A Dynamic Bayesian Security Game Framework for Strategic Defense Mechanism Design

Sadegh Farhang[1], Mohammad Hossein Manshaei[2],
Milad Nasr Esfahani[2], and Quanyan Zhu[1]

[1] Department of Electrical and Computer Engineering,
Polytechnic School of Engineering, New York University, New York, USA
[2] Department of Electrical and Computer Engineering,
Isfahan University of Technology, Isfahan, Iran
{farhang,quanyan.zhu}@nyu.edu,
{manshaei,m.nasresfahani}@cc.iut.ac.ir

Abstract. In many security problems, service providers are basically unaware of the type of their clients. The client can potentially be an attacker who will launch an attack at any time during their connections to service providers. Our main goal is to provide a general framework for modeling security problems subject to different types of clients connected to service providers. We develop an *incomplete information two-player* game, to capture the interaction between the service provider (i.e., the server) and an unknown client. In particular, we consider two types of clients, i.e., attacker and benign clients. We analyze the game using *perfect Bayesian Nash equilibrium* (PBNE) with different conditions. We finally design an algorithm using the computed PBNE strategy profiles to find the best defense strategy.

1 Introduction

With the rapid deployment of new computing and networking technologies and services, we are witnessing different types of clients having access to service providers via different communication infrastructures, such as the Internet. Service providers (e.g., servers in the Internet) are generally unaware of the type of their clients. These clients could be benign (legitimate) or attacker (malicious). Moreover, there exists different malicious clients with different goals and abilities. This includes but not limited to hackers, crackers, malicious insiders, industrial spy, cybercriminals, hacktivist, and cyber terrorist. In summary, in many security problems the identity of a client is unknown to the server.

Note that if a server only considers legitimate clients (i.e., optimistic point of view) to design its defense mechanism, attackers would breach to the system easily. But the server can provide a good quality of services to benign clients in such cases. On the other hand, if the server assumes that each client is potentially an attacker (i.e., pessimistic point of view), it would degrade the quality of services for the connected benign clients. Therefore, to design optimal defense

R. Poovendran and W. Saad (Eds.): GameSec 2014, LNCS 8840, pp. 317–326, 2014.

mechanism that prevents malicious activities and provides good quality of services to benign clients, we should consider both types of clients simultaneously. Game theory is an appropriate tool that can be used to deal with such problems.

Game theory has been used widely to tackle security issues in computer and communication networks [7,4]. Most security problems are usually modeled between a defender (i.e., server) and an attacker (i.e., client), where the identity of the players is clearly distinguished. However, it is not always possible to assume that the identity of client (i.e., benign or malicious) is known to the server [6,12,10,5]. Game theory enables the server to model its interaction with clients whose identities are unknown to the server [2,13,8]. The main goal of this paper is to propose a new class of security games that can be used to model the interactions between a server and its client which can be either an attacker or a benign client. By using *multi-stage games with observed action and incomplete information*, we capture uncertainties that are dynamically evolving in this type of security problems. This leads to the definition of *perfect Bayesian Nash equilibrium* concept. We apply the computed PBNE to identify server's uncertainty about its clients. Furthermore, we propose the mechanism for the server to prevent the malicious activities of the attacker client as well as provide good quality service to the benign client.

Bayesian games have been used to model the uncertainties of one player about its opponent. In [10], Parunchuri et al. consider Bayesian Stackelberg games to model airport security problem, in which the leader is uncertain about the types of adversary. In this model, leader assigns prior probability to each type of adversary, i.e., follower. However, during the game, these prior probabilities will remain constant. Our model modifies the belief of defender based on the clients' behavior during the game. As an example in network security, Liu et al. [3] use Bayesian games to model the interaction between defender and the connected node that can be malicious or regular in wireless ad hoc network. In this model, the best strategy of defender is computed by only considering malicious nodes. In our model, we consider both types of clients in finding the server's best defense strategy.

This paper is organized as follows. In Section 2, we propose our system model. We analyze the game in Section 3, followed by protocol design in Section 4. Finally, we conclude the paper in Section 5.

2 System Model

In this section, we propose our model for security games when a server is uncertain about type of a client. We name this security game as G^S. Game theory enables us to deal with the lack of knowledge about the identity of players [9]. As shown in Fig. 1, security game G^S is a *two-player* game with a server $\mathcal{S}$ and client $\mathcal{C}$ as players. Each player $i \in \{\mathcal{S}, \mathcal{C}\}$ has a type θ_i in a finite set Θ_i.

In the security game G^S, we consider two types ($\theta_C = 0$ denotes benign client and $\theta_C = 1$ denotes attacker one) for clients of our server. Indeed, in our game the server type is always $\theta_S = 0$. The nature of communications between server

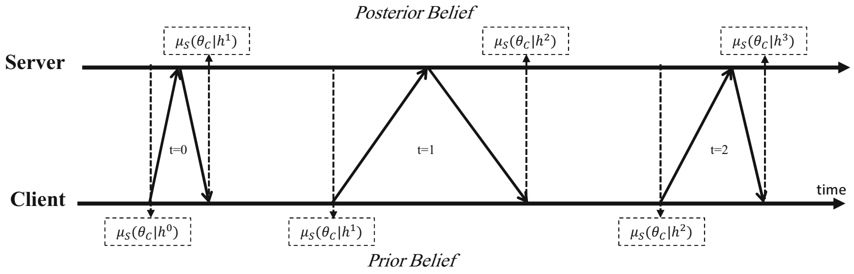

Fig. 1. G^S is a two-player repeated game between server and client. Client could be either benign or attacker. At each stage (with various duration), the server updates its belief about the type of its client based on the client's current action and the history.

and client is repeated and played at stages $t = 0, 1, 2, \ldots, T$. Different packets are sent by the client to the server and vice versa. Each packet and the corresponding response can be considered as one stage of the security game G^S.

We model security game G^S using *multi-stage games with observed actions and incomplete information* [1] to deal with incomplete information about client $\mathcal{C}$'s type. Our security game G^S is perfect information since each player can observe the action of another player. Moreover, the server does not know the identity of client. So, security game G^S is incomplete information. Note that different G^S games are played in parallel when different clients are connected to the server at the same time.

In G^S game, h^t is the history at the beginning of stage t. History denotes actions of all players in all previous stages until stage t. h^0 is the history at the beginning of the game. h^1 is the history at the end of stage zero and denotes actions of both players at stage zero. Similarly, h^2 represents the actions of both players at stages zero and one.

The server has a prior knowledge about the type of its clients, i.e., $\mu_S(\theta_C|h^0)$. In other words, this prior knowledge is the belief of server about type of its clients at the beginning of the game. In general, $\mu_S(\theta_C|h^t)$ is the belief of server about the type of player $\mathcal{C}$ at time t. When $t > 0$, this belief is also called posterior probability. This belief could be potentially updated at each stage. Players consider history, action of other players at this stage and the belief at the previous stage to update their beliefs about their opponents. Bayes' rule is used to update belief at the end of each stage. In stage zero, server uses its prior knowledge besides client's action to update its belief at the end of the stage. In stage 1, server updates its belief using Bayes' rule, $\mu_S(\theta_C|h^0)$, h^1, and client's current action. In other stages, for example stage t, server updates its belief similar to stage 1, i.e., using Bayes' rule, $\mu_S(\theta_C|h^{t-1})$, h^t, and client's current action. In the rest of this section, we first define both players' strategies and then calculate players' payoffs.

In our security game G^S presnted in Table 1, the strategy set of player $\mathcal{C}$, i.e., client, is limited to *Greedy* and *Normal*, i.e., $s_C = \{Greedy, Normal\}$, and the server should only select between two strategies of *Defend* or *Not Defend*, i.e., $s_S = \{Defend, Not\,D-\ \ efend\}$. Let's explain how strategies can be defined with two security examples.

Table 1. Strategic Form of Security Game G^S

Player $\mathcal{S}$/ Player $\mathcal{C}$	**Normal**	**Greedy**
Not Defend	$0, 0$	$0, 0$
Defend	$-\alpha, -\beta$	$-\alpha, -\beta$

Player $\mathcal{C}$ is Benign

Player $\mathcal{S}$/ Player $\mathcal{C}$	**Normal**	**Greedy**
Not Defend	$-G^{'}, G^{'} - \tau^{'}$	$-G, G - \tau$
Defend	$g^{'} - G^{'} - \alpha, G^{'} - g^{'} - \tau^{'}$	$g - G - \alpha, G - g - \tau$

Player $\mathcal{C}$ is Attacker

The first example is Intrusion Detection Systems (IDS) in mobile ad hoc network which has been presented in [6]. In this case, *Greedy* strategy means that client sends more packets than a certain threshold to the server. From server's point of view, *Defend* strategy could be interpreted as *monitoring* the client. The Second example is password reset which is presented in [11]. In this example, *Defend* strategy is *moving* into a new state. In other words, server changes the password because the attacker might have penetrated to its system. Moreover, *Greedy* strategy means that client tries to penetrate server's system by examining different passwords.

Our security game G^S is a two-player repeated game between a server and a client as players. The server is uncertain about type of its client which could be either benign or attacker. We assume that the players' identities remain consistent throughout the game. To calculate server's payoff, we consider a constant cost for its *Defend* strategy, i.e., α, regardless of the type of player $\mathcal{C}$. On the other hand, we do not consider a cost for server's strategy of *Not Defend.*

Benign type of player $\mathcal{C}$, i.e., $\theta_C = 0$, might play *Greedy.* If the server is smart enough, it will not play *Defend.* Server sometimes plays *Defend* against benign type of player $\mathcal{C}$ due to lack of sufficient information. This server's action leads to degradation in service or problem in communication to the benign client. We quantify this degradation or problem by β. Note that there does not exist any difference between *Greedy* strategy and *Normal* strategy of benign type of player $\mathcal{C}$. Because, the goal of server is to provide good quality service to benign clients.

Following our discussion in payoff calculation, now we consider attacker type of player $\mathcal{C}$, i.e., $\theta_C = 1$. We represent attacker's cost of playing *Greedy* and *Normal* by τ and $\tau^{'}$, respectively. In some cases, playing *Normal* leads to spend more time to launch successful attack. Therefore, alarming tools, such as IDS, will be suspicious about type of this client. So, we do not consider $\tau^{'} = 0$ and we assume that $\tau^{'} \geq \tau$.

If every step of game is done completely, the attack will be successful. Some steps of attack are necessary for successful attack. Let us assume that each step of attack gives attacker fraction of information for successful attack. We define G as the information that is gained by attacker when attacker plays *Greedy.* Similarly, we define $G^{'}$ as the attacker's gain of information when it plays *Normal.* One can simplify the model by not considering the attacker's gain of information when playing *Normal* strategy. Note that, attacker's gain of information when playing *Greedy* is not lower than attacker's gain when playing *Normal*, i.e., $G \geq G^{'}$.

In our security game G^S, playing *Defend* has a cost for server as well as prevents leakage of information to the attacker. This prevention must be different with respect to attacker's strategy. So, we define g and $g^{'}$ as prevention from information's leakage when attacker's strategies are *Greedy* or *Normal*, respectively. We assume that $g \geq g^{'}$.

3 Security Game Analysis

In this section, we analyze the security game G^S to propose optimum probability of playing *Defend* strategy for the server where it is uncertain about type of its client. We find the best responses of players in Lemmas 1, 2, and 3. Furthermore, Conjecture 1 shows that how the server can distinguish between the attacker or the benign type of player $\mathcal{C}$ by using Bayes' rule.

First, we show that four requirements in the definition of PBNE, i.e., B(i)-B(iv), are satisfied for our security game G^S (Please see [1] for the definition of PBNE and its four requirements). B(i) is satisfied, because server has one type. B(ii) is satisfied, since we use Bayes' rule to update server's belief. The action of the server does not have any impact on the belief of the server about the type of its client. In other words, $\mu_S(\theta_C|h^t)$ is just affected by the action of client $\mathcal{C}$. Therefore, B(iii) is also satisfied. Finally, B(iv) is satisfied, because this game is a two-player game.

Let us define the following parameters to simplify the representations of the actions' probabilities given players' actions as well as the type of clients:

$$\begin{aligned} r_0 &:= \sigma_S(a_S^t = Defend|h^t, \theta_C = 0) \\ r_1 &:= \sigma_S(a_S^t = Defend|h^t, \theta_C = 1) \\ q &:= \sigma_C(a_C^t = Greedy|h^t, \theta_C = 0) \\ p &:= \sigma_C(a_C^t = Greedy|h^t, \theta_C = 1) \\ r &:= \sigma_S(a_S^t = Defend|h^t) \end{aligned} \tag{1}$$

Let's first assume that the server knows the type of its client. Lemma 1 shows the best strategy of the server in such cases (All proofs can be found in Appendix A.).

Lemma 1 *In our security game G^S, if the server knows that its client is an attacker, then it defends with probability equal to r_1^* given in Table 2. Otherwise, it does not defend, i.e., $r_0^* = 0$.*

Lemma 1 identifies five different cases, given that the server knows that its client is attacker. In each case, it shows that how the server plays *Defend*.

- **Cheap Defense:** in this state, server plays *Defend* in all stages against an attacker. Because the cost of playing *Defend* is lower than what server acquires when playing *Defend*, i.e., $\alpha < g^{'} \leq g$.

Table 2. The best *Defend* strategy, given different power of client and the cost of defend, when the server knows that its client is an attacker

Defense State	**Condition**	r_1^*
Cheap Defense	$\alpha < g^{'}$	1
Expensive Defense	$\alpha > g$	0
Greedy	$G - g - \tau > G^{'} - g^{'} - \tau^{'}$ $\& \; g^{'} \leq \alpha \leq g$	1
Uncertain	$G - g - \tau \leq G^{'} - g^{'} - \tau^{'}$ $\& \; g^{'} \leq \alpha \leq g \; \& \; g > g^{'}$	$\frac{(G-\tau)-(G^{'}-\tau^{'})}{g-g^{'}}$
Baffled	$G - g - \tau \leq G^{'} - g^{'} - \tau^{'}$ $\& \; g^{'} = \alpha = g$	Any Probability

- **Expensive Defense:** in this state, cost of playing *Defend* is greater than what server acquires when playing *Defend*, i.e., $\alpha > g \geq g^{'}$. Therefore, server does not play *Defend* at all.

- **Greedy:** in this state, attacker always plays *Greedy* and consequently, the server plays *Defend* in all stages.

- **Uncertain:** in this state, server plays *Defend* by certain probability. In this condition, if attacker plays *Greedy*, server will play *Defend.* Moreover, server plays *Not Defend* when attacker plays *Normal.*

- **Baffled:** in this state, there is no difference between *Normal* and *Greedy* strategy of attacker. Similarly, there is no difference between *Defend* and *Not Defend* strategy of server. Hence, server can play *Defend* by any probability.

Lemma 2 *In the security game G^S, the benign type of player $\mathcal{C}$ plays Greedy by any probability, i.e., q^*.*

Lemma 2 states that the behavior of benign type is independent from the belief of the server, as there is no difference between *Normal* and *Greedy* strategy of benign client. The server does not know the type of its client. The server has a belief about the type of its client. Attacker uses this belief to find its best response. Lemma 3 represents the best response of the attacker in different conditions.

Lemma 3 *In our security game G^S, the attacker plays Greedy with probability p^*:*

$$p^* = \begin{cases} median\,\{0, \frac{\alpha - \mu_S(\theta_C = 1|h^t)g^{'}}{\mu_S(\theta_C=1|h^t)(g-g^{'})}, 1\} & if\, g > g^{'} \\ 1 & if\, g = g^{'} \;\&\; G > G^{'} \\ 1 & if\, g = g^{'} \;\&\; \tau < \tau^{'} \\ any\, probability & if\, g = g^{'} \;\&\; \tau = \tau^{'} \;\&\; G = G^{'}. \end{cases} \tag{2}$$

Note that the attacker knows that the server is uncertain about type of its clients. Server only has a belief about the type of its client, i.e., $\mu_S(\theta_C = 1|h^t)$. Therefore, attacker uses this belief to calculate its best response, i.e., p^*. The higher the belief is, the lower the p^* is. In other words, attacker tries to decrease belief of server in the next stage by playing *Greedy* with lower probability.

Server updates its belief about the type of client at the end of each stage by using Bayes' rule. Note that server uses p^* and q^* according to Lemma 2 and Lemma 3, respectively, in Bayes' rule to update its belief.

Finally, Conjecture 1 shows the optimum strategy of server in which it is uncertain about type of its client.

Conjecture 1 *In our security game G^S, server must play Defends with probability r^*:*

$$r^* = r_1^* \mu_S(\theta_C = 1|h^t) + r_0^* \mu_S(\theta_C = 0|h^t) \tag{3}$$

Where r_1^* and r_0^* are calculated according to Lemma 1. The belief of server about the type of its client is calculated based on Bayes' rule.

Note that in Conjecture 1 contrary to Lemma 1, we consider both types of player $\mathcal{C}$ in calculating probability of playing *Defend*. In Equation (3), the server's belief weights the probability of playing *Defend* given that the server knows the identity of its client. For example, when $\mu_S(\theta_C = 1|h^t)$ is high (low), i.e., more probable that the client is attacker (benign), server plays *Defend* (r) with higher (lower) probability. In other words, the higher the $\mu_S(\theta_C = 1|h^t)$ is, the higher the r is.

4 Protocol Design

The above results provides guidelines for designing a defense mechanism named *SmartTypeDetector*, enabling the server to prevent malicious activities of the attacker while providing service to the benign clients. In other words, we employ our results for optimal *Defend* strategy presented in Conjecture 1 to compute the probability of *Defend* strategy, i.e., r. Note that, one G^S game is played for one client and different clients are independent from each other. In summary, the server finds p^* and q^* at each stage when the client plays *Greedy*. When the client plays *Normal*, the server calculates $1-p^*$ and $1-q^*$. The server uses these probabilities, Bayes' rule, and its belief in the previous stage to update its belief (update μ). The server calculates r according to Conjecture 1 where the server's belief has important influence in r.

Let's consider a situation of the game G^S between server and the attacker in which $p^* = 1$. Rational attacker will always play *Greedy*. But, irrational attacker may play *Normal* strategy in some stages. If the attacker plays *Normal* in this situation, the server's belief will be equal to zero and remain constant for the rest of the game. To avoid irrational behavior of the attacker, we also apply upper and lower bounds on p^*.

Algorithm 1. SmartTypeDetector

```
run this algorithm for each stage
if the client plays Greedy then
    find $q^*$ (i.e., Lemma 2)
    find $p^*$ (i.e., Lemma 3)
else
    find $1 - q^*$ (i.e., Lemma 2)
    find $1 - p^*$ (i.e., Lemma 3)
update $\mu$
calculate $r$ (according to Conjecture 1)
$A = rand$ (random number with uniform distribution in [0,1])
if $r \geq A$ then
    Defend
else
    Not Defend
```

5 Conclusion

In this paper, we have proposed a Bayesian security game framework to tackle with lack of knowledge about the type of the server's client. In our game-theoretic model, the game is between server and its client which could be either benign or attacker. We analyzed the game using *perfect Bayesian Nash equilibrium* concept and proposed SmartTypeDetector algorithm, based on our PBNE calculation. In this algorithm, server uses its belief about the identity of its client to determine which client is connected to the server. This framework can be applied in many security problems, such as OS fingerprinting attack and IDS. We believe that the framework is an efficient tool to model security problems in real life, where defender does not have enough information about the type of attackers.

References

1. Fudenberg, D., Tirole, J.: Game theory. MIT Press (1991)
2. Jain, M., An, B., Tambe, M.: Security games applied to real-world: Research contributions and challenges. In: Moving Target Defense II, pp. 15–39. Springer (2013)
3. Jin, X., Pissinou, N., Pumpichet, S., Kamhoua, C.A., Kwiat, K.: Modeling cooperative, selfish and malicious behaviors for trajectory privacy preservation using bayesian game theory. In: Local Computer Networks (LCN), pp. 835–842. IEEE (2013)
4. Liang, X., Xiao, Y.: Game theory for network security. IEEE Communications Surveys & Tutorials 15(1), 472–486 (2013)
5. Lin, J., Liu, P., Jing, J.: Using signaling games to model the multi-step attack-defense scenarios on confidentiality. In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 118–137. Springer, Heidelberg (2012)
6. Liu, Y., Comaniciu, C., Man, H.: A bayesian game approach for intrusion detection in wireless ad hoc networks. In: Proceeding from the 2006 Workshop on Game Theory for Communications and Networks, p. 4. ACM (2006)
7. Manshaei, M.H., Zhu, Q., Alpcan, T., Bacşar, T., Hubaux, J.-P.: Game theory meets network security and privacy. ACM Comput. Surv. 45(3), 1–39 (2013)

8. Nguyen, K.C., Alpcan, T., Basar, T.: Security games with incomplete information. In: International Conference on Communications (ICC), pp. 1–6. IEEE (2009)
9. Osborne, M.J.: An introduction to game theory, vol. 3. Oxford University Press, New York (2004)
10. Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordonez, F., Kraus, S.: Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In: Proceedings of AAMAS 2008, pp. 895–902 (2008)
11. Pham, V., Cid, C.: Are we compromised? Modelling security assessment games. In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 234–247. Springer, Heidelberg (2012)
12. Rahman, M.A., Manshaei, M.H., Al-Shaer, E.: A game-theoretic approach for deceiving remote operating system fingerprinting. In: IEEE CNS, pp. 73–81 (2013)
13. Tsai, J., Kiekintveld, C., Ordonez, F., Tambe, M., Rathi, S.: Iris-a tool for strategic security allocation in transportation networks (2009)

A Proof of Lemmas

Proof of Lemma 1: If the server knows that its client is benign, *Defend* strategy is strictly dominated by *Not Defend* strategy. So, probability of *Defend* given that the server knows that dealing with benign client is equal to zero, i.e., $r_0^* = 0$. On the other hand, when the server knows that its client is attacker, r_1^* is calculated as follows:

- **Cheap Defense** ($\alpha < g^{'} \leq g$): the server's dominant strategy is to play *Defend*, i.e., $r_1^* = 1$.
- **Expensive Defense** ($\alpha > g \geq g^{'}$): the *Defend* strategy of the server is strictly dominated by *Not Defend* strategy, i.e, $r_1^* = 0$.
- **Greedy** ($G - g - \tau > G^{'} - g^{'} - \tau^{'}$ and $g^{'} \leq \alpha \leq g$): attacker's dominant strategy is to play *Greedy*. So, the server will play *Defend*, i.e., $r_1^* = 0$.
- **Uncertain** ($G - g - \tau \leq G^{'} - g^{'} - \tau^{'}$ and $g^{'} \leq \alpha \leq g$ and $g > g^{'}$): in this condition, there does not exist any dominant or dominated strategy. To determine r_1^*, first we calculate attacker's expected payoff by playing *Normal* when the server plays its mixed strategy:

$$Eu_C[((r_1, 1 - r_1), Normal|\theta_C = 1)] = r_1(G^{'} - g^{'} - \tau^{'}) + (1 - r_1)(G^{'} - \tau^{'}) \quad (4)$$

Then, attacker's expected payoff by playing *Greedy* when the server plays it mixed strategy is:

$$Eu_C[((r_1, 1 - r_1), Greedy|\theta_C = 1)] = r_1(G - g - \tau) + (1 - r_1)(G - \tau) \quad (5)$$

We derived r_1^* by setting the Equations (4) and (5) equal:

$$r_1^* = \frac{(G - \tau) - (G^{'} - \tau^{'})}{g - g^{'}} \quad (6)$$

- **Baffled** ($G - g - \tau \leq G^{'} - g^{'} - \tau^{'}$ and $g^{'} \leq \alpha \leq g$ and $g = g^{'}$): as presented in Section 2, we assume that $G - G^{'} \geq 0$ and $\tau - \tau^{'} \leq 0$. The condition $G - g - \tau \leq$

$G^{'} - g^{'} - \tau^{'}$ could be written as $G - G^{'} \leq \tau - \tau^{'}$. The left side of this inequality is nonnegative, while the right side is nonpositive. So, both sides must be equal to zero, i.e., $G = G^{'}$ and $\tau = \tau^{'}$. In Baffled, there is no difference between server's strategies as well as attacker's. Hence, the server could play *Defend* by any probability.

Proof of Lemma 2: There is no difference between *Normal* and *Greedy* strategy of the benign client. So, the benign client could play *Greedy* by any probability.

Proof of Lemma 3: To calculate probability of playing given that the client is attacker, we consider following conditions:

- $g > g^{'}$: The server's expected payoff for playing *Defend* given that both types of its client playing their mixed strategy is calculated as:

$$\begin{aligned} Eu_S(Defend) = \mu_S(\theta_C = 1|h^t)(p(g - G - \alpha) + (1-p)(g^{'} - G^{'} - \alpha)) \\ + \mu_S(\theta_C = 0|h^t)(-\alpha) \end{aligned} \tag{7}$$

And the server's expected payoff for playing *Not Defend* when both types of its client playing their mixed strategy is:

$$Eu_S(Not\,Defend) = \mu_S(\theta_C = 1|h^t)(p(-G) + (1-p)(-G^{'})) \tag{8}$$

Note that in Equations (7) and (8), these expected payoffs are not function of q. Since, there is no difference between *Normal* and *Greedy* strategy of the benign client regardless of the server's actions.

The attacker chooses p^* to keep the server indifferent between *Defend* and *Not Defend* strategy. p^* is derived by setting Equations (7) and (8) equal, i.e.,

$$p^* = \frac{\alpha - \mu_S(\theta_C = 1|h^t)g^{'}}{\mu_S(\theta_C = 1|h^t)(g - g')} \tag{9}$$

In Equation (9), p^* is function of $\mu_S(\theta_C = 1|h^t)$. If $\mu_S(\theta_C = 1|h^t) < \frac{\alpha}{g}$, p^* is bigger than 1. In this situation, we use $p^* = 1$. Moreover, p^* is less than 0 when $\mu_S(\theta_C = 1|h^t) > \frac{\alpha}{g^{'}}$. In this situation, we use $p^* = 0$. Hence, we have $median\,\{0, \frac{\alpha - \mu_S(\theta_C = 1|h^t)g^{'}}{\mu_S(\theta_C = 1|h^t)(g - g')}, 1\}$. Where, the median of a finite list of numbers can be found by arranging all the numbers from lowest value to highest value and picking the middle one.

- $g = g^{'}$ and $G > G^{'}$: *Normal* strategy of the attacker is strictly dominated by *Greedy*, i.e., $p^* = 1$.
- $g = g^{'}$ and $\tau < \tau^{'}$: *Normal* strategy of the attacker is strictly dominated by *Greedy*, i.e., $p^* = 1$.
- $g = g^{'}$ and $\tau = \tau^{'}$ and $G = G^{'}$: there is no difference between *Normal* and *Greedy* strategy of the attacker, i.e., $p^* = Any\,probability$.