Foresighted Deception in Dynamic Security Games

Xiaofan He and Mohammad M. Islam
Dept. of EE, Lamar University
Emails: {xhe1,mislam11}@lamar.edu

Richeng Jin and Huaiyu Dai
Dept. of ECE, North Carolina State University
Emails: {rjin2,hdai}@ncsu.edu

***Abstract*—Deception has been widely considered in literature as an effective means of enhancing security protection when the defender holds some private information about the ongoing rivalry unknown to the attacker. However, most of the existing works on deception assume static environments and thus consider only myopic deception, while practical security games between the defender and the attacker may happen in dynamic scenarios. To better exploit the defender's private information in dynamic environments and improve security performance, a stochastic deception game (SDG) framework is developed in this work to enable the defender to conduct foresighted deception. To solve the proposed SDG, a new iterative algorithm that is provably convergent is developed. A corresponding learning algorithm is developed as well to facilitate the defender in conducting foresighted deception in *unknown* dynamic environments. Numerical results show that the proposed foresighted deception can offer a substantial performance improvement as compared to the conventional myopic deception.**

I. Introduction

Along the path of human civilization, there has never been a ceasefire in security. In view of the adversary's constant growing in intelligence and escalating in attacking tactics, game-theoretic approaches have been widely employed in literature to analyze the security competitions between the defender and the attacker [1–3].

Nevertheless, in many practical security competitions, the defender and the attacker possess different private knowledge about the ongoing rivalry, making the corresponding game-theoretic analysis non-trivial. When the defender holds extra private information unknown to the attacker, *deception* mechanisms are often employed to enhance security. In fact, deception has a long history of effective use in military (e.g., [4, 5]), anti-terrorism (e.g., [6, 7]), and is recently being exploited to protect information systems (e.g., [8–10]). Two common approaches have been adopted in literature to fulfill deception. The first approach adopts a signaling game model and allows the defender to design and proactively send out falsified signals to mislead the attacker about the ongoing security rivalry [7, 8, 11, 12]. In the second approach, the defender spends extra security resource to either hide its private information from the attacker's surveillance or create fake targets (e.g., honeypots [13]) to spoof the attacker [5, 9, 14, 15].

One limitation of existing works on deception is that most of them consider only static environments, while practical security problems often take place in dynamic environments (e.g., due to channel condition variations in a wireless environment, configuration changes in a computer network, or state evolvement in a cyber-physical system). In such cases, existing (myopic) deception mechanisms may not be able to achieve the best performance, as they fail to consider the influence of the current deception action on the future. To address this problem, foresighted deception that can better adapt to and fully exploit the environmental dynamics is considered in this work. In addition, a stochastic deception game (SDG) framework is developed to model the corresponding interactions between the foresighted deceptive defender and the attacker. To solve the proposed SDG and find a good foresighted deception strategy for the defender, a new iterative algorithm that is provably convergent is developed. In addition, considering that practical security problems may occur in unknown dynamic environments, a corresponding learning algorithm is developed as well to enable the defender to gradually learn the deception strategy.

The remainder of this paper is organized as follows. Section II starts with a motivating example of network protection game and then introduces the proposed SDG model. A new iterative algorithm is developed in Section III to solve the proposed SDG so as to find the defender's foresighted deception strategy. A learning algorithm is developed in Section IV to further enable the defender to conduct foresighted deception in unknown dynamic environments. Numerical results are presented in Section V to corroborate the effectiveness of foresighted deception. Conclusions and future works are discussed in Section VI.

II. Problem Formulation

A. The Network Protection Game

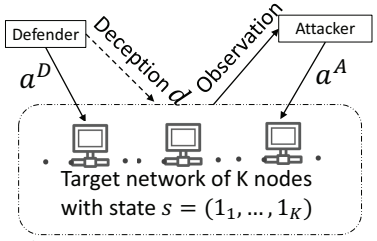

Fig. 1. The Network Protection Game.

To motivate the proposed work, consider a network protection game as depicted in Fig. 1. In this game, the defender (D) has to prevent the target network consisting of a set of K nodes $\mathbb{K} = \{1, ..., K\}$ from being brought down by the attacker (A) who strategically injects malwares into the network. Particularly, at each timeslot n, the defender (attacker) chooses a set of nodes $a_n^D \subseteq \mathbb{K}$ ($a_n^A \subseteq \mathbb{K}$) in the target network to enforce security protection (inject malwares) with a per node protection (attacking) cost φ_D

This work was supported in part by the NCSU Science of Security Lablet, and in part by National Science Foundation under Grants ECCS-1307949 and EARS-1444009.

(φ_A). It is also assumed that each of the K nodes in the target network can be in either a healthy (0) or an infected (1) state. In addition, it is assumed that when both the defender and the attacker act on the same node, this node will transit from infected (healthy) state to healthy (infected) state with probability p_{10} (p_{01}); when only the defender (attacker) acts on a node, this node will result in a healthy (infected) state; otherwise, the state of the node remains unchanged. Considering the possibility of malware spreading, it is further assumed that, after both the attacker and the defender taking their actions, an infection phase will occur, in which, any healthy node may be infected by an infected node with probability p_{inf}. In this security game, as the defender's objective is to maximize the number of healthy nodes at the minimum cost, its immediate reward at timeslot n can be modelled as

$$r_n^D = R^D(s_n, a_n^D, a_n^A) = \varphi\left(k_n\right) - \varphi_D \cdot |a_n^D| + \varphi_A \cdot |a_n^A|, \quad (1)$$

where R^D is the reward function of the defender; $|\cdot|$ denotes the cardinality of a set; the state of the target network $s_n \triangleq \{\mathbb{1}_{1,n}, ..., \mathbb{1}_{K,n}\}$ with $\mathbb{1}_{i,n}$ the indicator of whether or not node-i is infected; the non-decreasing function $\varphi(\cdot)$ gives the network's profit when $0 \leq k_n \triangleq K - \sum_{i=1}^{n} \mathbb{1}_{i,n} \leq K$ nodes remain healthy; and the last term is the attacker's attacking cost, which is accounted here due to the assumption of a zero-sum security game. The attacker's immediate reward is $r_n^A = -r_n^D$.

It can be seen from (1) that the optimal strategies of both the defender and the attacker depend on the current state s_n of the target network. If the defender can use deception techniques (e.g., deploying honeypots [13]) to disrupt the attacker's perception of the current state information, better defense performance may be obtained. Nonetheless, as deception usually comes with a certain cost, it is important for the defender to allocate the right amount of security resource for deception. This is a non-trivial task in dynamic environments, since the deception action at the current time will influence the attacker's current action and hence affect the future state of the target network as well as the future rewards of the defender. Conventional myopic deception originally designed for static security games may no longer be effective in such cases, and one may need to appeal to *foresighted deception* proposed in the next section.

B. The Proposed SDG Model

To achieve foresighted deception, a novel SDG framework is proposed by augmenting the classic *stochastic game* (c.f. [16] and references therein) with an extra deception procedure. As depicted in Fig. 2, the proposed SDG unfolds as follows: At the beginning of each timeslot n, the defender directly observes the state of the target system (or environment) $s_n \in \mathcal{S}$, with $\mathcal{S}$ the set of system states. Based on s_n, the defender takes a deception action $d_n(s_n)$ to conceal the true state according to a deception policy σ^D, at a cost $C^D(d_n)$. As a result, the attacker observes a false state $g(s_n) \in \mathcal{S} \cup \{\emptyset\}$ with probability $\mathbb{P}(d_n)$ and the true state s_n with probability $1 - \mathbb{P}(d_n)$.

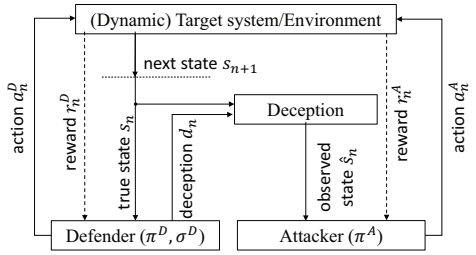

Fig. 2. Diagram of the proposed stochastic deception game.

The possibly stochastic function $g(\cdot)$ specifies the defender's selection of the false state in its deception mechanism. After this, the attacker will launch an attack a_n^A based on its observed state $\hat{s}_n$ (which equals either s_n or $g(s_n)$) and its attacking strategy π^A; the defender takes a defense action a_n^D based on the true state s_n and possibly $\hat{s}_n$ according to a policy π^D.[1] Immediate security rewards received by the defender and the attacker are $r_n^D = R^D(s_n, a_n^D, a_n^A)$ and $r_n^A = -r_n^D$, respectively, due to the zero-sum assumption. Then, the target system transits into a new state $s_{n+1} \in \mathcal{S}$ with a probability $\mathbb{P}(s_{n+1}|s_n, a_n^D, a_n^A)$ depending on the current state and actions from both the defender and the attacker. The objective of the defender is to find a strategy pair $\left(\sigma^D, \pi^D\right)$ to optimize its expected cumulative discounted reward $\mathbb{E}\left[\sum_{n=1}^{\infty} \beta^{n-1} \cdot \left(r_n^D - C^D(d_n)\right)\right]$ with discounting factor $\beta \in [0, 1)$. That is, the defender considers its long-term security performance in this dynamic environment, through evaluating the consequences of its actions on the future, but discounted for increasing uncertainty. As compared to existing deception games, the proposed SDG can further capture the dynamics in the target system/environment, and hence enables the defender to plan for the future. This procedure is coined by us as **foresighted deception**. It is worth mentioning that the proposed SDG reduces to 1) the conventional myopic deception when $\beta = 0$ and 2) the conventional stochastic game when the attacker directly observes the true state, respectively.

The proposed SDG is actually very general and may be applied to a wide range of security problems. In the following, the network protection game will be used as example to illustrate how the proposed SDG framework can guide the defender to conduct effective foresighted deception. Several assumptions that are reasonable to underlying application are clarified first. In this work, it is assumed that the defender always selects the null state as the false state (i.e., $g(s) = \emptyset$ for all $s \in \mathcal{S}$); more sophisticated state disguising mechanisms will be considered in our future work. It is also assumed that deception succeeds with probability $\mathbb{P}(d_n) = 1 - \exp(-\lambda d_n)$ (with $\lambda \geq 0$ the deception coefficient that represents the effectiveness of deception) when the defender spends d_n deception resource at a cost of $C^D(d_n) \triangleq d_n$. Note that the discussions in the following also hold for other forms of $\mathbb{P}(d_n)$. Moreover, it is assumed that, when the attacker observes $\hat{s} = \emptyset$, it will take an action a^A chosen uniformly at random.

[1]The deception result is assumed known to the defender in this work.

Based on the proposed SDG described above, the defender needs to solve the following optimization problem so as to find its optimal foresighted deception strategy $\{d(s^i)\}_{i=1}^{|\mathcal{S}|}$ for each of the possible state $s^i \in \mathcal{S}$.

$$\max_{\{d(s^i)\}_{i=1}^{|\mathcal{S}|}\geq 0} \quad \left(V^D(s^1), ..., V^D(s^{|\mathcal{S}|})\right) \qquad (\mathbf{P})$$

$$\begin{aligned}
\text{s.t.} \quad & (C1)\ V^D(s) = -d(s) + \exp(-\lambda d(s)) \cdot \tilde{V}^D(s,s) \\
& \quad + (1-\exp(-\lambda d(s))) \cdot \tilde{V}^D(s, g(s)),\ \forall s \in \mathcal{S} \\
& (C2)\ \tilde{Q}^D(s, a^D, a^A) = R^D(s, a^D, a^A) \\
& \quad + \beta \cdot \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s, a^D, a^A) \cdot V^D(s'), \\
& \qquad \forall s \in \mathcal{S}, \forall a^D \in \mathcal{D}, \forall a^A \in \mathcal{A}, \\
& (C3)\ \tilde{V}^D(s,s) = \mathrm{NE}\left(\tilde{Q}^D, s, s\right),\ \forall s \in \mathcal{S}, \\
& (C4)\ \tilde{V}^D(s, g(s)) = \mathrm{BR}\left(\tilde{Q}^D, s, g(s)\right),\ \forall s \in \mathcal{S}.
\end{aligned}$$

Some illustrations of the above optimization problem are in order. The *value function* $V^D(s)$ represents the defender's optimal long-term reward at a given current state s. In (C1), the *intermediate value function* $\tilde{V}^D(s, g(s))$ ($\tilde{V}^D(s,s)$) represents the defender's long-term reward when deception is successful (failed) *after* taking deception $d(s)$. (C2) is the well-known Bellman equation [17] and the *Q-function* $\tilde{Q}^D(s, a^D, a^A)$ represents the defender's long-term reward when players take actions (a^D, a^A) for the current timeslot and follow optimal strategies later on. (C3) and (C4) indicate that when deception fails, the defender should follow the Nash equilibrium (NE) at the current timeslot with a corresponding best possible long-term reward related to $\tilde{Q}^D$ and s; while if deception is successful, since the attacker will take an action chosen uniformly at random by assumption, the defender can take the corresponding best response (BR) [17] to seek a better reward. Also, a larger β indicates that the defender puts more emphasize on the future rewards.

Nonetheless, finding the optimal solution of this multi-objective optimization problem is challenging due to the inherent non-linear non-convex structure and equilibrium constraints.

III. Solving the Proposed SDG

In this section, instead of seeking the optimal solution, an algorithm that can always find a reasonably good solution to the optimization problem (**P**) with convergence assurance is proposed, enabling the defender to conduct effective foresighted deception in dynamic environment.

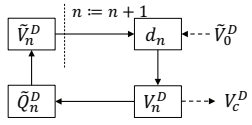

Fig. 3. Diagram of the Algorithm 1.

Particularly, by starting with some initial values, the proposed algorithm iteratively updates the deception strategy d, the value function V^D, the Q-function $\tilde{Q}^D$, and the intermediate value function $\tilde{V}^D$ through the order shown in Fig 3.

At the nth iteration of the proposed algorithm, based on the constraint (C1) of (**P**), for given $\tilde{V}_{n-1}^D(s,s)$ and $\tilde{V}_{n-1}^D(s, g(s))$, the optimal deception $d_n(s)$ and the corresponding value function $V_n^D(s)$ are updated by

$$d_n(s) = \max\left\{0, \frac{1}{\lambda}\log\left(\lambda \cdot \left(\tilde{V}_{n-1}^D(s, g(s)) - \tilde{V}_{n-1}^D(s,s)\right)\right)\right\} \quad (2)$$

and

$$\begin{aligned}
V_n^D(s) = & -d_n(s) + e^{-\lambda d_n(s)} \cdot \tilde{V}_{n-1}^D(s,s) \\
& + \left(1 - e^{-\lambda d_n(s)}\right) \cdot \tilde{V}_{n-1}^D(s, g(s)), \quad (3)
\end{aligned}$$

respectively. Note that the parameter of the log function in (2) is always well-defined due to the following fact.

Fact 1: The defender always assumes a higher security value when deception is successful. That is, $\tilde{V}^D(s, g(s)) \geq \tilde{V}^D(s,s)$, for all $s \in \mathcal{S}$.

Proof: This is straightforward and hence is omitted in the interest of space. ■

Then, the Q-function is updated by using

$$\tilde{Q}_n^D(s, a^D, a^A) = R^D(s, a^D, a^A) + \beta \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s, a^D, a^A) \cdot V_n^D(s'). \quad (4)$$

Finally, based on the updated Q-function, the updated intermediate value functions and the corresponding strategies are given by

$$\tilde{V}_n^D(s,s) = \mathrm{NE}(\tilde{Q}_n^D, s, s), \quad (5)$$

$$\pi_{\mathrm{NE,n}}^D(s, \cdot) = \arg \mathrm{NE}(\tilde{Q}_n^D, s, s), \quad (6)$$

$$\tilde{V}_n^D(s, g(s)) = \mathrm{BR}(\tilde{Q}_n^D, s, g(s)), \quad (7)$$

$$\pi_{\mathrm{BR,n}}^D(s, \cdot) = \arg \mathrm{BR}(\tilde{Q}_n^D, s, g(s)), \quad (8)$$

where $\pi_{\mathrm{NE,n}}^D$ ($\pi_{\mathrm{BR,n}}^D$) denotes the defender's strategy when deception fails (succeeds). These steps are summarized in Algorithm 1.

Algorithm 1 Computing Algorithm for the Defender

Initialization: set $\tilde{V}_0^D(s,s) = 0$, $\tilde{V}_0^D(s, g(s)) = 0$ and $V_n^D(s) = 0$ for all $s \in \mathcal{S}$; set $n = 1$.

Repeat

- (S1) Compute $d_n(s)$ using (2) based on $\tilde{V}_{n-1}^D(s, g(s))$ and $\tilde{V}_{n-1}^D(s,s)$, respectively, for all $s \in \mathcal{S}$.
- (S2) Compute $V_n^D(s)$ using (3) based on $d_n(s)$, $\tilde{V}_{n-1}^D(s, g(s))$ and $\tilde{V}_{n-1}^D(s,s)$, for all $s \in \mathcal{S}$.
- (S3) Compute $\tilde{Q}_n^D$ using (4) based on $V_n^D(s)$, for all $s \in \mathcal{S}$, $a^D \in \mathcal{D}$ and $a^A \in \mathcal{A}$.
- (S4) Compute $\tilde{V}_n^D(s,s)$ and $\tilde{V}_n^D(s, g(s))$ using (5) and (7), respectively, based on $\tilde{Q}_n^D$, for all $s \in \mathcal{S}$. Then, the updated $\pi_{\mathrm{NE,n}}^D$ and $\pi_{\mathrm{BR,n}}^D$ will be given by (6) and (8), respectively.
- $n := n+1$.

Until $|V_n^D(s) - V_{n-1}^D(s)| \leq \epsilon$.

Proposition 1: For all $s \in \mathcal{S}$, $\{V_n^D(s)\}_{n=0}^{\infty}$ in Algorithm 1 converges monotonically to a converging point $V_c^D(s)$, which is denoted by $V_n^D(s) \to V_c^D(s)$. Consequently, $\tilde{Q}_n^D(s, \cdot, \cdot) \to$

$\tilde{Q}_c^D(s,\cdot,\cdot)$, $\pi_{\mathrm{NE,n}}^D \to \pi_{\mathrm{NE,c}}^D$, $\pi_{\mathrm{BR,n}}^D \to \pi_{\mathrm{BR,c}}^D$, $\tilde{V}_n^D(s,s) \to \tilde{V}_c^D(s,s)$, $\tilde{V}_n^D(s,g(s)) \to \tilde{V}_c^D(s,g(s))$, and $d_n(s) \to d_c(s)$.

Proof: Please see Appendix A. ∎

As will be shown in Section V, by using the deception d_c and strategies $\pi_{\mathrm{NE,c}}^D$ and $\pi_{\mathrm{BR,c}}^D$ found in Algorithm 1, the defender can achieve effective foresighted deception that leads to substantial performance improvement as compared to conventional myopic deception.

IV. Learning in SDG

One limitation of Algorithm 1 developed in the previous section is the requirement of statistical knowledge about the target system dynamics (i.e., $\mathbb{P}(s'|s,a^D,a^A)$). In practice, such information may not be readily available to the defender. For example, in the network protection game considered in Section II-A, the infection rate p_{inf} may be unknown to the defender for new malwares. To conduct effective foresighted deception in such *unknown* dynamic environments, the learning counterpart of Algorithm 1 will be devised in this section.

This new learning algorithm is built by further embedding a reinforcement learning procedure [18] into Algorithm 1. Particularly, instead of directly computing the updated $\tilde{Q}^D$ through (4) as in Algorithm 1, the following reinforcement learning procedure will be used to allow the defender to gradually accumulate knowledge about $\tilde{Q}^D$ without knowing $\mathbb{P}(s'|s,a^D,a^A)$ beforehand.

$$\tilde{Q}_n^D(s,a^D,a^A) = \begin{cases} (1-\alpha_n)\tilde{Q}_{n-1}^D(s,a^D,a^A) + \alpha_n\Big(R^D(s,a^D,a^A) \\ \quad + \beta V_{n-1}^D(s_{n+1})\Big), & \text{for } (s,a^D,a^A) = (s_n,a_n^D,a_n^A) \\ \tilde{Q}_{n-1}^D(s,a^D,a^A), & \text{otherwise,} \end{cases} \tag{9}$$

where α_n is the so-called learning rate and satisfies $0 \le \alpha_n < 1$, $\sum_{n=0}^{\infty}\alpha_n = \infty$ and $\sum_{n=0}^{\infty}\alpha_n^2 < \infty$ [18].

The new learning algorithm is summarized in Algorithm 2 and its convergence is given in Proposition 2.

Proposition 2: All the quantities in Algorithm 2 converge to their corresponding converging points given by Algorithm 1.

Proof: Please see Appendix B. ∎

V. Numerical Results

The effectiveness of the proposed foresighted deception will be validated through simulations in this section. Particularly, consider the network protection game with the following setting: The network consists of $K = 2$ nodes, and the per node protection (attacking) costs are set to $\varphi_D = 1$ and $\varphi_A = 1$, respectively. In addition, the profit function of the network is assumed to be $\varphi(k_n) = \varphi_0 \cdot \mathbb{1}_{\{k_n>0\}} + \Delta\varphi \cdot k_n$, where $\varphi_0 = 10$ is the profit for keeping the network on (i.e., $k_n > 0$) and $\Delta\varphi = 1.5$ is the profit increasing rate with respect to the number of healthy nodes k_n. The infection probability p_{inf} is set to 0.5 and p_{01} and p_{10} are set to 0.3.

The convergence of Algorithm 1 (c.f. Proposition 1) is shown in Fig. 4. As it can be seen that, after about 20 iterations, the value functions V_n^D's for the four different states converge respectively.[2] It is not difficult to realize based on Algorithm 1 and constraints (C1)–(C4) that the convergence of V_n^D implies the convergence of $\tilde{V}_n^D$, $\tilde{Q}_n^D$, d_n, $\pi_{\mathrm{BR,n}}^D$ and $\pi_{\mathrm{NE,n}}^D$, which are not shown here due to space limitation.

Algorithm 2 Learning Algorithm for the Defender

Initialization: set $\tilde{Q}(s,\cdot,\cdot) = \mathbf{0}$, $\tilde{V}_0^D(s,s) = 0$, $\tilde{V}_0^D(s,g(s)) = 0$ and $V_n^D(s) = 0$ for all $s \in \mathcal{S}$; set $\pi_{\mathrm{NE},0}^D(s,\cdot)$ and $\pi_{\mathrm{BR},0}^D(s,\cdot)$ as the uniform random strategies and $n = 1$.

Repeat

- Observe the current state s_n.
- (M2) Update $d_n(s_n)$ and $V_n^D(s_n)$ using (2) and (3), respectively.
- Devote $d_n(s_n)$ amount of resource for deception.
- **if** deception suceeds **then**
  Taking action a_n^D at current state s_n
  - uniformly at random with an exploration probability p_{explr} [18];
  - otherwise, with probability $\pi_{\mathrm{BR,n-1}}^D(s_n,a_n^D)$;
- **else if** deception fails **then**
  Taking action a_n^D at current state s_n
  - uniformly at random with probability p_{explr};
  - otherwise, with probability $\pi_{\mathrm{NE,n-1}}^D(s_n,a_n^D)$;
- **end if**
- Learning: after receiving a reward $R^D(s_n,a_n^D,a_n^A)$ and observing the system state transition from s_n to s_{n+1}
  - (M3) Compute $\tilde{Q}_n$ using (9);
  - (M1) Compute $\tilde{V}_n^D(s,s)$ and $\tilde{V}_n^D(s,g(s))$ using (5) and (7), respectively, and $\pi_{\mathrm{NE,n}}^D$ and $\pi_{\mathrm{BR,n}}^D$ are updated by (6) and (8), respectively.
- $n := n+1$

The performance of the proposed foresighted deception is compared with that of the conventional myopic deception. In particular, the average security reward of using foresighted deception (with parameter β) $\bar{r}_n^{D,\beta} \triangleq \frac{1}{n}\sum_{i=1}^{n} r_i^D - C^D(d_i)$ is considered as the metric of interest here, and when $\beta = 0$ this coincides with the average security reward of myopic deception. The relative performance gain, defined as $\eta \triangleq \frac{\bar{r}_T^{D,\beta} - \bar{r}_T^{D,0}}{\bar{r}_T^{D,0}} \times 100\%$ over a period of $T = 5000$ timeslots, is shown in Fig. 5 for different β's and deception coefficient λ's. It can be seen that a substantial performance gain can be obtained by using the proposed foresighted deception. For example, when $\beta = 0.9$ and $\lambda = 1.5$, the proposed foresighted deception provides about 4 times extra reward as compared to myopic deception. Also, the performance gain becomes more significant when the defender puts more emphasis on future reward (corresponding to larger β) and when the deception coefficient λ is larger.

When statistical knowledge of the target system dynamics (i.e., p_{01}, p_{10} and p_{inf}) is unknown a priori, the defender can employ Algorithm 2 to gradually learn its foresighted deception strategy. The convergence of Algorithm 2 (c.f. Proposition 2) is shown in Fig. 6. It can be seen that V_n^D

[2] $V_n^D(s_2)$ and $V_n^D(s_3)$ overlap, since the two symmetric states $s_2 \triangleq \{\mathbb{1}_1 = 0, \mathbb{1}_2 = 1\}$ and $s_3 \triangleq \{\mathbb{1}_1 = 1, \mathbb{1}_2 = 0\}$ have equal values.

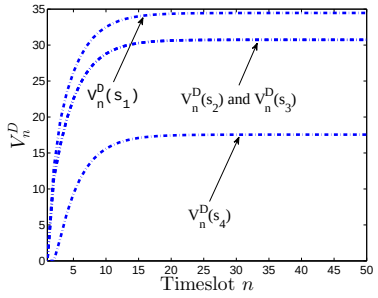

Fig. 4. Convergence of $\{V_n^D\}_{n=1}^{\infty}$ of Algorithm 1.

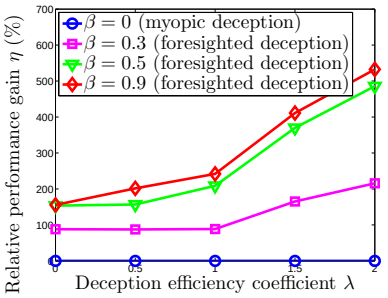

Fig. 5. Relative performance gain.

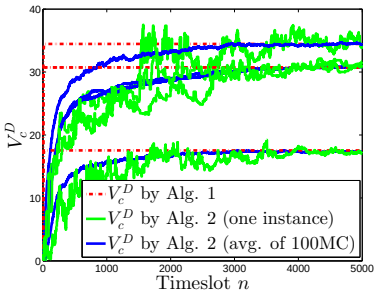

Fig. 6. Convergence of $\{V_n^D\}_{n=1}^{\infty}$ of Algorithm 2.

learnt through Algorithm 2 eventually converges to V_c^D given by Algorithm 1.

VI. Conclusions and Future Works

In this work, the novel concept of foresighted deception is proposed for enhancing the security performance in dynamic environments. In particular, an SDG framework is developed to guide the defender to conduct effective foresighted deception. To solve the proposed SDG, a new iterative algorithm that is provably convergent is developed. In addition, a corresponding learning algorithm is developed to enable the defender to gradually learn the foresighted deception strategy in unknown dynamic environments. Using the network protection game as an exemplary application, numerical results show that the proposed foresighted deception can lead to a substantial performance gain as compared to the conventional myopic deception. As to future work, it is interesting to consider the situations where the attacker may also spend some extra resource to conduct surveillance.

Appendix A
Proof of Proposition 1

Proof: The proof will consist of two parts where we will show that, for all $s \in \mathcal{S}$, $\{V_n^D(s)\}_{n=1}^{\infty}$ produced by Algorithm 1 is upper bounded and non-decreasing.

Define $M \triangleq \frac{1}{1-\beta}||R^D||_\infty \leq \infty$ (with $||\cdot||_\infty$ the max-norm). We will show that if $||V_n^D||_\infty \leq M$, then $||V_{n+1}^D||_\infty \leq M$. Particularly, we have,

$$||\tilde{Q}_n^D||_\infty \leq ||R^D||_\infty + \beta||V_n^D||_\infty \leq ||R^D||_\infty + \beta M = M, \quad (10)$$

where the first inequality is due to (S3) of Algorithm 1. In addition, it is not difficult to realize that $||\tilde{V}_n^D||_\infty \leq ||\tilde{Q}_n^D||_\infty$ based on (S4), and hence $||\tilde{V}_n^D||_\infty \leq M$. According to (S2), for all $s \in \mathcal{S}$, one has $V_{n+1}^D(s) \leq \tilde{V}_n^D(s, g(s))$. Then, we have $||V_{n+1}^D||_\infty \leq ||\tilde{V}^D||_\infty \leq M$. Therefore, given $||V_0^D||_\infty = 0$, it can be readily shown by induction that $\{V_n^D(s)\}_{n=1}^{\infty}$ is upper bounded by M.

The non-decreasing property of $\{V_n^D(s)\}_{n=1}^{\infty}$ is then shown in the following. First, notice that $d_1(s) = 0$ by (S1) and the given initial values of $\tilde{V}_0^D(s,s)$ and $\tilde{V}_0^D(s, g(s))$, and hence $V_1^D(s) = V_0^D(s) = 0$, for $s \in \mathcal{S}$. In the following, it will be shown that, given $V_n^D(s) \geq V_{n-1}^D(s)$, $V_{n+1}^D(s) \geq V_n^D(s)$ holds. To this end, notice that

$$\begin{aligned} V_{n+1}^D(s) &= -d_{n+1}(s) + e^{-\lambda d_{n+1}(s)} \cdot \tilde{V}_n^D(s,s) \\ &\quad + \left(1 - e^{-\lambda d_{n+1}(s)}\right) \cdot \tilde{V}_n^D(s, g(s)) \\ &\geq -d_n(s) + e^{-\lambda d_n(s)} \cdot \tilde{V}_n^D(s,s) \\ &\quad + \left(1 - e^{-\lambda d_n(s)}\right) \cdot \tilde{V}_n^D(s, g(s)) \\ &\geq -d_n(s) + e^{-\lambda d_n(s)} \cdot \tilde{V}_{n-1}^D(s,s) \\ &\quad + \left(1 - e^{-\lambda d_n(s)}\right) \cdot \tilde{V}_{n-1}^D(s, g(s)) = V_n^D(s), \end{aligned} \quad (11)$$

where the first and the last equalities are due to (S2) in Algorithm 1, and the first inequality is due to (S1); the last inequality is true if the following two conditions hold:

$$\tilde{V}_n^D(s,s) \geq \tilde{V}_{n-1}^D(s,s), \quad (12)$$

$$\tilde{V}_n^D(s, g(s)) \geq \tilde{V}_{n-1}^D(s, g(s)). \quad (13)$$

Before proving (12) and (13), notice that, due to (S3) in Algorithm 1 and the inductive assumption $V_n^D(s) \geq V_{n-1}^D(s)$, it can be verified that $\tilde{Q}_n^D(s,d,a) \geq \tilde{Q}_{n-1}^D(s,d,a)$ (and $\tilde{Q}_n^A(s,d,a) \leq \tilde{Q}_{n-1}^A(s,d,a)$ due to the zero-sum assumption) for all $(s,d,a) \in \mathcal{S} \times \mathcal{D} \times \mathcal{A}$. Since it is well-known that both the NE and the BR operators can be formulated as the following linear-programming problems:

$$\begin{aligned} \max_{[\pi_{NE}^D(s,\cdot)]^T \cdot \mathbf{1} = 1, \tilde{V}^D(s,s)} \quad & \tilde{V}^D(s,s) \\ \text{s.t.} \quad & [\pi_{NE}^D(s,\cdot)]^T [\tilde{Q}(s,\cdot,\cdot)] \geq \tilde{V}^D(s,s) \cdot \mathbf{1}^T \end{aligned}$$

and

$$\max_{[\pi_{BR}^D(s,\cdot)]^T \cdot \mathbf{1} = 1} \quad \tilde{V}^D(s, g(s)) = [\pi_{BR}^D(s,\cdot)]^T [\tilde{Q}(s,\cdot,\cdot) \cdot \mathbf{1}/|\mathcal{A}|],$$

from which, it can be seen that $\tilde{V}^D(s,s)$ and $\tilde{V}^D(s, g(s))$ are non-decreasing functions of $\tilde{Q}(s,\cdot,\cdot)$. Therefore, (12) and (13) hold.

Since for any $s \in \mathcal{S}$, $\{V_n^D(s)\}_{n=1}^{\infty}$ is an upper bounded and non-decreasing sequence, it converges [19] and the converging point is denoted by $V_c^D(s)$. Then, the convergence of other quantities readily follow. ■

APPENDIX B
PROOF OF PROPOSITION 2

Proof: The following lemma will be used to prove Proposition 2.

Lemma 1: (Szepesvari and Littman [20]) Assume a learning rate sequence α_n that satisfies $0 \leq \alpha_n < 1$, $\sum_{n=0}^{\infty} \alpha_n = \infty$ and $\sum_{n=0}^{\infty} \alpha_n^2 < \infty$, and a sequence of (random) mappings $\mathcal{T}_n$ from $\tilde{\mathcal{Q}}$ to $\tilde{\mathcal{Q}}$ (with $\tilde{\mathcal{Q}}$ denoting the set of quality functions $\tilde{Q}$'s) that satisfies: (**c1**) $\mathbb{E}[\mathcal{T}_n \tilde{Q}_c] = \tilde{Q}_c$; (**c2**) There exist a number $0 < \gamma < 1$ and a positive sequence φ_n converging to zero with probability 1, such that $||\mathcal{T}_n \tilde{Q} - \mathcal{T}_n \tilde{Q}_c||_\infty \leq \gamma \cdot ||\tilde{Q} - \tilde{Q}_c||_\infty + \varphi_n$, for all $\tilde{Q} \in \tilde{\mathcal{Q}}$. Then, the iteration defined by

$$\tilde{Q}_n(s, a^D, a^A) = \begin{cases} (1-\alpha_n)\tilde{Q}_{n-1}(s, a^D, a^D) + \alpha_n[\left(\mathcal{T}_n \tilde{Q}_{n-1}\right)(s, a^D, a^A)], \\ \qquad\qquad \text{if } (s, a^D, a^A) = (s_n, a_n^D, a_n^A), \\ \tilde{Q}_{n-1}(s, a^D, a^A), \qquad \text{otherwise}, \end{cases}$$

converges to Q_c with probability 1.

To prove Proposition 2, we will first define several mappings $\mathcal{M}_n^1$, $\mathcal{M}_n^2$ and $\mathcal{M}_n^3$ corresponding to (M1)–(M3) in Algorithm 2. Particular, for any $\tilde{Q} \in \tilde{\mathcal{Q}}$, $\mathcal{M}_n^1$ maps $\tilde{Q}$ to $\tilde{V}^D$ using (5) and (7) (with subscript n eliminated); $\mathcal{M}_n^2$ maps $\tilde{V}^D$ to V^D using (2) and (3); $\mathcal{M}_n^3$ randomly[3] maps V^D to a point in $\tilde{\mathcal{Q}}$ as follows: If $(s, a^D, a^A) = (s_n, a_n^D, a_n^A)$, then

$$\left(\mathcal{M}_n^3 V^D\right)(s, a^D, a^A) = R^D(s_n, a_n^D, a_n^A) + \beta V^D(s_{n+1}), \quad (14)$$

otherwise, $\left(\mathcal{M}_n^3 V^D\right)(s, a^D, a^A) = \tilde{Q}_{n-1}(s, a^D, a^A)$.

Define $\mathcal{T}_n$ as the composition of $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$. Note that $\mathcal{T}_n$ satisfies condition (c1) of Lemma 1, since, for $\tilde{Q}_c$ and V_c^D obtained in Algorithm 1,

$$\mathbb{E}\left[\mathcal{T}_n \tilde{Q}_c(s, a^D, a^A)\right] = \sum_{s'} \mathbb{P}(s'|s, a^D, a^A) \cdot \big(R^D(s, a^D, a^A) + \beta V_c^D(s')\big) = \tilde{Q}_c(s, a^D, a^A), \ \forall (s, a^D, a^A) \in \mathcal{S} \times \mathcal{D} \times \mathcal{A}, \quad (15)$$

where the last equality is due to the fact that the converging point $\tilde{Q}_c$ and V_c^D of Algorithm 1 must satisfy (C2) of the optimization problem (**P**). In the following, it will be shown that $\mathcal{T}_n$ also satisfies condition (c2). To this end, we show the following lemma first.

Lemma 2: For any $\tilde{Q} \in \tilde{\mathcal{Q}}$ and corresponding V^D,

$$|V^D(s) - V_c^D(s)| \leq ||\tilde{Q}(s) - \tilde{Q}_c(s)||_\infty, \quad (16)$$

where $\tilde{Q}(s) \triangleq [\tilde{Q}(s, a^D, a^A)]_{a^D \in \mathcal{D}, a^A \in \mathcal{A}}$ is the $|\mathcal{D}| \times |\mathcal{A}|$ Q-matrix for state s, with $|\mathcal{D}|$ and $|\mathcal{A}|$ the cardinalities of action sets of the defender and the attacker, respectively.

Proof: Based on (2) and (3), it is not difficult to realize that $V^D(s) \in \left[\tilde{V}^D(s, s), \tilde{V}^D(s, g(s))\right]$ and $V_c^D(s) \in \left[\tilde{V}_c^D(s, s), \tilde{V}_c^D(s, g(s))\right]$. Hence, we have

$$|V^D(s) - V_c^D(s)| \leq \max\big\{|\tilde{V}^D(s, g(s)) - \tilde{V}_c^D(s, s)|, \quad (17)$$

$$|\tilde{V}_c^D(s, g(s)) - \tilde{V}^D(s, s)|\big\}.$$

In addition, based on (5) and (7), it is clear that both $|\tilde{V}^D(s, g(s)) - \tilde{V}_c^D(s, s)|$ and $|\tilde{V}_c^D(s, g(s)) - \tilde{V}^D(s, s)|$ cannot exceed $||\tilde{Q}(s) - \tilde{Q}_c(s)||_\infty$. Therefore, (16) holds. ■

Then we can show (c2) holds (with $\varphi_n = 0$) as follows.

$$||\mathcal{T}_n \tilde{Q} - \mathcal{T}_n \tilde{Q}_c||_\infty = \beta \cdot \max_s |V_c^D(s) - V^D(s)| \leq \beta \cdot \max_s ||\tilde{Q}(s) - \tilde{Q}_c(s)||_\infty = \beta \cdot ||\tilde{Q} - \tilde{Q}_c||_\infty, \quad (18)$$

where the first equality is by (14) and the definition of $\mathcal{T}_n$, and the first inequality follows from Lemma 2.

Therefore, by Lemma 1 (with $\gamma = \beta$ and $\varphi_n = 0$), $\tilde{Q}_n$ in Algorithm 2 converges to $\tilde{Q}_c$, and hence V_n^D, d_n $\pi_{\text{NE,n}}^D$ and $\pi_{\text{BR,n}}^D$ converge to V_c^D, d_c $\pi_{\text{NE,c}}^D$ and $\pi_{\text{BR,c}}^D$, respectively. ■

[3] $\mathcal{M}_n^3$ is a random mapping since s_{n+1} is random given the information up to timeslot n; so is the mapping $\mathcal{T}_n$.

REFERENCES

[1] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. In *Proc. of IEEE HICSS*, Honolulu, HI, Jan. 2010.

[2] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J. P. Hubaux. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)*, 45(3):25, 2013.

[3] X. Liang and Y. Xiao. Game theory for network security. *IEEE Communications Surveys & Tutorials*, 15(1):472–486, 2013.

[4] J. P. Hespanha, Y. Ateskan, and H. Kizilocak. Deception in non-cooperative games with partial information. In *DARPA-JFACC Symposium on Advances in Enterprise Control*, 2000.

[5] D. Li and J. B. Cruz. Information, decision-making and deception in games. *Decision Support Systems*, 47(4):518–527, 2009.

[6] J. Zhuang and V. M. Bier. Secrecy and deception at equilibrium, with applications to anti-terrorism resource allocation. *Defence and Peace Economics*, 22(1):43–61, 2011.

[7] J. Zhuang, V. M. Bier, and O. Alagoz. Modeling secrecy and deception in a multiple-period attacker–defender signaling game. *European Journal of Operational Research*, 203(2):409–418, 2010.

[8] T. E. Carroll and D. Grosu. A game theoretic investigation of deception in network security. *Security and Communication Networks*, 4(10):1162–1172, 2011.

[9] F. Cohen and D. Koike. Misleading attackers with deception. In *Proc. of IEEE SMC information assurance workshop*, Jun. 2004.

[10] H. Xu, A. X. Jiang, A. Sinha, Z. Rabinovich, S. Dughmi, and M. Tambe. Security games with information leakage: Modeling and computation. *arXiv preprint arXiv:1504.06058*, 2015.

[11] J. Pawlick and Q. Zhu. Deception by design: Evidence-based signaling games for network defense. *arXiv preprint arXiv:1503.05458*, 2015.

[12] H. Xu, Z. Rabinovich, S. Dughmi, and M. Tambe. Exploring information asymmetry in two-stage security games. In *Proc. of AAAI*, 2015.

[13] N. C. Rowe, E. J. Custy, and B. T. Duong. Defending cyberspace with fake honeypots. *Journal of Computers*, 2(2):25–36, 2007.

[14] Y. Yin, B. An, Y. Vorobeychik, and J. Zhuang. Optimal deceptive strategies in security games: A preliminary study. In *Proc. of AAAI*, 2013.

[15] J. Xu and J. Zhuang. Modeling costly learning and counter-learning in a defender-attacker game with private defender information. *Annals of Operations Research*, 236(1):271–289, 2016.

[16] L. Busoniu, R. Babuska, and B. De Schutter. A comprehensive survey of multiagent reinforcement learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):156–172, 2008.

[17] T. Alpcan and T. Başar. *Network security: A decision and game-theoretic approach*. Cambridge University Press, 2010.

[18] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. of ICML*, 1994.

[19] J. Yeh. *Real analysis: Theory of measure and integration*. World Scientific, 2006.

[20] C. Szepesvári and M. Littman. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural computation*, 11(8):2017–2060, 1999.