raised broadly. The Man-in-the-Middle (MITM) attack is one of the attacks that can intercept sensitive information which would result in tremendous loss in terms of privacy and finance. It is reported that 95% of HTTPS servers are vulnerable to the trivial MITM attacks [Mutton, 2016]. Some events indicate the endangerment of the MITM attacks such as the MITM attack against the GitHub in China [Martin, 2013] which leads to the damage of the servers and the leakage of information. Except the attacks against the server, the attacker also launches the MITM attacks against the Internet of Things (IoT). One notable example is the MITM attack against smart cars. The hackers may be able to access and control vehicles' basic functions, such as brakes, steering and acceleration [Simko, 2016].

Significant research efforts have been devoted to addressing the MITM attack. These conventional defense technologies are mainly divided into two categories. One is to increase the difficulty of launching the attack, such as applying complicated encryption algorithm [Albina *et al.*, 2013]. The other is to detect the attacks and take the corresponding measures. There are many detection techniques such as certificate validation [Dacosta *et al.*, 2012] and utilizing the characteristic of TCP packet [Vallivaara *et al.*, 2014]. The measures taken after detection are relatively simple such as enhancing the defense for weak points and invoking reconnection. However, the above works cannot eliminate MITM attacks completely.

Recently, a security game-theoretic model is proposed to address the MITM attack problem given that the attacks are inevitable [Li *et al.*, 2017]. This model assumes that all defenders are cooperative and are willing to sacrifice their own utilities for the sake of minimizing system-level loss. However, in real world, each defender is usually self-interested and may not have the incentive to cooperate with others at the cost of sacrificing his utility. Besides, it assumes that all defenders' strategies are known to the attacker in advance, which may be unrealistic in practice. Therefore, in this paper, we assume that all defenders are self-interested and model the strategic interaction between multiple defenders and an attacker as a simultaneous-move game instead. It is natural to adopt its Nash equilibrium as the solution concept. We theoretically show that there always exists a unique Nash equilibrium and also analyze the conditions when a unique pure strategy Nash equilibrium exists. This theoretical property is desirable since it eliminates the *equilibrium selection problem*.

However, it might be infeasible to compute Nash equilibrium strategy beforehand since each player's payoff information may not be completely available to its opponents in practice. The attacker and the defenders have to learn towards their optimal (equilibrium) strategies through repeated interactions. To this end, we propose practical learning algorithms for the defenders and the attacker respectively. Simulation results show that our learning algorithms can converge to the unique Nash equilibrium effectively. To summarize, our work contributes to the state of the art in the following aspects:

- We model the strategic interaction between the MITM attacker and multiple defenders as a simultaneous-move

[*] corresponding author

game and provide theoretical analysis of the uniqueness of Nash equilibrium. The proof also provides us a theoretical method to compute Nash equilibrium.

- We propose practical learning algorithms for the defenders and the attacker. The simulation results show that the defenders can approximate the Nash equilibrium solution, which thus can be used as a practical way of computing the optimal (equilibrium) strategy.

2 Background

2.1 Game Theory in Security

Nowadays, game theory has been used in the security area widely. Earlier works mainly focus on how to protect critical infrastructures with limited resources against physical attacks [Jain *et al.*, 2010; Shieh *et al.*, 2012; An *et al.*, 2012; Kiekintveld *et al.*, 2013]. Gradually, some research begin to employ security game theory to model cyber attacks [Laszka *et al.*, 2015; 2016; Zhao *et al.*, 2016]. These works typically consist of a Stackelberg game model in which the defender makes his move first, while the attacker chooses an optimal subset of targets to attack based on the defender's strategy.

However, a Stackelberg model is not appropriate in all cases. In some cases, a simultaneous-move game may be a better reflection of the real situation [Xu *et al.*, 2016]. In practical MITM attack scenarios, an attacker usually cannot know the defender's strategy in advance [Mishra, 2013], which leads us to believe that it would be more reasonable to model it as a simultaneous-move game and we can employ Nash equilibrium as the optimal defending strategy. There also are some research on how to design defender's strategy against different attacker under repeated interactions [Klłma *et al.*, 2014; 2015; Gutierrez and Kiekintveld, 2016]. We also propose algorithms to learn the optimal strategy through repeated interactions due to the infeasibility of computing NE directly.

2.2 Man-In-The-Middle Attack

Taking the MITM attack against the server as an example, the MITM attack is an attack where the attacker communicates with the server by disguising as the end user and communicates with the end user by disguising as the server. The information transferred between the server and the end user will be intercepted by the attacker while the user and the server do not know the existence of the attacker.

The MITM attacker prefers to intercept the data packets to obtain sensitive information. The data packets are transferred through a port and each type of services is associated with a default port. For example, port 80 is assigned for providing the web service and the user who requires the web service will communicate with the server through port 80. The attacker usually monitors the default port of a particular service and intercepts the useful information. To confuse the attacker, the technology of port hopping is proposed to defend this attack by mapping a service's port to an unused pseudo-random port [Luo *et al.*, 2014; 2015].

The conventional defense technologies against the MITM attack are mainly divided into two categories. One is to increase the difficulty of launching attack, such as adopting the complicated encryption algorithm or taking safety measures in key change [Kumar *et al.*, 2012]. The other line is to detect the attack and take countermeasures accordingly. There are many works on how to detect the attack, for example, verifying whether the server's received certificate matches the legitimate certificate [Huang *et al.*, 2014] or utilizing the difference of characteristics under the attack, such as the timestamps of TCP packet headers [Vallivaara *et al.*, 2014]. The measures taken after detection are to enhance the defense for weak points and invoke the reconnection. However, these defense approaches cannot eliminate the attacks completely.

Recently, a security game-theoretic model is proposed to address the MITM attack-defense problem given that the attacks are inevitable [Li *et al.*, 2017]. The optimal defense strategy is proposed which aims at minimizing the total loss of the system under the assumptions that the attacker can know the defender's strategy in advance and all users are cooperative. However, in practice, the defender's strategy may not be available for the attacker. Furthermore, individual users are rational and may not be willing to sacrifice their own utilities to achieve socially optimal outcome. For example, consider the case of many users who require web service and an MITM attacker who intercepts the information, the web server aims at minimizing the total loss of information leakage, while each individual user is interested in whether his own information is intercepted. To this end, we revisit the problem from a different perspective by attempting to answer the following question: what is the optimal (equilibrium) defense strategy for each self-interested defender given that the MITM attacker cannot know the defenders' strategies in advance?

3 MITM Attack-Defense Problem

Normally each type of services is associated with a default port, but a service can be provided by multiple ports using the technology of port hopping. We can classify all available ports into different groups by the type of services it can provide. For each service v, there exists a corresponding set T_v of ports available for providing service v. Here, we consider the simplified case in which all users require the same service v and the group T_v of ports can provide the service. We need to distribute the ports within group T_v to users who need the service v. To simplify the analysis, we assume that there is a bijection relationship between users and ports. Each port can be used to represent an individually rational user who uses it to communicate with the server. Therefore, we can model each port and its representing user as one entity which can also be treated as a defender. In the remainder of this paper, the terms port and defender are used interchangeably.

To protect the information from being acquired, a defender can insert some noise packets into the original packets and thus reduce the proportion of useful information in the packets to decrease the total loss of the system under the MITM attack in [Li *et al.*, 2017]. Here, we follow the setting of this work. Formally, we denote the proportion of useful information in the mixed packets which is corresponding to the probability that an attacker can obtain useful information as p and the extent of communication delay due to the noise packets as q. They are both proportional to the percentage of inserted noise packets. If the percentage of noise packets increas-

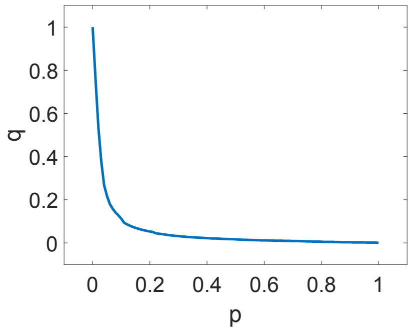


Figure 1: The relationship between p and q ($F(p)$)

es, the proportion of useful information will decrease and the communication delay will increase because more packets need to be transferred in order to send the certain quantity of useful information. We represent the relationship between p and q as the function $q = F(p), F(p) : [0,1] \mapsto [0,1]$, which can be obtained through simulation. Figure 1 shows an example of the function $F(p)$. We can see that the function $F(p)$ is a smooth and non-increasing function. Therefore, we assume that $F(p)$ is a continuous, strictly decreasing and convex function of p in the following analysis. Here, we would consider the loss of information leakage and the cost of communication delay. Let v_i denote the value of user i's information and c_i represent the loss that user i sustains caused by the communication delay. The values of v_i and c_i are relative to user i's social status and urgency degree of the task.

Therefore, the defender needs to balance the tradeoff between the information leakage and the communication delay to minimize its overall utility loss, i.e., computing the optimal value of p which is the proportion of useful information.

4 Game-theoretic Modeling

In practice, the MITM attacker cannot observe the defenders' strategies before launching the attack. Because the proportion of useful information in a package can only be known after analysing it which can only be obtained by launching a MITM attack. Therefore, it is unrealistic to adopt the Stackelberg game model in this case, while a Simultaneous-move game model is more suitable to model the interaction. Certainly, the attacker can learn the information of ports' strategies during the interactions and change his strategy towards maximizing his payoff. Therefore, the model we adopt here is a repeated simultaneous-move game. For the convenience of the analysis, Table 1 lists the symbols used in our model.

Table 1: symbols of our model

Symbol	Description
p_i	probability of getting the useful information from port i
$F(p_i)$	extent of communication delay when port i selects p_i as its strategy
v_i	information value of the user using port i
c_i	cost that the user using port i suffers from due to the communication delay
$L_i^{a_i}$	expected loss of port i given that it is attacked with probability a_i
$p_i^{a_i}$	optimal probability p of port i given that it is attacked with probability a_i
K	number of ports attacked by the attacker
S	set of attacked ports

Due to the practical resource limitations of the attacker, we assume that it can only attack a limited number of ports. Therefore, an attacker's pure strategy is to select a subset S of ports from which he intercepts valuable information ($|S| \leq K$, where K is a constant). The port i's pure strategy is to choose a p_i value which is the proportion of useful information. Let vector $\boldsymbol{P}$ denote the set of all N ports' pure strategies. Given a pure strategy profile ($\boldsymbol{p}$, S), where $\boldsymbol{p} \in \boldsymbol{P}$ and $\boldsymbol{p} = (p_1, p_2, ..., p_N)$, the attacker's payoff can be defined as

$$U_{\text{attacker}} = \sum_{i \in S} p_i v_i. \tag{1}$$

The loss (i.e., the inverse of payoff) of port i under attack is defined as

$$L_i^1 = p_i v_i + F(p_i) c_i. \tag{2}$$

If port i is not attacked, its loss is defined as

$$L_i^0 = F(p_i) c_i. \tag{3}$$

A mixed strategy of port i is a continuous probability distribution over its pure strategy (i.e., $p_i \in [0,1]$). We will prove that the best response for a port is always a pure strategy in Section 4.1, thus only pure strategies need to be considered for ports. Here, we will consider the mixed strategy for the attacker. Let A represent the set of all subsets of K ports and M denote the number of members of set A (i.e., $M = |A|$). We can define the attacker's mixed strategy as $\triangle A$ which is the probability distribution over A. Formally, $\triangle A$ can also be represented by the vector $\boldsymbol{b}$= $(b_1, ..., b_M)$ for which $b_i \geq 0$ and $\sum_{i=1}^{M} b_i = 1$. b_i represents the probability that the attacker chooses the ith subset of set A. For convenience, we use a_i to denote the probability that the attacker targets port i. Given a mixed strategy $\boldsymbol{b}$, we can compute the corresponding vector of probabilities $\boldsymbol{a}$= $(a_1, ..., a_N)$, which satisfies the constraint $\sum_i a_i = K$. For the remainder of this paper, we will represent the attacker's mixed strategy as the vector of probabilities over different ports. For a given strategy profile ($\boldsymbol{p}, \boldsymbol{a}$), the attacker's expected payoff can be expressed as

$$U_{\text{attacker}} = \sum_{i=1}^{N} a_i p_i v_i \tag{4}$$

and port i's expected loss can be represented as

$$\begin{aligned} L_i^{a_i} &= a_i L_i^1 + (1 - a_i) L_i^0 \\ &= a_i p_i v_i + F(p_i) c_i. \end{aligned} \tag{5}$$

To facilitate the analysis, we introduce the notation $p_i^{a_i}$ to represent the optimal value of p_i given that port i is attacked with probability a_i. In essence, $p_i^{a_i}$ is the value at which the minimum of $L_i^{a_i}$ is attained.

4.1 Theoretical Analysis

This section provides theoretical analysis of the model and proves the existence of a unique Nash equilibrium. The uniqueness property eliminates the equilibrium selection problem and ensures that there must exist an optimal defense strategy for the ports. We begin our analysis with a theorem on the port's best response.

Lemma 1. *The best-response strategy for a port is always a pure strategy.*

Proof. Firstly, we assume that there is a mixed strategy which is the best response for port i. The mixed strategy is a continuous probability distribution over pure strategy. Let $g(x)$ represents the probability density function of the distribution and variable X follows the distribution. Then, we construct a pure strategy by computing the expected value of the distribution as $p_i = E(X) = \int_0^1 xg(x)\,dx$. When port i plays the mixed strategy, the attacker's expected payoff of targeting port i is $\int_0^1 a_i x v_i g(x)\,dx = a_i E(X) v_i = a_i p_i v_i$. It indicates that if the port changes its strategy from the mixed strategy to the pure strategy p_i, other players' payoffs and their best responses would remain the same. Finally, given the attacker's strategy $\boldsymbol{a}$, the expected loss of port i playing the mixed strategy is $L_i^{a_i}(X) = \int_0^1 (a_i x g(x) v_i + F(xg(x)) c_i)\,dx = a_i E(X) v_i + E(F(X)) c_i$ and the loss corresponding to the pure strategy is $L_i^{a_i}(p_i) = a_i E(X) v_i + F(E(X)) c_i$. Since the function $F(p)$ is strictly convex, we have $L_i^{a_i}(p_i) < L_i^{a_i}(X)$ according to the Jensen's inequality [Chandler and Percus, 1987]. The port's loss of pure strategy is strictly less than that of mixed strategy which implies that the best response for a port must be a pure strategy. □

Next, we provide a necessary and sufficient condition for the existence of a pure strategy Nash equilibrium. Finally, we consider both pure and mixed strategy and prove that there is a unique Nash equilibrium.

Lemma 2. *The game has a pure strategy Nash equilibrium if and only there exists a set S of K ports such that $\min_{i \in S} p_i^1 v_i \geq \max_{i \notin S} p_i^0 v_i$.*

Proof. Given a attacker's pure strategy S, we know that p_i^0 is the best response for port $i \notin S$ and p_i^1 is the best response for port $i \in S$. So the port's strategy would be either p_i^0 or p_i^1 in the pure strategy NE. From equation 1, we know the attacker's best response is to select a set S of ports with the highest $p_i v_i$ values. So in pure strategy NE, the set of ports S must satisfy the condition in the Lemma 2; otherwise, the attacker's strategy would not be the best response. Based on the definition of the $p_i^{a_i}$, we know $p_i^0 > p_i^1$. The condition in the Lemma 2 can be transformed as $\min_{i \in S} p_i^1 v_i \geq \max_{i \notin S} p_i^0 v_i > \max_{i \notin S} p_i^1 v_i$. From the above inequality, we can know that there exists at most one set S satisfying the condition. Next, if set S exists, then the attacker targeting at S and the users playing their best responses p_i^0 or p_i^1 constitute obviously an equilibrium. □

Theorem 1. *There always exists a unique Nash equilibrium.*

Proof. Firstly, we provide a sufficient and necessary condition of Nash equilibrium that a strategy profile $(\boldsymbol{p}, \boldsymbol{a})$ is Nash equilibrium if and only if there exists a value λ such that for every port i,

- $a_i = 0 \Rightarrow p_i = p_i^0$ and $p_i v_i \leq \lambda$;
- $0 < a_i < 1 \Rightarrow p_i = p_i^{a_i}$ and $p_i v_i = \lambda$;
- $a_i = 1 \Rightarrow p_i = p_i^1$ and $p_i v_i \geq \lambda$.

Then we will prove it briefly. Given the attacker's strategy $\boldsymbol{a}$, we can see that every port plays its best response $p_i^{a_i}$ which is the optimal value that minimizes its expected loss. Given ports' strategies $\boldsymbol{p}$, the attacker's best response is to select a set of ports with the highest $p_i v_i$ values. There must be such a value λ that the attacker will target the ports whose values of $p_i v_i$ are larger than λ and the ports whose values of $p_i v_i$ equal λ with certain probability. Obviously, the strategy $\boldsymbol{a}$ which satisfies the above condition is the attacker's best response. Thus the strategy pair $(\boldsymbol{p}, \boldsymbol{a})$ is the best response for both sides.

Next, we define $p_i(\lambda) = \begin{cases} p_i^0 & \text{if } p_i^0 v_i \leq \lambda; \\ p_i^1 & \text{if } p_i^1 v_i \geq \lambda; \\ \frac{\lambda}{v_i} & \text{otherwise.} \end{cases}$ We can see that this function is continuous and non-decreasing in λ and the function is strictly increasing if the value of $p_i(\lambda)$ is strictly larger than p_i^1 and strictly smaller than p_i^0. We define $a_i(\lambda) = a^*$ and a^* satisfies $p_i(\lambda) = p_i^{a^*}$. Notice that $p_i^1 \leq p_i(\lambda) \leq p_i^0$ always holds and $p_i(\lambda)$ is continuous and non-increasing, thus $a_i(\lambda)$ is a continuous and non-increasing function of λ. Similarly, the function is strictly decreasing if the value of $a_i(\lambda)$ is strictly larger than 0 and strictly smaller than 1. Then, we define $E(\lambda) = K - \sum_i a_i(\lambda)$. It is obvious that $E(\lambda)$ is a continuous and increasing function of λ. If $\lambda = 0$, each $a_i(\lambda)$ will be 1 and we have $E(\lambda) = K - \text{number of ports} < 0$; and if the value of λ is so large that each $a_i(\lambda)$ equals 0, then we have $E(\lambda) = K - 0 > 0$. Therefore, we can find a value λ^* such that $E(\lambda^*) = 0$.

If the game has a pure strategy equilibrium, then there must be a gap between the Kth highest $p_i^1 v_i$ and $(K+1)$th highest $p_i^0 v_i$ according to the Lemma 2. The value of $E(\lambda)$ is 0 whenever λ is in the gap, thus λ^* is not unique, while the strategy given by $p_i(\lambda^*)$ and $a_i(\lambda^*)$ is unique (i.e., pure strategy NE). If the game does not have a pure strategy equilibrium, then λ^* is unique because $E(\lambda)$ is a continuous function.

Finally, the strategy profile given by $p_i(\lambda^*)$ and $a_i(\lambda^*)$ is a Nash equilibrium, because they satisfy the sufficient and necessary conditions of Nash equilibrium established at the beginning of the proof. Furthermore, for any λ^* value, the only strategy profile that satisfies the conditions is the one given by $p_i(\lambda^*)$ and $a_i(\lambda^*)$. There is a unique λ^* in a mixed strategy equilibrium and thus the equilibrium strategy is unique. Though the value of λ^* is not unique in a pure strategy equilibrium, the equilibrium strategy given by $p_i(\lambda^*)$ and $a_i(\lambda^*)$ is unique. Therefore, there always exists a unique NE. □

5 A Practical Learning Algorithm

The proof of Theorem 1 provides a way of computing the Nash equilibrium when we know the game's perfect information. However, in practice, it is unlikely for a port to access the perfect information of the game. Thus it is infeasible for ports to compute the Nash equilibrium using the above method. To this end, we propose a learning framework to enable each port to learn towards Nash equilibrium strategy.

The overall learning framework is described as follows. First, each port and the attacker selects their defending and attacking policies simultaneously. Second, each player plays its policy and receives the corresponding feedback information. Finally, each player updates its strategy based on the interaction information. The learning algorithms used by the ports and the attacker extend the fictitious play and the Policy Hill-climbing (PHC) algorithm respectively and will be described in detail in the following subsections.

5.1 Learning Algorithm of the Defenders

Recall that the expected loss of port i is $L_i^{a_i} = a_i p_i v_i + F(p_i)c_i$ (Equation 5). If we can get the value of a_i, then we can compute the optimal value of p_i easily since the function $F(p)$ is continuous, strictly decreasing and convex. We design the learning algorithm for the port inspired by the idea of fictitious play. The main idea of the algorithm is to predict the value of a_i and compute the optimal value of p_i based on the predicted a_i. Here, we use the frequency of being attacked in the past round to estimate the value of a_i.

Algorithm 1 The learning algorithm of port i

```
 1: Initialize $attackflag \leftarrow 0$ and $A_p \leftarrow 0$;
 2: for each round $t$ do
 3:     Choose the best defence strategy $p_i$
        $p_i \leftarrow \arg\min_{p_i} L_i^{A_p}$;
 4:     Play policy and receive the feedback of the interaction;
 5:     if port $i$ is attacked then
 6:         $attackflag \leftarrow 1$;
 7:     else
 8:         $attackflag \leftarrow 0$
 9:     end if
10:     Update the probability $A_p$
        $A_p \leftarrow A_p + \frac{1}{t}(attackflag - A_p)$;
11: end for
```

Algorithm 1 shows the learning algorithm of the defender (port i). The variable $attackflag$ is used to indicate whether port i is attacked and A_p is the estimated value of a_i. In practice, there are many different detection techniques against the MITM attack by which the port can know whether it is being attacked [Conti *et al.*, 2016]. The value of $attackflag$ is 1 if it is attacked, otherwise, the value is 0. The initial value of $attackflag$ is 0 which implies that the port is positive and believes that it will not be attacked. First, the port selects its best strategy p_i which is the value at which the minimum of $L_i^{A_p}$ is attained (Line 3). Then, the port plays its strategy and records its status of whether it has been attacked. Finally, the port updates the variable $attackflag$ and A_p based on the interaction information (Lines 5-10).

5.2 Learning Algorithm of the Attacker

From the attacker's expected payoff function shown in Equation 1 and the constraint condition $\sum_i a_i = K$, we know that the attacker's myopic best response is to set a high value of a_i to the port who has the highest value of $p_i v_i$. However, the attacker cannot get the value of $p_i v_i$ in advance, thus he maintains a Q-value for each port i to estimate the expected payoff when he attacks port i. Here, we use a_i to denote the probability that the attacker selects port i to launch attack and adopt the policy gradient-based learning algorithm to adjust a_i according to the Q-values.

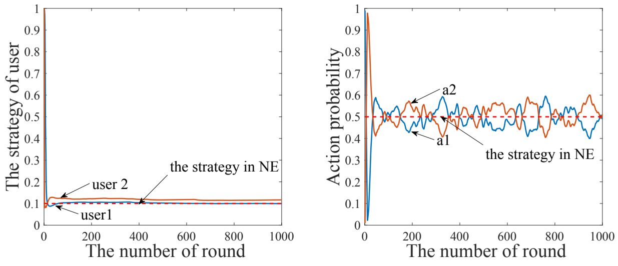
Algorithm 2 The learning algorithm of the attacker

```
 1: Initialize $a_i \leftarrow \frac{K}{N}$ and $Q_i \leftarrow 0$;
 2: for each round $t$ do
 3:     Choose the set $S$ of port based on the vector $\boldsymbol{a}$;
 4:     Play policy and receive the feedback;
 5:     for each port $i$ do
 6:         if $i \in S$ then
 7:             $Q_i \leftarrow (1-\alpha)Q_i + \alpha r_i$;
 8:         end if
 9:     end for
10:     Select the set $C$ of $K$ ports with the highest $Q$-value;
11:     Select the set $D$ of $K$ ports with the lowest $Q$-value;
12:     Update $\boldsymbol{a}$ and constrain it to a legal probability distribution;
        $a_i \leftarrow \begin{cases} a_i + \delta & i \in C \text{ and } i \notin D \\ a_i - \delta & i \in D \text{ and } i \notin C \\ a_i & \text{otherwise} \end{cases}$;
13: end for
```

Algorithm 2 depicts the learning algorithm of the attacker. We use vector $\boldsymbol{a}$ to represent the attacker's mixed strategy and initialize a_i with $\frac{K}{N}$, which means that the attacker targets all ports without preference originally. First, the attacker selects a set S based on vector $\boldsymbol{a}$ (Line 3). Then, the attacker launches attacks against the ports which are in set S. The Q-values are updated based on the feedback received in the current round (Lines 5-9). Finally, the attacker adjusts vector $\boldsymbol{a}$ according to the Q-values (Lines 10-12). The update of vector $\boldsymbol{a}$ is similar to the Policy Hill-climbing (PHC) algorithm. In PHC algorithm, the value of a_i will be increased if its corresponding value of Q_i is the highest and others will be decreased. However, we select a set C of K ports with the highest Q-value and a set D of K ports with the lowest Q-value. The a_i value of port i in set C will be increased and it will be decreased if port i in set D. The a-values for the rest of ports would remain unchanged. Meanwhile, the value of vector $\boldsymbol{a}$ must be constrained to follow a legal probability distribution (i.e., $0 \leq a_i \leq 1, \sum_i a_i = K$).

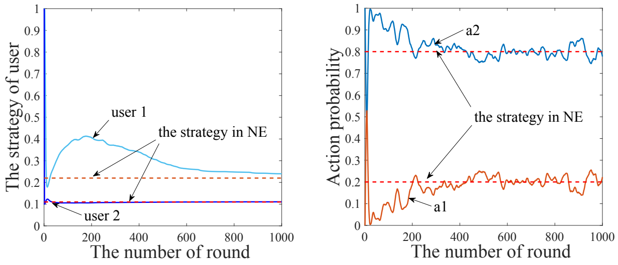
6 Experimental Evaluation

This section evaluates the performance of the algorithms described in Section 5 through comparing with the theoretical Nash equilibrium strategy. Before the experiment, the relationship between p and q should be obtained through simulation. Intuitively, if there are no noise packets inserted ($p = 1$), then there should be no communication delay ($q = 0$). On the contrary, if all packets are noise ($p = 0$), the communication delay must be the maximum and we set the value of q to 1 in this case. As for those non-extreme situations, we compute the value of q according to communication time which is relative to communication delay. Suppose that the quantity of useful packets is constant, we only need to record communication times by varying p within range of [0,1]. The function $F(p)$ can be obtained after normalizing communication times to the range of [0,1], which is illustrated in Figure 1.



(a) Users' strategies (b) Attacker's strategy

Figure 2: The case of two same users



(a) Users' strategies (b) Attacker's strategy

Figure 3: The case of two different users

In the following experiments, all the results are averaged over 100 runs. For the sake of exposition, we start with a simple case of two users and one attacker who can only attack a single user. In this case, we consider the following two situations.

1) We assume that the profiles of two users are the same. The values of v and c are defined as follows: $v_1 = v_2 = 2$, $c_1 = c_2 = 1$. Intuitively, we know that the attacker's best response is to attack each user with the same probability. We can compute mixed-strategy Nash equilibrium $(p_1 = 0.1, p_2 = 0.1, a_1 = 0.5, a_2 = 0.5)$ and the expected payoff of the attacker in Nash equilibrium is 0.2. Figure 2 shows that each user's strategy converges to the strategy in Nash equilibrium quickly and stabilizes on it finally. The attacker's strategy fluctuates up and down near the equilibrium. Though the attacker's strategy cannot stabilize, his expected payoff (0.2033) obtained in experiment approximates the expected payoff in Nash equilibrium efficiently.

2) Next, we consider the case of two different users. Let $v_1 = 1, v_2 = 2, c_1 = 1$ and $c_2 = 2$. At the beginning, the attacker may target user 2 with a higher probability since user 2 has the higher information value. Thus, user 1 may relax his defense strategy and there is an ascent process as shown in Figure 3. Finally, we can see that the users' strategies converge to the NE. The strategy of attacker fluctuates around the equilibrium and the expected payoff of attacker (0.2240) approximates the expected payoff in NE (0.22).

Table 2: The results of user's strategy

User	Theoretical	Experimental
user 1	0.2073	0.2176
user 2	0.1756	0.1764
user 3	0.1702	0.1726
user 4	0.1603	0.1654
user 5	0.2006	0.2105
user 6	0.2342	0.2451
user 7	0.2277	0.2320
user 8	0.2265	0.2248

Next, we consider a general case in which there are eight users and one attacker who can attack two users at the same time. We assume that users' information value v_i follows the power law distribution. The motivation behind this hypothesis is the hierarchical structure in organizations [Griffin, 2016]. It indicates that most users have low social status and very few have high social status. The value of c_i is related to urgency degree of the task. Here, we also assume that the value of c_i follows the power law distribution.

Table 2 presents the theoretical value and the experimental results of users' strategies which are obtained at 1000 round. The error will be smaller with the increase of rounds since a_i value that the user estimates will be more accurately when using more historic information. The differences between the final converged value and the true NE are statistically insignificant when the significance level is 5%. As for the attacker, we find that the attacker's strategy still fluctuates around the equilibrium but his expected payoff (2.9040) at 1000 round is close to the expected payoff (2.8642) in NE and the difference between them is also no significant at 5% level.

In summary, our learning algorithms can ensure the users converge to NE and the expected payoff of attacker approximates the expected payoff in NE within an acceptable error range. One deficiency may be that it requires hundreds of interactions before convergence in some cases. However, the user's average reward can quickly stabilize around a high level within dozens of rounds, though its strategy has not yet converged to the true NE. It implies that its strategy could perform well after a small number of interactions.

7 Conclusion

We solve the MITM attack-defense problem under a repeated simultaneous-move game model aiming at minimizing the defenders' own loss. Nash equilibrium is adopted as the optimal defense strategy for the defenders. We also provide theoretical analysis of the uniqueness of Nash equilibrium which addresses the equilibrium selection problem. Since the user cannot compute the Nash equilibrium directly, we propose learning algorithms for the defenders and the attacker to learn towards the Nash equilibrium. Simulation results show that our learning algorithm can converge to Nash equilibrium efficiently. The attacker's strategy fluctuates within a certain range but his expected payoff approximates the expected payoff in Nash equilibrium.

Acknowledgements

This work has partially been sponsored by the National Science Foundation of China (No. 61572349, No. 61272106), Tianjin Research Program of Application Foundation and Advanced Technology (No.:16JCQNJC00100).

References

[Albina *et al.*, 2013] Miss. N. Albina, U.J. Raju, G. K. Revathi, and K. Raghava Rao. Protection against man-in-the-middle attack in banking transaction using steganography. *International Journal of Scientific & Engineering Research*, pages 457–464, 2013.

[An *et al.*, 2012] Bo An, Eric Shieh, Rong Yang, Milind Tambe, Craig Baldwin, Joseph Direnzo, Ben Maule, and Garrett Meyer. Protect - a deployed game-theoretic system for strategic security allocation for the united states coast guard. *Ai Magazine*, 33(4):96–110, 2012.

[Chandler and Percus, 1987] David Chandler and Jerome K. Percus. *Introduction to Modern Statistical Mechanics*. Oxford University Press, 1987.

[Conti *et al.*, 2016] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials*, 18(3):2027–2051, 2016.

[Dacosta *et al.*, 2012] Italo Dacosta, Mustaque Ahamad, and Patrick Traynor. Trust no one else: Detecting mitm attacks against ssl/tls without third-parties. In *European Symposium on Research in Computer Security*, pages 199–216. Springer, 2012.

[Griffin, 2016] Dana Griffin. A hierarchical organizational structure. *http://smallbusiness.chron.com/hierarchical-organizational-structure-3799.html*, 2016.

[Gutierrez and Kiekintveld, 2016] Marcus Paul Gutierrez and Christopher Kiekintveld. Bandits for cybersecurity: Adaptive intrusion detection using honeypots. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, pages 165–166, 2016.

[Huang *et al.*, 2014] Lin Shung Huang, Alex Rice, Erling Ellingsen, and Collin Jackson. Analyzing forged ssl certificates in the wild. In *IEEE Symposium on Security and Privacy*, pages 83–97, 2014.

[Jain *et al.*, 2010] Manish Jain, Jason Tsai, James Pita, Christopher Kiekintveld, Shyamsunder Rathi, Milind Tambe, Ordonez, and Fernando Ez. Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces*, 40(4):267–290, 2010.

[Kiekintveld *et al.*, 2013] Christopher Kiekintveld, Towhidul Islam, and Vladik Kreinovich. Security games with interval uncertainty. In *AAMAS*, pages 231–238, 2013.

[Klłma *et al.*, 2014] Richard Klłma, Christopher Kiekintveld, and Viliam Lisy. Online learning methods for border patrol resource allocation. In *International Conference on Decision and Game Theory for Security*, pages 340–349, 2014.

[Klłma *et al.*, 2015] Richard Klłma, Viliam Lisy, and Christopher Kiekintveld. Combining online learning and equilibrium computation in security games. In *International Conference on Decision and Game Theory for Security*, pages 130–149, 2015.

[Kumar *et al.*, 2012] C. Krishna Kumar, G. Jai Arul Jose, C. Sajeev, and C. Suyambulingom. Safety measures against man-in-the-middle attack in key exchange. *Journal of Engineering & Applied Sciences*, 7(2):243–246, 2012.

[Laszka *et al.*, 2015] Aron Laszka, Yevgeniy Vorobeychik, and Xenofon D Koutsoukos. Optimal personalized filtering against spear-phishing attacks. In *AAAI*, pages 958–964, 2015.

[Laszka *et al.*, 2016] Aron Laszka, Jian Lou, and Yevgeniy Vorobeychik. Multi-defender strategic filtering against spear-phishing attacks. In *AAAI*, pages 537–543, 2016.

[Li *et al.*, 2017] Xiao Hong Li, Shu Xin Li, Jian Ye Hao, Zhi Yong Feng, and Bo An. Optimal personalized defense strategy against man-in-the-middle attack. In *AAAI*, pages 593–599, 2017.

[Luo *et al.*, 2014] Yue Bin Luo, Bao Sheng Wang, and Gui Lin Cai. Effectiveness of port hopping as a moving target defense. In *SecTech*, pages 7–10, 2014.

[Luo *et al.*, 2015] Yue Bin Luo, Bao Sheng Wang, and Gui Lin Cai. Analysis of port hopping for proactive cyber defense. *International Journal of Security & Its Applications*, 9(2):123–134, 2015.

[Martin, 2013] Martin. China, github and the man-in-the-middle. *https://en.greatfire.org/blog/2013/jan/china-github-and-man-middle*, 2013.

[Mishra, 2013] Praveen Mishra. Analysis of mitm attack in secure simple pairing. *Journal of Global Research in Computer Science*, 4(2):42–45, 2013.

[Mutton, 2016] Paul Mutton. 95% of https servers vulnerable to trivial mitm attacks. *https://news.netcraft.com/archives/2016/03/17/95-of-https-servers-vulnerable-to-trivial-mitm-attacks.html*, 2016.

[Shieh *et al.*, 2012] Eric Anyung Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. Protect: An application of computational game theory for the security of the ports of the united states. In *AAAI*, pages 2173–2179, 2012.

[Simko, 2016] Christian Simko. Man-in-the-middle attacks in the iot. *https://www.globalsign.com/en/blog/man-in-the-middle-attacks-iot/*, 2016.

[Vallivaara *et al.*, 2014] Visa Antero Vallivaara, Mirko Sailio, and Kimmo Halunen. Detecting man-in-the-middle attacks on non-mobile systems. In *ACM Conference on Data and Application Security and Privacy*, pages 131–134, 2014.

[Xu *et al.*, 2016] Haifeng Xu, long Tran-Thanh, and Nicholas R Jennings. Playing repeated security games with no prior knowledge. In *AAMAS*, pages 104–112, 2016.

[Zhao *et al.*, 2016] Mengchen Zhao, Bo An, and Christopher Kiekintveld. Optimizing personalized email filtering thresholds to mitigate sequential spear phishing attacks. In *AAAI*, pages 658–665, 2016.