A Cost-Effective Security Management for Clouds: A Game-Theoretic Deception Mechanism

Mohammad Taghi Adili†, Amin Mohammadi†,
Mohammad Hossein Manshaei†, and Mohammad Ashiqur Rahman‡
† Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran
‡ Department of Computer Science, Tennessee Tech University, Cookeville, TN, USA
Emails: {m.adili,amin.mohammadi}@ec.iut.ac.ir, manshaei@cc.iut.ac.ir, marahman@tntech.edu

***Abstract*—The Information Technology (IT) is observing a rising shift toward cloud computing due to its attractive on-demand storage and computing capabilities that allow moving the computing and storage load from the owner's side to the service provider's place and enjoying the data or computed results efficiently anywhere anytime. This growing use of clouds also introduces significant security concerns, as sensitive data and critical applications are increasingly being moved to clouds. Recent work also reveals different security threats, e.g., side-channel attacks, against cloud services. In this work, we address the need of improved solutions for the security management of cloud computing. We propose a moving target-based deceptive defense mechanism where the moving target idea is centered on frequent migrations of the virtual machines (VMs). We make the moves cost-efficient by modeling the problem as a signaling game between the adversary and the VMs and introducing deceptions. We solve the game and obtain two Nash equilibria. These results illustrate the best possible moves by the adversary and the corresponding strategy for the VMs that should reduce the adversary's chance of being successful at most.**

***Index Terms*—Cloud computing; VM migration; moving target defense; game-theoretic analysis; deceptive strategies.**

I. Introduction

Cloud computing has become a major trend in computing services with its inspiring features of elastic "data anywhere" and "computing anywhere" [1]. Generally, there are three types of cloud services: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Among these services, the IaaS is the most fundamental one, where a cloud user owns a virtual machine (VM) and purchases necessary virtual power to execute applications on it. A typical example of public IaaS is the Amazon Elastic Compute Cloud (Amazon EC2) [2]. Although IaaS offers cost-efficient and ease-of-use services to cloud users, there are significant security concerns that need to be addressed especially when critical applications and sensitive data are moved to the clouds.

There is a broad consensus that virtualization technology can improve the security and reliability of cloud computing. This is mainly because of the seemingly strong isolation, which prevents the guest VMs located in the same host from interfering with each other. However, such logical isolation may not be sufficient [3]. By launching a side channel attack (SCA), which is firstly introduced by Kocher [4], malicious users can circumvent the isolation mechanism and extract private information from other users by analyzing responses of third party shared resources [5], [6]. If the attacker and the victim reside in the same host, SCA is called intra-host SCA. Using the shared resource in the host, such as data cache and instruction cache, an attacker can steal the private information from the victim VMs [7]–[9]. In the scenario of inter-host SCA, the attacker and the victim are not co-resident. This kind of SCA is implemented based on the network traffic, through which an attacker can steal private information from the VMs located on different hosts [10], [11].

In this paper, we focus on improving cloud security. Considering the functionality of virtualization and the flexibility of VMs, we apply a VM moving technique to provide security of the cloud against potential attacks. More specifically, we propose a deception mechanism that enhances the security of the data by means of VM migration, while reducing the migration cost using deception. When the network controller informs a VM about potential attacks, which correspond to IDS alarms, the VM can migrate to another physical system which has enough space. This migration can be done either live or non-live. Since migrating at every alarm will be highly expensive, a VM in one hand tends not to migrate but would like to be secure on the other hand.

The proposed deception technique follows a signaling game in which VMs occasionally do not migrate (neither live nor non-live) but send live or non-live migration signals to attackers. When an attacker receives this signal (e.g., a non-live migration message), it becomes confused as it does not know whether the target (VM) is going to migrate non-lively or it is just a deceptive message. From the analysis of the signaling game, we observe that this technique can significantly enhance the security of the cloud network (with respect to individual VMs), while it uses a fewer number of migrations than similar techniques. With the best of our knowledge, this work is the first of its kind that uses a deception-based moving target defense (MTD) technique for the cloud that ensure security but with reduced cost. Therefore, in summary, our contributions are as follows:

- We propose a deception-based moving target defense technique. Our proposed solution intelligently perform the VM migration to increase the security of data while keeps the migration overhead limited.
- We model the deception technique as a signaling game and

solve it to find out the equilibrium strategies with respect to the defender (the VM) and the attacker. The solution of the game results in two Nash equilibria, which help us understand the attacker's best possible strategies and the most efficient defense actions.

The paper is organized as follows. We briefly explain the related works, their limitations, and comparative study with respect to our solution in Section II. In Section III, we present the deception technique and define the parameters of the corresponding signaling game. Signaling game analysis and result extraction and explanation are presented in Section IV. The conclusion of this paper is provided in Section V.

II. Related work

A number of migration approaches have been proposed in literatures to improve the performance of cloud platforms and reduce resource consumptions. Generally, these approaches are based on either VM migration or hardware based techniques.

The authors in [12] and [13] proposed a VM placement strategy in order to optimize the propper cost. The mechanism does not consider the cloud security enhancement. In [14]–[19], authors proposed hardware-based approaches to protect user privacy in cloud environment. These hardware based methods are typically expensive to deploy.

As mentioned in [20], the VMs with shared memory pages can be optimized by placing them in the same physical servers. This effective virtual machine migration strategy can greatly improve the performance of cloud platform. VMs in [20] , can deliver data through shared memory rather than network.

Some approaches dynamically cluster VMs to improve the efficiency of the cloud and then distribute resources to different clusters [13], [21], [22]. The authors in [12] developed a generic algorithm to create a placement plan to reduce estimated total execution time. In [23] a scheduling model has been provided to optimize virtual cluster placement through cloud offers. The experimental results with real data show that dynamic placement plan can bring more benefits in reducing user's costs than the fixed one. Some approaches like [24], [25] used such placement policies in pooling scheme in order to enhance security of each unique placement technique.

A periodic migration strategy based on game theory is proposed in [12]. This mechanism makes it much harder for adversaries to locate the target VMs in terms of survivability measurement. Although VM migrations can improve security of cloud system, the major concern of their approach is periodic migrating of VMs that cannot be a good idea because of predictability of this strategy as mentioned in [26]. Another concern is about the nature of VCG (VickreyClarkeGroves) mechanism, i.e., the capability of a VM to migrate. In networks which include crucial data, when secrecy of stored information is under menace, network administrator determines which VMs have to migrate in order to reduce threats. In this situation, such a VCG mechanism is not appropriate since may endanger security of the entire network.

In [27] authors focused on co-resident attack and tries to design and improve a virtual machine policy not only to circumvent co-locating with target VMs but also to satisfy workload balancing. They defined and modeled some security metrics to assess the attack and compared difficulty of achieving co-residence under three common VM placement policies. Their simulation results comes from Openstack platform. [25] also tried to compare some basic VM placement policies for cloud computing systems in a random pooling based manner for each VM request. [28] incorporated two commonly use MTD techniques (Shuffle and Diversity) to model and analyze security using HARMs (Hierarchical Attack Representation Models) to show that MTD techniques are appropriate for security enhancement. It is worth mentioning that signaling game has already applied in different security problems, such as OS fingerprinting [29], cyber security [30], client and provider relation in cloud computing [31], and fake avatar deception in social networks [32].

III. System Model

Consider a network as shown in Figure 1, in which each physical system (i.e., node shown in figure) contains a number of virtual machines (VMs). Each VM can communicate with external users independently. Suppose that confidential information is stored on these systems (for example according to Shamir's distributed secret sharing [33]). We consider an adversary who launches several known and unknown attack techniques to steal confidential information available on these machines. For example, as mentioned in [34] and [35], side channel attacks can compromise privacy of VMs and therefore the secrecy of vital data will be violate.

We would like to deploy a *moving target defense* (MTD) in this network. MTD moves a number of VMs from one physical system to another in order to increase the uncertainty of attackers. An important parameter of this MTD scenario is the optimum time in which VMs should be moved. While in [36], VMs are moved periodically, this pattern may cause significant vulnerabilities in networks, because best moving strategy is not necessarily periodic as mentioned in [26].

As shown in our system model in Figure 1, we assume the network controller obtains required information about the VMs or physical systems that are potentially under attack. This information can be provided by a smart and powerful detection system. As soon as the controller receives alarms from several systems and VMs, it must decide to move them in an untraceable manner. It means that vulnerable VMs should be located at physical systems that has free space. According to MTD, this is an active defense to overwhelm advance attacks, such as side channel attack. Although this defense can increase security and privacy of our network against wide variety of attacks, it is not cost-effective. In fact, the network controller tends to increase the network security as well as reduce the total cost of moves.

Network controller can improve the total cost by managing migrations. It can perform two types of migrations, i.e., *live* and *non-live* migrations. Live migration needs more resources and is also more costly than the non-live migrations. Network controller can decide which VMs should perform live

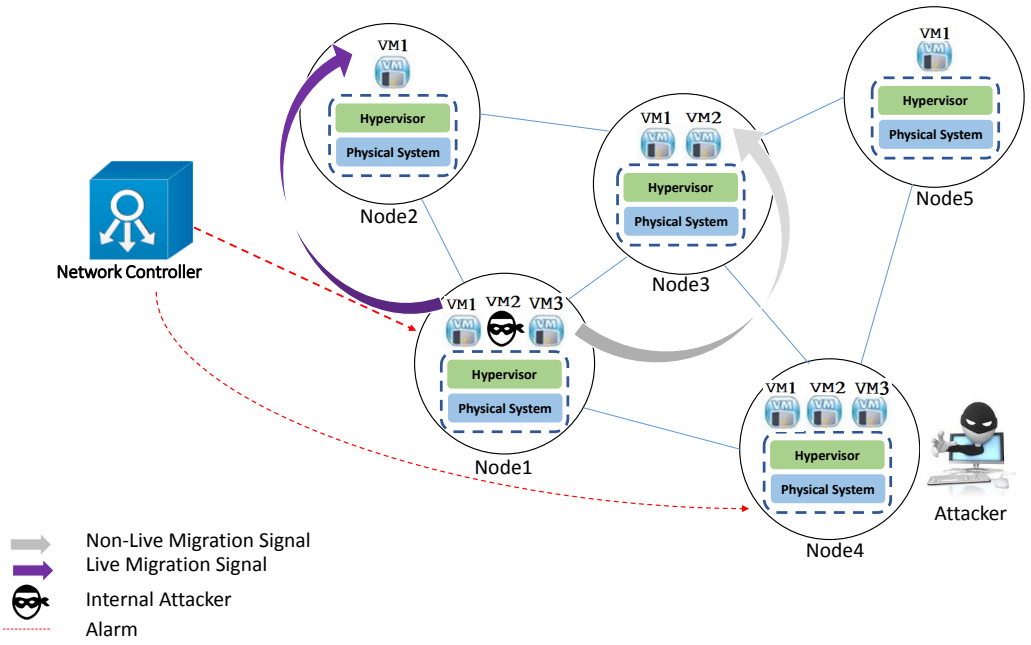

Fig. 1: System model includes network controller and VMs. Network controller receives alarm from nodes. It then performs MTD. In summary, it decides which VM must be moved and whether it moves live or non-live.

migration and which ones should carry out non-live migration in order to minimize the total cost of MTD. In this study, we propose an effective deceptive-based technique that can improve security of the network as well as reduce the cost of migration in such scenarios. In summary, after receiving the alert from the intrusion detection system, the network controller must decide which VMs must be moved. In the case where the VM must move, it should also decide the type of migration. Following, we propose an effective deception technique that can improve the security of the network as well as reduce the cost.

The idea of deception is as follows. For every received alarm, the controller decide to migrate the VM or not. In order to reduce the cost and improve the security of network, the controller may not migrate a particular VM but behaves such as it has been migrated. This means that when a VM does not migrate, still it is possible to make an attacker to deem that a live or non-live migration has been occurred. This can be done by producing different traffic behaviors in VMs. In this case, a non-migrated VM must make a deliberate disturbance in his current traffic or not. Hence, those attackers who sniff traffic of a VM think that it is going to migrate, either live or non-live. When attackers see traffic without disturbance, they think that the migration was live and when there is disturbance in traffic they think that the migration of VM was non-live.

In summary, when the network controller receives alarm for a given VM, it decides to migrate it with a given probability. It should also decide about the type of migration, i.e., *live* or *non-live*. Moreover, if the controller decides not to migrate the VM, it can also perform a deception scenario. The VM should either create deliberate disruption on his traffic or perform no action and continue to provide usual services. Consequently, when an attacker observes disruption or no disruption on traffic, it cannot detect that VM has been migrated or not. It can either trust to that signal or not. Under this uncertainty, the controller can improve security of the network by using deception while uses a limited number of migration to reduced the cost of MTD.

A. Game Model

Considering the proposed deception mechanism, we model the interaction between the attacker and the defender with a signaling game as shown in Figure 2. Signaling game is an incomplete information game that is defined as follows [37].

Definition 1. ***(Signaling game)*** *A signaling game is a two-player game in which Nature selects a game to be played according to a commonly known distribution, player 1 is informed of that choice and chooses an action, and player 2 then chooses an action without knowing Nature' s choice, but knowing player 1's choice.*

In our defined signaling game, the first player could be any of VMs in the network or defender which is designated by D. The attacker A is the second player that does not have complete information about the type of defender, i.e., whether D has decided to move or not. The intrusion detection system which provides alerts or the network controller could be the Nature in our signaling game. The Nature does not have a utility function (or, alternatively, can be viewed as having a constant one), and has the unique strategy of randomizing in a commonly known way to both the defender and the attacker.

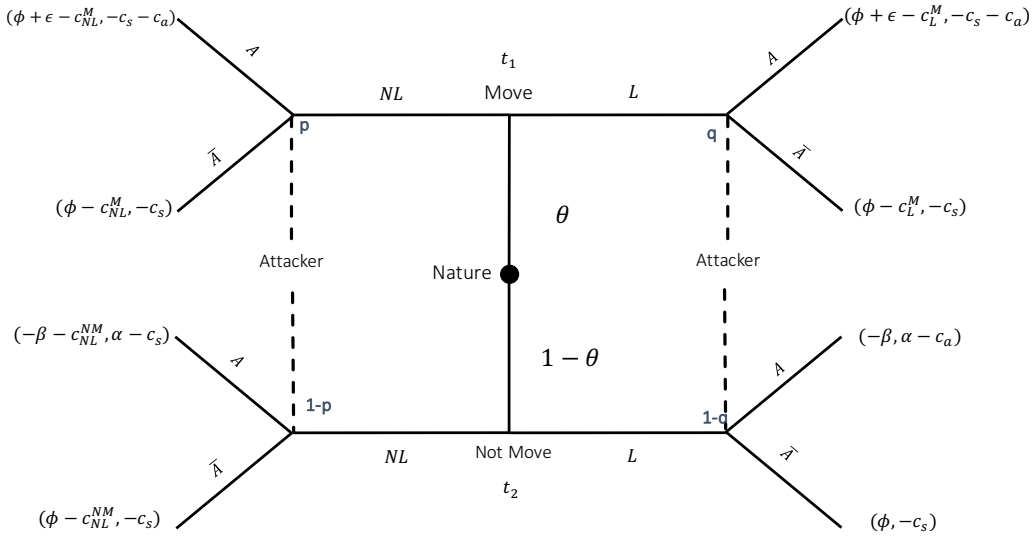

Fig. 2: Representation of the deception as a signaling game $\mathcal{G}_{\mathcal{M}}$. The players are the defender and the attacker. The belief of the attacker about the defender type (i.e., whether defender is moved or not) is modeled by p and q. The attacker observes action of the defender, i.e., L and NL. The attacker's actions are Attack (A) and Not-Attack ($\overline{A}$). The payoffs of both players are represented on the leaves of the tree.

In other words, players receive individual signals about the Nature's choice and both have common knowledge about that.

Let's call the defined MTD signaling game $\mathcal{G}_{\mathcal{M}}$, which is played in the following steps as shown in Figure 2:

1) The Nature (controller) draw type t_1 or t_2 with probability θ and $1-\theta$
   - If the Nature selects t_1, the VM should migrate.
   - If the Nature selects t_2, the VM should not migrate.
2) If the user received the order to migrate, he must decide whether to move live or non-live, but if the user received the order to not migrate, he must decide either to imitate live migration or non-live migration.
3) The attacker observes the user's migration, but does not know whether the defender is migrating or just emulate migration.

We define the user strategy profile by an ordered pair $(m, m^{'})$ in which m is type 1 strategy and $m^{'}$ is type 2 strategy. Similarly, the attacker's strategy is an ordered pair $(a, a^{'})$ in which a and $a^{'}$ demonstrate attacker's strategy following non-live and live, respectively. Table I summarizes the notations used throughout the paper.

B. Moving Target Defense

The game begins when the attacker attempts to compromise a VM on the defender's network. The intrusion detection system detects this threat. Given this information, the controller (i.e., the Nature in the game) chooses the type of each VM. In other words, it decides which VMs should migrate. The controller draw the type of each VM with probability of θ.

Now suppose that one of the VMs that was at risk received "migration" order from the controller. Since any migration could be performed in live or non-live form, the VM should decide whether to make a live or non-live migration. Note that each type of migration has different costs. We designate the cost of live and non-live migrations by c_l^m and c_{nl}^m, respectively. When a VM migrates, it could send an appropriate signal and the attacker observes the corresponding signal. This signal is visible either by the VM's traffic scanning or by listening to VM. The attacker can choose one of two strategies attack (A) or not-attack ($\overline{A}$) after the signal observation. c_s represents the cost of each attack that launches by attacker from the beginning.

When a VM migrates, the attacker has to perform all steps of attacks from the beginning. This is because the attacker's information about the VM is no longer valid. Therefore the attacker has to restart scanning, eavesdropping and other steps that were mentioned in [38], in order to make a successful attack. c_s represents these costs for the attacker. Hence, in all states related to migrated VMs, the attacker should pay this cost as it is shown in Figure 2. But if the attacker attacks to a migrated VM, the attack will not be successful. Consequently the attacker suffers from the cost equal to c_a. In two top states, the attacker should pay this cost because he attacks to a moved or migrated VM. In other two states he should just pay cost of restarting attack steps.

If a VM migrates just before an attack, the attack would fail. Suppose that the attacker decides not to attack (i.e., play $\bar{A}$) after that he observe a live migration signal. This results in security improvement for VM. Consequently, we consider a gain of ϕ for the VM and the cost c_l^m for the live migration. Therefore VM's payoff is $\phi - c_l^m$. If the attacker decides to attack (i.e., plays A) in response to the same signal, the VM

TABLE I: Table of Symbols

Symbol	Definition
θ	Probability of Nature to designate t_1
L	Action of defender to migrate Lively which also denoted by m_1 in signaling game setting
NL	Action of defender to migrate non-lively which also denoted by m_2 in signaling game setting
A	Reaction of the attacker to attack the targeted defender which also denoted by a_1 in signaling game setting
$\overline{A}$	Reaction of the attacker to not attack the targeted defender which also denoted by a_2 in signaling game setting
t_1	Type of defender in signaling game setting which is ordered to move
t_2	Type of defender in signaling game setting which is ordered to not move
p	The belief of the attacker about the defender's type when she received NL
q	The belief of the attacker about the defender's type when she received L
α	Attacker payoff when deception is not successful
c_s	Attacker cost when he has to restart all attack steps
c_a	Attacker cost when continues attacking to a migrated VM (in a wrong way)
c_{nl}^{m}	Cost of non-live migration for the defender when he is ordered to Move
c_{l}^{m}	Cost of live migration for the defender when he is ordered to Move
c_{nl}^{nm}	Cost of non-live migration for the defender when he is ordered not to Move
β	Cost of the defender when the attacker successfully attacked
ϕ	Benefit of the defender, when he successfully deceived the attacker
ϵ	Additional benefit for the defender when the attacker has attacked rather than not attacked the moved target

obtains ϕ because he migrated and is immune from the attack. Moreover, he gains more until the attacker can realize his mistake and restart a new attack. Since it takes a while and VM is secure during this time period, we assume that VM obtains more benefit shown by ϵ.

Note that in all four states mentioned above, the payoff of the VM is positive and it is larger than that of the attacker. In all these states the VM is secure until the next attack. This happened because of the benefits of moving target defense approach. In fact this approach is an active defensive technique performed before the attack. Therefore, this technique is one of the most attractive tools even against advanced persistent threats.

As mentioned in [39], migration cost reduction is one the most effective ways to overcome against attacker in moving target defense techniques. Although moving target defense is an attractive technique, moving costs may increase rapidly. As a simple example, let us suppose a powerful attacker who can explore and track all our migrations quickly. It is clear that high mobility against this attacker will result in high cost. This problem can be solved by reduction in migration's cost or by techniques such as deception. By applying our deception technique, one can maintain his system secure without too much migrations. In the following section, we address this deception technique in more details.

C. Deception Technique

Suppose the network controller determines that one VM should not move and sends this order to the VM. In this situation, the VM can produce signals like live or non-live migration in order to deceive the attacker. Live signal imitation means "no action" because there is no interrupt within traffic in a live migration. Non-live signal imitation means that the VM should make a deliberative disturbance, exactly like that in real non-live migration. With these imitations if the attacker receives a signal he cannot determine the real type of the defender. When a VM does not migrate and sends live signal to the attacker, if attacker plays attack strategy (i.e., A), his attack would be successful. Therefore he gains α and pays the cost of attack c_a. But the defender will suffer by cost β.

If after the reception of live migration, the attacker decides not to attack (i.e., plays $\overline{A}$), the attacker has to restart attack steps and pay the cost c_s. In this situation the defender earns ϕ because there is no attack until the next round. In fact the deception was successful in this case. Note that there is no cost for imitation of live migration since there is no need for any action. Clearly this state is very attractive for the defender because without any cost he could gain his system security by deception. As we will discuss in the next section, the cost of moves are very important for the defender and the calculated equilibrium of the game would be the function of this cost.

Let's consider the case where the not moved VM sends non-live signal. If the attacker in response decides to the attack (i.e., plays A), it obtains benefit equal to α and pay the cost of attack. In this condition, the defender loses his security and suffers by cost β, moreover the cost of non-live imitation (i.e., c_{nl}^{nm}) must be considered for the defender. If the attacker decides to not attack (i.e., play $\overline{A}$), he has to restart the attack steps until his uncertainty be resolved. With this reason he pays cost c_s. While the defender earns ϕ since there is no attack to his network and just pays the cost of non-live imitation, i.e., c_{nl}^{nm}. This state is also attractive for the defender because the deception was successful here too. This shows that there are different achievable conditions in which we can protect system's security similar to MTD but with less moves and consequently lower costs.

IV. Game Analysis and Protocol Design

In the following, we examine the signaling game for the existence and properties of any pure strategy Perfect Bayesian Nash Equilibria (PBE). We will then use our analysis to design a defensive protocol to optimize defender strategies. Fundamentally, in non-Bayesian games, a strategy profile is a Nash equilibrium (NE) if every strategy in that profile is a best response to every other strategy. But, in Bayesian games,

players are seeking to maximize their expected payoff, given their beliefs about the other players [40].

Gibbons [37] characterizes perfect Bayesian equilibrium as strategy profiles and beliefs that satisfy the following four requirements.

Requirement 1 : After observing any message m_j from M, the attacker must have belief about which types could have sent m_j. Denote this belief by the probability distribution $\mu(t_i|m_j)$, where $\mu(t_i|m_j) \geq 0$ for each t_i in T, and

$$\sum_{t_i \in T} \mu(t_i|m_j) = 1$$

Requirement 2 : For each m_j in M, the attacker's action $a^*(m_j)$ must maximize his expected utility, given the belief $\mu(t_i|m_j)$ about which types could have sent m_j. That is, $a^*(m_j)$ satisfies

$$\max_{a_k \in M} \sum_{t_i \in T} \mu(t_i|m_j) U_A(t_i, m_j, a_k)$$

Requirement 3 : For each $t_i \in T$, the defender's message $m^*(t_i)$ must maximize his utility, given the attacker's strategy $a^*(m_j)$. That is, $m^*(t_i)$ satisfies

$$\max_{m_j \in M} U_D(t_i, m_j, a^*(m_j))$$

Requirement 4 : For each $m_j \in M$, if there exists $t_i \in T$ such that $m^*(t_i) = m_j$, then the attacker's belief at the information set corresponding to m_j must follow from Bayes' rule and the defender's strategy:

$$\mu(t_i|m_j) = \frac{p(t_i)}{\sum_{t_i \in T_j} p(t_i)}$$

Where T_j denotes the set of types that send the message m_j.

Definition 2. *A pure-strategy **perfect Bayesian equilibrium** in a signaling game is a pair of strategy $m^*(t_i)$ and $a^*(m_j)$ and a belief $\mu(t_i|m_j)$ satisfying Requirement 1 to 4.*

In the following, we use $(p, 1-p)$ and $(q, 1-q)$ to denote the second player's beliefs at its two information sets. For the defined signaling game in Figure 2, the defender's pure strategy determined by an ordered pair $(m(t_1), m(t_2))$ which $m(t_1)$ is t_1 chosen strategy and $m(t_2)$ is t_2 chosen strategy. Similarly, attacker strategy is determined by an ordered pair $(a(NL), a(L))$, in which $a(NL)$ and $a(L)$ demonstrate attacker strategy following defender's non-live and live signal respectively.

Furthermore, a pure strategy PBE profile is determined as tuple $\{\mathcal{S}_D, \mathcal{S}_A, p, q\}$, in which $\mathcal{S}_D$ is the pair of sender strategy chosen by each type, $\mathcal{S}_A$ is the pair of attacker strategy in response to each signal, and p and q are attacker belief concerning the type of sender for non-live and live signal, respectively. According to the sender pure strategy, two kinds of PBE could exist in signaling game, called *pooling* and *separating*.

A PBE is called *pooling equilibrium* if $m(t_1) = m(t_2)$. In other words, defender sends the same signal, regardless of his type. In contrast, a PBE is called *separating equilibrium* if $m(t_1) \neq m(t_2)$, i.e., defender sends different signals, depending on his type. We now examine $\mathcal{G}_\mathcal{M}$ for (pure) PBE. We first probe the existence of separating equilibria.

Theorem 1. *There is no separating equilibrium in $\mathcal{G}_\mathcal{M}$ signaling game.*

Proof. Two different separating strategies would be possible: (L, NL) and (NL, L)

1) **Separating on** (L, NL): Suppose (L, NL) is a pair of defender's strategy, then both of the attacker's information sets are on the equilibrium path, so both beliefs are determined by Bayes' rule and sender strategy: $q = 1$ and $p = 0$. Attacker's best response following these beliefs are A and $\overline{A}$, respectively. It remains to check if the defender's strategy is optimal given the attacker strategy (L, NL). It is not, because if the defender of type 2 deviates by playing L signal instead of NL, attacker responds with $\overline{A}$, giving t_2 a payoff of ϕ, which is exceeds t_2's payoff of $-\beta - c_{nl}^m$ from playing NL.
2) **Separating on** (NL, L): Suppose (NL, L) is a pair of defender's strategy, then again according to the Bayes rule the attacker beliefs must be $q = 0$ and $p = 1$, hence the attacker's best response is $(\overline{A}, A)$ which means to select $\overline{A}$ and A following NL and L respectively. It remains to check whether the defender's strategy is optimal given the attacker strategy (NL, L). It is not, because if the defender of type 2 deviates by playing NL signal instead of L, attacker responds with $\overline{A}$, earning t_2 a payoff of $\phi - c_{NL}^{NM}$, which is exceeds t_2's payoff of $-\beta$ from playing L.

Consequently, there are no separating equilibrium in $\mathcal{G}_\mathcal{M}$. □

Theorem 1 explains that if the selected strategy by the different type of the defender is not the same, there is no equilibrium. In other words, for being in the equilibrium, the defender of type t_1 and type t_2 should play the same strategy. Since we do not have any separating equilibrium, a pure PBE in $\mathcal{G}_\mathcal{M}$ are either pooling on NL (i.e., non-live) or pooling on L (i.e., live). Thus, there are exactly two cases left to study: pooling on L and pooling on NL. In the following theorems, we investigate the existence of pooling equilibria.

Theorem 2. *For any values of θ, there exists a pooling on $Live$ equilibrium in $\mathcal{G}_\mathcal{M}$ signaling game only if $c_{nl}^m \geq c_l^m$.*

Proof. Suppose that the defender strategy is (L, L). Then the attacker's information set corresponding to L is on the equilibrium path, so the attacker's beliefs $(q, 1-q)$ at this information set is determined by Bayes' rule and defender strategy: $q = \theta$. Since $\frac{\alpha + c_s - c_a}{\alpha + c_s}$ is between 0 and 1, then for any $0 \leq \theta \leq 1$ we have two potential outcomes:

The attacker's expected payoff for playing A is:

$$\theta \times (-c_s - c_a) + (1 - \theta) \times (\alpha - c_a) \tag{1}$$

And attacker's expected payoff for playing $\overline{A}$ is:

$$\theta \times (-c_s) + (1 - \theta) \times (-c_s) \tag{2}$$

- $\theta \leq \theta^* := \frac{\alpha+c_s-c_a}{\alpha+c_s}$: Therefore playing A dominate $\overline{A}$ following L signal because,

$$\theta \times (-c_s - c_a) + (1-\theta) \times (\alpha - c_a) \geq \theta \times (-c_s)$$
$$+(1-\theta) \times (-c_s) \Longleftrightarrow q = \theta \leq \frac{\alpha + c_s - c_a}{\alpha + c_s} = f$$

To determine whether both defender types willing to choose L, we need to specify how the attacker would react to NL. If the attacker chooses strategy $\overline{A}$ for responding message NL, defender of type 2 can easily send NL instead of L and earn $\phi - c_{nl}^{nm}$ instead of $-\beta$.
But if the attacker chooses strategy A for responding message NL, there is no incentive for the defender to deviate from his strategy, because in that case defender of type 1 (t_1) achieves $\phi + \epsilon - c_{nl}^m$ instead of $\phi + \epsilon - c_l^m$ and defender of type 2 (t_2) obtains $-\beta - c_{NL}^{NM}$ instead of $-\beta$ which are less since $c_l^m \leq c_{nl}^m$.Thereby, there is no pooling on L equilibrium for $\theta \leq \theta^*$ if $c_l^m \geq c_{nl}^m$.
Otherwise, it remains to consider the attacker's belief at the information set corresponding to NL, and optimality of playing A given this belief. Attacker expected payoff for playing A and $\overline{A}$ following NL is $p \times (-c_s - c_a) + (1-p) \times (\alpha - c_a)$ and $p \times (-c_s) + (1-p) \times (-c_s)$ respectively. Accordingly, playing A is optimal following NL for $p \leq \theta^*$ because:

$$p \times (-c_s - c_a) + (1-p) \times (\alpha - c_a) \geq p \times (-c_s)$$
$$+(1-p) \times (-c_s) \Longleftrightarrow p \leq \frac{\alpha + c_s - c_a}{\alpha + c_s} = \theta^*$$

So $\{(L,L),(A,A),p,q = \theta\}$ for any $p \leq \theta^*$ is pooling equilibrium in $\mathcal{G}_{cs}$ only if $c_{nl}^m \geq c_l^m$.
- $\theta \geq \theta^*$: Attacker's best response following L's signal is $\overline{A}$ since,

$$\theta \times (-c_s - c_a) + (1-\theta) \times (\alpha - c_a) \leq \theta \times (-c_s)$$
$$+(1-\theta) \times (-c_s) \Longleftrightarrow q = \theta \geq \frac{\alpha + c_s - c_a}{\alpha + c_s} = \theta^*$$

To determine whether both defender types willing to choose L, we need to specify how the attacker would react to NL. If the attacker plays $\overline{A}$ following NL, then there is no incentive for defenders to deviate from their strategies only if $c_{nl}^m \geq c_l^m$. This happens, because they gain less by deviation (t_1 achieves $\phi - c_{nl}^m$ instead of $\phi - c_{nl}^m$ and t_2 attains $\phi - c_{NL}^{NM}$ instead of ϕ).
Note that if the attacker plays A following NL, then the defender of type 1 can increase its utility by sending NL instead of L if $c_{nl}^m \leq c_l^m$, i.e., $\phi - c_l^m$ to $\phi + \epsilon - c_{nl}^m$. Hence, there is no pooling equilibrium on L for $\theta \leq \theta^*$ if $c_l^m \geq c_{nl}^m$.
Otherwise, similar to the previous case, it remains to consider the attacker's belief at the information set corresponding to NL, and optimality of playing A given this belief. Given the attacker's expected payoff following NL, playing $\overline{A}$ is optimal for attacker following NL for $p \geq \theta^*$ because:

$$p \times (-c_s - c_a) + (1-p) \times (\alpha - c_a) \leq p \times (-c_s)$$
$$+(1-p) \times (-c_s) \Longleftrightarrow p \geq \frac{\alpha + c_s - c_a}{\alpha + c_s} = \theta^*$$

Hence $\{(L,L),(\overline{A},\overline{A}),p,q = \theta\}$ is pooling equilibrium for any $p \geq \theta^*$ in G_{cs} only if $c_l^m \geq c_{nl}^m$.

□

Theorem 2 represents that if the selected strategy of both types of the defender is L, there is an equilibrium. In this case, depending on the value of θ, the attacker should select one of the strategies A or $\overline{A}$ upon receiving signal L. In other words, if the probability of defender being ordered to move (i.e., θ) is greater than θ^*, the attacker should not attack. Otherwise, the best response for the attacker is playing A strategy. In case of $\theta \geq \theta^*$ and $\theta \leq \theta^*$, if the defender select strategy L, the attacker's response will be $\overline{A}$ and A respectively. In the following theorem, we investigate the existence of pooling on NL equilibrium.

Theorem 3. *There is no pooling on NL equilibrium in $\mathcal{G}_{\mathcal{M}}$ unless $c_l^m \geq c_{nl}^m + \epsilon$ and $\theta \geq \theta^*$.*

Proof. Suppose that defender strategy is pooling on live's signal. Then the attacker's information set corresponding to NL is on the equilibrium path, so the attacker's beliefs $(p, 1-p)$ at this information set is determined by Bayes' rule and defender strategy: $p = \theta$. The attacker's expected payoff for playing A and $\overline{A}$ following NL is obtained by Equation 1 and 2 respectively. There are two possible outcome for θ.
- $\theta \leq \theta^*$. Then the attacker best strategy is A since

$$\theta \times (-c_s - c_a) + (1-\theta) \times (\alpha - c_a) \geq \theta \times (-c_s)$$
$$+(1-\theta) \times (-c_s) \Longleftrightarrow p = \theta \leq \frac{\alpha + c_s - c_a}{\alpha + c_s} = \theta^*$$

If the attacher chooses A strategy following L's signal, then defender of type 2 (t_2) could achieve more by choosing to send L signal instead of NL and obtain $-\beta$ instead of $-\beta - c_{NL}^{NM}$. Moreover if the attacker choose $\overline{A}$ strategy toward L's signal, then also defender of type 2 (t_2) could obtain more by choosing to send L signal instead of NL and obtain ϕ instead of $-\beta - c_{NL}^{NM}$. Hence there is no pooling on NL equilibrium when $\theta \leq \theta^*$.
- $\theta \geq \theta^*$ Then the attacker best strategy is $\overline{A}$ since

$$\theta \times (-c_s - c_a) + (1-\theta) \times (\alpha - c_a) \leq \theta \times (-c_s)$$
$$+(1-\theta) \times (-c_s) \Longleftrightarrow p = \theta \geq \frac{\alpha + c_s - c_a}{\alpha + c_s} = \theta^*$$

If the attacher chooses $\overline{A}$ strategy following L's signal, then defender of type 2 (t_2) could achieve more by choosing to send L signal instead of NL and obtain ϕ instead of $\phi - c_{NL}^{NM}$. Moreover if $c_l^m \leq c_{nl}^m + \epsilon$ and the attacker choose A strategy toward L's signal, then also defender of type 1 (t_1) could obtain more by choosing to send L signal instead of NL and obtain $\phi + \epsilon - c_l^m$ instead of $\phi - c_{nl}^m$. Hence there is no pooling on NL equilibrium when $\theta \geq \theta^*$ and $c_l^m \leq c_{nl}^m + \epsilon$.
But if $c_l^m \geq c_{nl}^m + \epsilon$ and the attacker choose A strategy toward L's signal, then there is no incentive for the defender to deviate from his strategy, because in that case defender of type 1 (t_1) achieves $\phi + \epsilon - c_l^m$ instead of $\phi - c_{nl}^m$ and

TABLE II: Equilibria and their condition

$\mathcal{G}_{\mathcal{CD}}$	Range of θ	#	PBE	Condition on Beliefs	
				On-equilibrium	Off-equilibrium
$c_l^m \leq c_{nl}^m$	$\theta \leq \theta^*$	$\mathcal{PBE}_1$	$\{(L,L),(A,A),p,q\}$	$q=\theta$	$p \leq \theta^*$
	$\theta \geq \theta^*$	$\mathcal{PBE}_2$	$\{(L,L),(\overline{A},\overline{A},),p,q\}$	$q=\theta$	$p \geq \theta^*$
$c_{nl}^m \leq c_l^m \leq c_{nl}^m + \epsilon$	–	–	–	–	–
$c_{nl}^m + \epsilon \leq c_l^m$	$\theta \geq \theta^*$	$\mathcal{PBE}_3$	$\{(NL,NL),(\overline{A},A),p,q\}$	$p=\theta$	$q \leq \theta^*$

defender of type 2 (t_2) obtains $-\beta$ instead of $\phi - c_{nl}^{nm}$ which are less. its remains to consider the attacker's belief at the information set corresponding to L, and optimality of playing A given this belief. Attacker expected payoff for playing A and $\overline{A}$ following L is $p \times (-c_s - c_a) + (1-p) \times (\alpha - c_a)$ and $p \times (-c_s) + (1-p) \times (-c_s)$ respectively. Accordingly, playing A is optimal following L for $q \leq \theta^*$ because:

$$q \times (-c_s - c_a) + (1-q) \times (\alpha - c_a) \geq q \times (-c_s) + (1-q) \times (-c_s) \Longleftrightarrow q \leq \frac{\alpha + c_s - c_a}{\alpha + c_s} = \theta^*$$

So $\{(NL,NL),(\overline{A},A),q,p=\theta\}$ for any $q \leq \theta^*$ is pooling equilibrium in $\mathcal{G}_{cs}$ only if $c_l^m \geq c_{nl}^m + \epsilon$.

□

Theorem 3 states that if the value of θ is more than θ^* and $c_l^m \geq c_{nl}^m + \epsilon$, then there is an equilibrium in playing the strategy NL by defender. Otherwise, there is no pooling on NL equilibrium. In this case (i.e., $\theta \geq \theta^*$) and by observing the strategy NL, the attacker response is to not attack.

A. MTD Protocol with Deception

Table II summarizes our results about the existence of separating and pooling equilibria, investigated in Theorem 1, 2 and 3. To conclude, in the case that $c_l^m \leq c_{nl}^m$, the only PBE in $\mathcal{G}_{\mathcal{M}}$ is obtained by defender pooling on L strategies. These are denoted by $\mathcal{PBE}_1$ and $\mathcal{PBE}_2$ for $\theta \leq \theta^*$ and $\theta \geq \theta^*$, respectively. In summary, optimal decision of defender and attacker are obtained according to equilibrium $\mathcal{PBE}_1$ and $\mathcal{PBE}_2$ which means that the defender must move lively, whether he ordered to migrate or not.

In the case that $c_{nl}^m \leq c_l^m \leq c_{nl}^m + \epsilon$, there is no PBE in $\mathcal{G}_{\mathcal{M}}$. Thereby, the only logical approach for defender is to select his strategy randomly, i.e., each type of defender select his strategy by equal probability.

Finally, if $c_{nl}^m + \epsilon \leq c_l^m$, the only equilibrium obtained as θ is less than θ^* and denoted by $\mathcal{PBE}_3$. In this case, the defender's best strategy is to move non-lively, no matter what he ordered to do. In Algorithm 1, we summarize the defender strategy process considering our equilibrium analysis presented in Table II.

Algorithm 1 *Moving Target Defense*

procedure MOVING TARGET ALGORITHM(α, c_s, c_l^m, c_{nl}^m, ϵ, c_a, θ)
  if $c_l^m \leq c_{nl}^m$ **then**
    $\mathcal{S}_D \leftarrow (L,L)$
  if $c_{nl}^m \leq c_l^m \leq c_{nl}^m + \epsilon$ **then**
    $\mathcal{S}_D \leftarrow Random$ ▷ Defender chooses his strategy randomly
  if $c_{nl}^m + \epsilon \leq c_l^m$ **then**
    Compute $\theta^* := \frac{\alpha + c_s - c_a}{\alpha + c_s}$:
    if $\theta \geq \theta^*$ **then**
      $\mathcal{S}_D \leftarrow (NL,NL)$
    else
      $\mathcal{S}_D \leftarrow Random$ ▷ Defender chooses his strategy randomly

V. CONCLUSION

With the wide-spreading use of the cloud computing for sensitive data and critical operations, the secure management of cloud services is exceedingly becoming crucial. In this paper, we have proposed a moving target defense technique that deceptively apply the VM migration, in order to increase the cloud computing security as well as reducing the capability of an adversary to launch a successful attack. The technique also keeps its overhead limited by reducing the number of migrations. We have modeled the deception technique as a signaling game and solved it for the equilibrium strategies with respect to the defender (the VM) and the attacker. Finally, according to these results, we have devised an algorithm corresponding to this defense mechanism.

In future work, we can analyze the proposed mechanism with some real scenarios and experiments. It would also be interesting to evaluate the signaling game model with some numerical analysis. We can also address possible implementation of live and non-live migration defense with some case studies.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and feedback. We also thank Monireh Mohebbi Moghadam for her suggestions and feedbacks.

REFERENCES

[1] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and paradigms*. John Wiley & Sons, 2010.
[2] E. Amazon, "Amazon elastic compute cloud (amazon ec2)," *Amazon Elastic Compute Cloud (Amazon EC2)*, 2010.
[3] N. Sonehara, I. Echizen, and S. Wohlgemuth, "Isolation in cloud computing and privacy-enhancing technologies," *Business & information systems engineering*, vol. 3, no. 3, pp. 155–162, 2011.
[4] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104–113.

[5] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th conference on Computer and communications security*. ACM, 2009, pp. 199–212.

[6] Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter, "Homealone: Co-residency detection in the cloud via side-channel analysis," in *Symposium on Security and Privacy*. IEEE, 2011, pp. 313–328.

[7] K. Okamura and Y. Oyama, "Load-based covert channels between xen virtual machines," in *Proceedings of the Symposium on Applied Computing*. ACM, 2010, pp. 173–180.

[8] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, "An exploration of l2 cache covert channels in virtualized environments," in *Proceedings of the 3rd workshop on Cloud computing security workshop*. ACM, 2011, pp. 29–40.

[9] S. Yu, G. Xiaolin, L. Jiancai, Z. Xuejun, and W. Junfei, "Detecting vms co-residency in cloud: Using cache-based side channel attacks," *Elektronika ir Elektrotechnika*, vol. 19, no. 5, pp. 73–78, 2013.

[10] Z. Ling, J. Luo, Y. Zhang, M. Yang, X. Fu, and W. Yu, "A novel network delay based side-channel attack: modeling and defense," in *INFOCOM, Proceedings*. IEEE, 2012, pp. 2390–2398.

[11] Y. Zhang, M. Li, K. Bai, M. Yu, and W. Zang, "Incentive compatible moving target defense against vm-colocation attacks in clouds," in *IFIP International Information Security Conference*. Springer, 2012, pp. 388–399.

[12] Z. I. M. Yusoh and M. Tang, "A penalty-based genetic algorithm for the composite saas placement problem in the cloud," in *Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–8.

[13] J. S. Chase, D. E. Irwin, L. E. Grit, J. D. Moore, and S. E. Sprenkle, "Dynamic virtual clusters in a grid site manager," in *International Symposium on High Performance Distributed Computing*. IEEE, 2003, pp. 90–100.

[14] S. Butt, H. A. Lagar-Cavilla, A. Srivastava, and V. Ganapathy, "Self-service cloud computing," in *Proceedings of the conference on computer and communications security*. ACM, 2012, pp. 253–264.

[15] E. Keller, J. Szefer, J. Rexford, and R. B. Lee, "Nohype: virtualized cloud infrastructure without the virtualization," in *SIGARCH Computer Architecture News*, vol. 38, no. 3. ACM, 2010, pp. 350–361.

[16] B. Dolan-Gavitt, T. Leek, M. Zhivich, J. Giffin, and W. Lee, "Virtuoso: Narrowing the semantic gap in virtual machine introspection," in *Symposium on Security and Privacy*. IEEE, 2011, pp. 297–312.

[17] F. Zhang, J. Chen, H. Chen, and B. Zang, "Cloudvisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization," in *Proceedings of the Twenty-Third Symposium on Operating Systems Principles*. ACM, 2011, pp. 203–216.

[18] A. M. Nguyen, N. Schear, H. Jung, A. Godiyal, S. T. King, and H. D. Nguyen, "Mavmm: Lightweight and purpose built vmm for malware analysis," in *Annual Computer Security Applications Conference*. IEEE, 2009, pp. 441–450.

[19] J. M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and A. Perrig, "Trustvisor: Efficient tcb reduction and attestation," in *Symposium on Security and Privacy*. IEEE, 2010, pp. 143–158.

[20] M. Sindelar, R. K. Sitaraman, and P. Shenoy, "Sharing-aware algorithms for virtual machine colocation," in *Proceedings of the twenty-third annual symposium on parallelism in algorithms and architectures*. ACM, 2011, pp. 367–378.

[21] L. Grit, D. Irwin, A. Yumerefendi, and J. Chase, "Virtual machine hosting for networked clusters: Building the foundations for autonomic orchestration," in *Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing*. IEEE Computer Society, 2006.

[22] L. Ramakrishnan, D. Irwin, L. Grit, A. Yumerefendi, A. Iamnitchi, and J. Chase, "Toward a doctrine of containment: grid hosting with adaptive resource control," in *Proceedings of the conference on supercomputing*. ACM, 2006, p. 101.

[23] J. L. L. Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Dynamic placement of virtual machines for cost optimization in multi-cloud environments," in *International Conference on High Performance Computing and Simulation*. IEEE, 2011, pp. 1–7.

[24] W. Peng, F. Li, C.-T. Huang, and X. Zou, "A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces," in *International Conference on Communications (ICC)*. IEEE, 2014, pp. 804–809.

[25] Y. Han, T. Alpcan, J. Chan, and C. Leckie, "Security games for virtual machine allocation in cloud computing," in *International Conference on Decision and Game Theory for Security*. Springer, 2013, pp. 99–118.

[26] M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "Flipit: The game of stealthy takeover," *Journal of Cryptology*, vol. 26, no. 4, pp. 655–713, 2013.

[27] Y. Han, J. Chan, T. Alpcan, and C. Leckie, "Using virtual machine allocation policies to defend against co-resident attacks in cloud computing," 2015.

[28] J. B. Hong and D. S. Kim, "Scalable security models for assessing effectiveness of moving target defenses," in *44th Annual International Conference on Dependable Systems and Networks*. IEEE, 2014, pp. 515–526.

[29] M. A. Rahman, M. H. Manshaei, and E. Al-Shaer, "A game-theoretic approach for deceiving remote operating system fingerprinting," in *Conference on Communications and Network Security (CNS)*. IEEE, 2013, pp. 73–81.

[30] W. Casey, J. A. Morales, T. Nguyen, J. Spring, R. Weaver, E. Wright, L. Metcalf, and B. Mishra, "Cyber security via signaling games: Toward a science of cyber security," in *International Conference on Distributed Computing and Internet Technology*. Springer, 2014, pp. 34–42.

[31] M. M. Moghaddam, M. H. Manshaei, and Q. Zhu, "To trust or not: a security signaling game between service provider and client," in *International Conference on Decision and Game Theory for Security*. Springer, 2015, pp. 322–333.

[32] A. Mohammadi, M. H. Manshaei, M. M. Moghaddam, and Q. Zhu, "A game-theoretic analysis of deception over social networks using fake avatars," in *International Conference on Decision and Game Theory for Security*. Springer, 2016, pp. 382–394.

[33] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[34] K. Falzon and E. Bodden, "Dynamically provisioning isolation in hierarchical architectures," in *International Information Security Conference*. Springer, 2015, pp. 83–101.

[35] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th conference on Computer and communications security*. ACM, 2009, pp. 199–212.

[36] Y. Zhang, M. Li, K. Bai, M. Yu, and W. Zang, "Incentive compatible moving target defense against vm-colocation attacks in clouds," in *IFIP International Information Security Conference*. Springer, 2012, pp. 388–399.

[37] R. Gibbons, *A primer in game theory*. Harvester Wheatsheaf, 1992.

[38] E. Al-Shaer, "Toward network configuration randomization for moving target defense," in *Moving Target Defense*. Springer, 2011, pp. 153–159.

[39] K. D. Bowers, M. Van Dijk, R. Griffin, A. Juels, A. Oprea, R. L. Rivest, and N. Triandopoulos, "Defending against the unknown enemy: Applying flipit to system security," in *International Conference on Decision and Game Theory for Security*. Springer, 2012, pp. 248–263.

[40] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.