Chapter 2
Proactive Network Defense with Game Theory

Sinong Wang and Ness Shroff

Abstract Traditional proactive network defenses deploy security resources in the network based on probabilistic policies to confuse potential attackers. However, this strategy can be exploited by stealthy attackers, leading to reduced efficiency and higher vulnerability. Game theory has been shown to provide a sound mathematical approach to overcome these deficiencies and determine an optimal defense strategy. However, existing game theoretic models typically either assume additive utility functions, or that the attacker can attack only one target. While such assumptions lead to tractable analyses, they miss key inherent dependencies that exist among different targets in current complex networks. In this chapter, we generalize the traditional security game model to the network scenario. We examine such a general security game from a theoretical perspective and provide a unified theoretical framework. In particular, we show that each security game is equivalent to a combinatorial optimization problem over a set system, which consists of defender's pure strategy space. The key technique we use is based on projection of a polytope based transformation, and the ellipsoid method. We also provide several important applications of our developed framework, and show that for several problem classes, optimal defense strategies can be developed in polynomial time. Our approach paves the way for a deeper investigation into using game theoretic techniques for solving designing security mechanisms in networks, and we conclude by outlining a number of important future directions that need to be investigated.

2.1 Introduction

Most critical systems use some type of proactive defense through firewalls, reinforcing systems through regular software updates, providing police protection of important locations, etc. However, one of the key problems in proactive network

S. Wang · N. Shroff (✉)
The Ohio State University, Columbus, OH, USA
e-mail: wang.7691@osu.edu; shroff.11@osu.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_2

defense is *how to efficiently allocate limited resources to protect targets in a network against potential threats*. For example, the government may have a limited police force to operate checkpoints and conduct random patrols over some city blocks, or have a limited number of coders that restricts how often and for what functionality new software updates are generated. However, the adversarial aspect in security domain poses a unique challenge for allocating resources. An intelligent attacker can observe the defender's strategy and gather information to schedule more effective attacks. Therefore, the simple random strategy of "rolling the dice" may be exploited by the attacker, which greatly reduces the effectiveness of the strategy. This is where game theory can help devise strategies that are optimal even under intelligent and stealthy attackers.

2.1.1 Why Game Theory?

Before we describe the importance of applying game theory to the proactive network defense, let us first look at the following example.

Example 2.1 As shown in Fig. 2.1, there exists a network with multiple nodes and links. The goal of defender or infrastructure service provider is to transmit the packets from the node s to node d along different paths. In practice, there might exists some hackers attempting to intercept the packet and subtract the confidential contents. To avoid interception from attackers, the defender can probabilistically choose a different routing path. For example, in the above network, we have four routing paths that has the possibility to confuse the attacker. However, the question is *is probabilistically mixing the strategy a secured policy in proactive network defense?* In practice, the stealthy attacker can observe the defender's probabilistic strategy and predict the defender's next move, which may lead to disastrous consequences.

With the development of computational game theory, such resource allocation problems can be cast in game-theoretic contexts, which provides a sounder math-

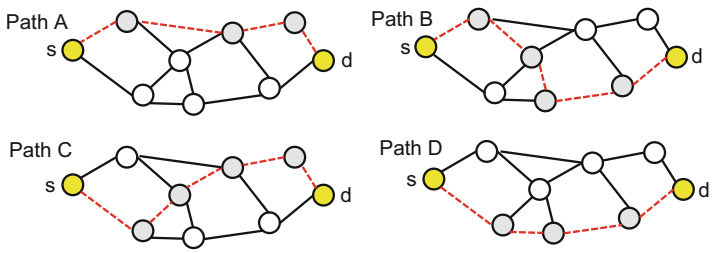


Fig. 2.1 A network with four possible routing paths. The yellow nodes are source and destination nodes. The grey nodes are intermediate nodes in the routing path

ematical approach to determine the optimal defense strategy. It allows the analyst to factor differential risks and values into the model, incorporate game-theoretic predictions of how the attacker would respond to the security policy, and finally determine an equilibrium strategy that cannot be exploited by adversaries to obtain a higher payoff. In the past decade, there has been an explosion of research attempting to address this approach, which has led to the development of well-known models of security games.

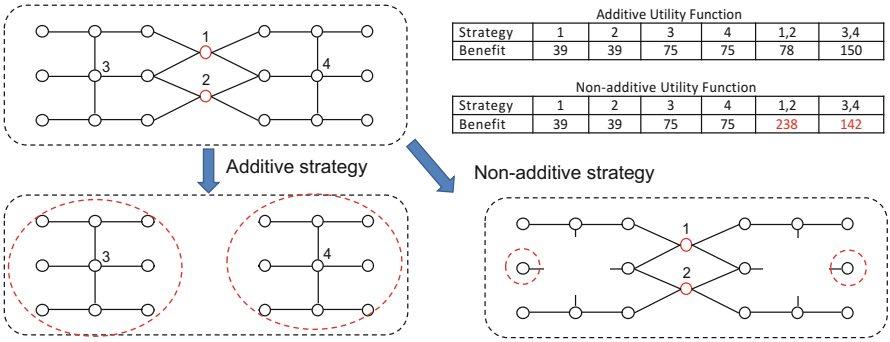
Moreover, it has become increasingly apparent that security failures in network and information systems are often caused by a misunderstanding of the incentives of the entities involved in the system instead of a lack of proper technical mechanisms [1, 2]. To this end, there exists game theoretical models trying to understanding this phenomenon using analytical approaches [3–6]. Some other recent works [7–9] also consider Advanced Persistent Threats (APT) in cyber security. APT attacks have several distinguishing properties that render traditional defense mechanism less effective. First, they are often launched by incentive driven entities with specific targets. Second, they are persistent in achieving the goals, and may involve multiple stages or continuous operations over a long period of time. Third, they are highly adaptive and stealthy, which requires the game model capturing the persistent and stealthy behavior of advanced attacks.

The classic *security game* is a two-player game played between a *defender* and an *attacker*. The attacker chooses one target to attack; The defender allocates (randomly) limited resources, subject to various domain constraints, to protect a set of targets. The attacker (defender) will obtain the benefits (losses) for those successfully attacked targets and losses (benefits) for those defended targets. The goal of the defender is to choose a random strategy so as to play optimally under some solution concepts such as Nash equilibrium and strong Stackelberg equilibrium. This *security game* model and its game-theoretic solution is *currently being used by many security agencies including US Coast Guard and Federal Air Marshals Service*(FAMS) [10], Transportation System Administration [11] and even in the wildlife protection [12]; see book by Tambe [13] for an overview.

2.1.2 Challenges in the Classical Security Game Model

Before we discuss the challenges in the classical security game model, let us first consider the following example.

Example 2.2 As shown in Fig. 2.2, we have a 20-node network. It is clear that nodes 1, 2, 3 and 4 are the critical battlefields in this network. Suppose that the attacker's and defender's strategies are {1}, {2}, {3}, {1, 2} or {3, 4}, where $\{v\}$ denotes the index of the nodes. We adopt the network value proposed by Gueye et al. [14] as the security measure for different nodes, which calculates the importance of a group of nodes by subtracting the value of the network by removing these nodes from



Additive Utility Function

Strategy	1	2	3	4	1,2	3,4
Benefit	39	39	75	75	78	150

Non-additive Utility Function

Strategy	1	2	3	4	1,2	3,4
Benefit	39	39	75	75	238	142

Fig. 2.2 Example of security game in a 20-nodes network with independent targets assumption (additive) or dependent target assumption (non-additive)

the value of the original network.[1] For example, if we adopt the network value as a function $f(\{n_i\}) = \sum_i n_i^2$, where n_i is number of nodes in the ith component, the value of the original network is $20^2 = 400$. After removing node 3, the network will be divided into two components: one 18-node network and one isolated node, the network value is reduced to $18^2 + 1^2 = 325$. Thus the benefit of node 3 is equal to the decrement $400 - 325 = 75$. Similarly, we can get the benefits of other nodes as illustrated in the bottom table of Fig. 2.2. In traditional security game models, they assume that the benefit of strategy {1, 2} and {3, 4} is equal to $39 + 39 = 78$ and $75 + 75 = 150$. The mixed strategy equilibrium[2] under this case is that defender choose nodes 1, 2 with probability 0.34 and nodes 3, 4 with probability 0.66. Instead, if we adopt the true value of nodes {1, 2} and {3, 4} (as illustrated in red of bottom table), the equilibria is that the defender chooses nodes 1, 2 with probability 0.63 and nodes 3, 4 with probability 0.37. From the point view of the network, the second one provides a more reliable strategy.

Based on the above example, we have the following observations: first, the traditional security game models do not consider dependency among the different targets; second, the attacker can attack at most one target. In particular, the payoff functions for both players are additive, i.e., the payoff of a group of targets is the sum of the payoffs of each target separately. This assumption means that the security agency measures the importance of several targets without considering the synergy among them. In practice, the attacker can simultaneously attack multiple targets and there exists some linkage structure among those targets such that attacking one target will influence the other targets. For example, an attacker attempts to destroy the

[1] Compared with traditional measures such as degree and betweenness centrality, the network value provides a more accurate description of the importance of different nodes.

[2] In this example, we adopt the zero-sum game model and assume the defender can protect the nodes with probability 1.

connectivity of a network and the defender aims to protect it. The strategy for each players is to choose the nodes of the network (to defend or to attack). If there are two nodes (node 1 and 2 in previous example) that constitute a bridge of this network, successfully attacking both of them will split the network into two parts and incur a huge damage, while attacking any one of them will have no significant impact. These observations show that proactive network defense introduces new challenges in computational game theory, and calls for the new theoretical development. The rest of this chapter mainly focus on how to develop a general game-theoretical path and algorithmic framework in proactive network defense.

2.2 Non-additive Security Game: A General Formulation of Network Security Game

Motivated by the previous example, we are now ready to define the non-additive security game (NASG) [15, 16].

Players and Targets The NASG contains two players (a *defender* and an *attacker*), and n targets. We use $[n] \triangleq \{1, 2, \ldots, n\}$ to denote the set of these targets. The attacker and defender need not be individuals, but could also be the organizations and groups who adopt a joint strategy. The target can be quite general and dependent on the application in mind. For example, they could represent links in the communication networks, roads in the urban networks or cities in the whole country.

Strategies and Index Function The *pure* strategy for each player is the subset of targets and all the pure strategies for each player constitute a collection of subsets of $[n]$. We assume that the attacker can attack at most c targets, where $c > 1$ is a constant. The attacker's pure strategy space is a uniform matroid $\mathscr{A} = \{A \subseteq [n] || A| \leq c\}$ and the number of attacker's pure strategies is $N_a \triangleq |\mathscr{A}|$. Similarly, we use $\mathscr{D} \in 2^{[n]}$ to denote the defender's pure strategy space and $N_d \triangleq |\mathscr{D}|$. Note that there exists some resource allocation constraints in practice and such that $\mathscr{D}$ is not always a uniform matroid. For example, if the defender has a budget and its resource are obtained at some costs, in which the costs are heterogeneous. In this case, the defender's feasible pure strategy corresponds to all the possible combinations of the targets with total cost less than the budget.

Suppose that the order of the pure strategy of the attacker is given by index function $\sigma(\cdot)$, which is a one-one mapping: $2^{[n]} \to \{1, 2, \cdots, 2^n\}$. Then, we define the following index function $\mu(\cdot)$ for the pure strategy of the defender as: $\mu(U) = \sigma(U^c)$ for any $U \in 2^{[n]}$. For simplicity, the index function $\sigma(\cdot)$ and $\mu(\cdot)$ are defined over all subsets of $[n]$. The reason behind this definition of the index function is to simplify the representation of most of the theoretical results. For example, if $n = 2$, $\mathscr{A} = \mathscr{D} = 2^{\{1,2\}}$, and the order of the attacker's pure strategy is $\sigma(\{1, 2\}) = 1$, $\sigma(\{2\}) = 2$, $\sigma(\{1\}) = 3$ and $\sigma(\emptyset) = 4$, then the order for defender's pure strategy is $\mu(\emptyset) = 1$, $\mu(\{1\}) = 2$, $\mu(\{2\}) = 3$ and $\mu(\{1, 2\}) = 4$.

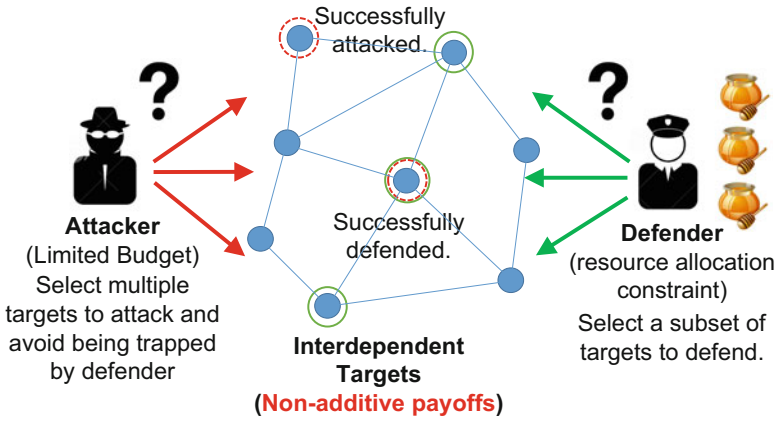


Fig. 2.3 Network security game with non-additive utility functions and multiple attacker resources

The *mixed* strategy is the probability distribution over the pure strategy space, which is employed when the player determines its strategy based on some random experiment. For example, if the attacker chooses $\mathbf{p}$ as its mixed strategy, the probability that strategy A is chosen is $\mathbf{p}_{\sigma(A)}$. The set of all the mixed strategies of the attacker and defender can be represented as the simplex Δ_{N_a} and Δ_{N_d}, where

$$\Delta_{N_a} = \{\mathbf{p} \in \mathbb{R}_+^{N_a} | \sum_{A \in \mathscr{A}} \mathbf{p}_{\sigma(A)} = 1\}. \tag{2.1}$$

A similar definition holds for Δ_{N_d}.

Payoff Structure The benefits and losses are represented by utility functions as follows. Let set function $B(\cdot) : \mathscr{A} \to \mathbb{R}$ and $L(\cdot) : \mathscr{A} \to \mathbb{R}$ be the attacker's benefit and loss functions, respectively. The standard assumption is that the benefit is always larger than the loss: $B(A) > L(A)$ for all $A \in \mathscr{A}$. If the attacker and defender choose strategy $A \in \mathscr{A}$ and $D \in \mathscr{D}$, the attacker's and defender's payoff is given by $B(A \backslash D) + L(A \cap D)$ and $-L(A \cap D) - B(A \backslash D)$, respectively.[3] In this payoff structure, one can see that the game is zero-sum such that one player's benefit is indeed the loss of the other players. For more complex non-zero sum games, please refer to [16].

Bilinear-Form Based on the above payoff structure, we can define the benefit matrices of attacker $\mathbf{B} : \forall A \in \mathscr{A}, D \in \mathscr{D}$,

$$\mathbf{B}_{\sigma(A),\mu(D)} = B_a(A \backslash D), \tag{2.2}$$

[3] $A \backslash D$ is the standard set difference, defined by $A \backslash D = \{x | x \in A, x \notin D\}$ and is equal to $A \cap D^c$, where D^c is the complementary set of subset D. An example of NASG is illustrated in (Fig. 2.3).

and the loss matrices: $\mathbf{L}$: $\forall A \in \mathscr{A}$, $D \in \mathscr{D}$,

$$\mathbf{L}_{\sigma(A),\mu(D)} = L_a(A \cap D). \tag{2.3}$$

Let $\mathbf{M}^a$ and $\mathbf{M}^d$ be the attacker's and defender's payoff matrices. It is clear that $\mathbf{M}^a = \mathbf{B} + \mathbf{L}$ and $\mathbf{M}^d = -\mathbf{B} - \mathbf{L}$. Then the expected payoffs for the attacker and defender are given by following bilinear form, when they play the mixed strategy $\mathbf{p} \in \Delta_{N_a}$ and $\mathbf{q} \in \Delta_{N_d}$, by

$$U_a(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T \mathbf{M}^a \mathbf{q} \quad \text{and} \quad U_d(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T \mathbf{M}^d \mathbf{q}. \tag{2.4}$$

Solution Concepts If both players move simultaneously, the standard solution concept is the *Nash equilibrium (NE)*, in which no single player can obtain a higher payoff by deviating unilaterally from this strategy. A pair of mixed strategies ($\mathbf{p}^*$, $\mathbf{q}^*$) forms a *NE* if and only if they satisfy the following: $\forall \mathbf{p} \in \Delta_{N_a}, \mathbf{q} \in \Delta_{N_d}$,

$$U_d(\mathbf{p}^*, \mathbf{q}^*) \geq U_d(\mathbf{p}^*, \mathbf{q}) \text{ and } U_a(\mathbf{p}^*, \mathbf{q}^*) \geq U_a(\mathbf{p}, \mathbf{q}^*). \tag{2.5}$$

In some application domain, the defender can build fortifications before the attack and is thus in the leader's position from the point view of the game, and able to move first. In this case, the *strong Stackelberg equilibrium (SSE)* serves as a more appropriate solution concept [17, 18], where the defender commits to a mixed strategy; the attacker observes this strategy and comes up with its best response(s). Formally, let $C(\mathbf{q}) = \arg\max_{\mathbf{p} \in \Delta_{N_a}} U^a(\mathbf{p}, \mathbf{q})$ denote the attacker's best response to defender's mixed strategy $\mathbf{q}$. A pair of mixed strategies ($\mathbf{p}^*$, $\mathbf{q}^*$) is a SSE, if and only if,

$$\mathbf{q}^* = \arg \max_{\mathbf{q} \in \Delta_{N_d}} U_d(C(\mathbf{q}), \mathbf{q}) \text{ and } \mathbf{p}^* = C(\mathbf{q}^*). \tag{2.6}$$

Our goal is to compute the defender's Nash equilibrium strategies and strong Stackelberg equilibrium strategies, and we call it the **equilibrium computation problem**.

2.3 Curse of Dimensionality and Compact Representation Technique

The Nash equilibrium is equivalent to the strong Stackelberg equilibrium in the zero-sum game. Therefore, we only need to focus on the computation of Nash equilibrium. Invoking the result in the von Neumann's minimax theorem, computing the NE of zero-sum game can be formulated as the following minimax problem,

$$\min_{\mathbf{q} \in \Delta_{N_d}} \max_{\mathbf{p} \in \Delta_{N_a}} U_a(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T \left(\mathbf{B}^a + \mathbf{L}^a\right) \mathbf{q}. \tag{2.7}$$

One standard solution path is transforming the above problem into the following linear programming problem.

$$\begin{aligned} \min_{\mathbf{q},u} \quad & u \\ s.t. \quad & \mathbf{v}^T \left(\mathbf{B}^a + \mathbf{L}^a\right) \mathbf{q} \leq u, \forall \mathbf{v} \in \Delta_{N_a}, \\ & \mathbf{q} \in \Delta_{N_d}. \end{aligned} \tag{2.8}$$

Curse of Dimensionality It is well known that the linear programming problem can be solved in polynomial time of number of variables and constraints by using the interior point method. However, the above linear programming problem contains $N_d + 1$ number of variables and $N_a + N_d$ constraints, which is at least the size of defender's pure strategy space. In the worst case, i.e., the defender can protect any subsets of targets and $N_a = \Theta(2^n)$. Moreover, unlike the traditional security game [10] that assumes that attacker only attack one target, there exists poly(n) number of variables and exponential number of constraints. One can use the cutting plane (ellipsoid method) to get a polynomial time reduction. However, in this problem, due to multiple attacker resources, it becomes a much more complicated issue, and calls for the development of a new theoretical path.

The goal of the rest of this subsection is to develop a technique to compactly and equivalently represent the zero-sum and non-additive security game with only poly(n) variables. To convey our idea more easily, we begin with an example.

We first use gauss elimination on matrices $\mathbf{B}^a$ and $\mathbf{L}^a$ to transform them into row canonical form, which is to left and right multiply such matrices by elementary matrices $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{N_a \times N_a}$ and $\mathbf{F}_1, \mathbf{F}_2 \in \mathbf{R}^{N_d \times N_d}$.

$$\begin{aligned} \min_{\mathbf{q}\in\Delta_{N_d}} \max_{\mathbf{p}\in\Delta_{N_a}} \mathbf{p}^T \left(\mathbf{B}^a + \mathbf{L}^a\right) \mathbf{q} &= \min_{\mathbf{q}\in\Delta_{N_d}} \max_{\mathbf{p}\in\Delta_{N_a}} \mathbf{p}^T \mathbf{E_1} \mathbf{E}_1^{-1} \mathbf{B}^a \mathbf{F}_1^{-1} \mathbf{F_1} \mathbf{q} \\ &\quad + \mathbf{p}^T \mathbf{E}_2 \mathbf{E}_2^{-1} \mathbf{L}^a \mathbf{F}_2^{-1} \mathbf{F}_2 \mathbf{q} \\ &= \min_{\mathbf{q}\in\Delta_{N_d}} \max_{\mathbf{p}\in\Delta_{N_a}} \mathbf{p}^T \mathbf{E}_1 \begin{bmatrix} \mathbf{B}_r^a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}_1 \mathbf{q} \\ &\quad + \mathbf{p}^T \mathbf{E}_2 \begin{bmatrix} \mathbf{L}_s^a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}_2 \mathbf{q}. \end{aligned}$$

where r and s are the rank of matrices $\mathbf{B}^a$, $\mathbf{L}^a$, and $\mathbf{B}_r^a$, $\mathbf{L}_s^a$ are the corresponding non-zero blocks of their row canonical form. If we define the affine transformation: $f_1(\mathbf{p}) = \left(\mathbf{p}^T \mathbf{E}_1\right)^T$, $f_2(\mathbf{p}) = \left(\mathbf{p}^T \mathbf{E}_2\right)^T$, $g_1(\mathbf{q}) = \mathbf{F}_1 \mathbf{q}$ and $g_2(\mathbf{q}) = \mathbf{F}_2 \mathbf{q}$. Let[4]

$$\begin{aligned} \Delta_{N_a}^a &= \{(f_1(\mathbf{p}), f_2(\mathbf{p})) | \mathbf{p} \in \Delta_{N_a}\}, \\ \Delta_{N_d}^d &= \{(g_1(\mathbf{q}), g_2(\mathbf{q})) | \mathbf{q} \in \Delta_{N_d}\}. \end{aligned}$$

we can obtain the following equivalent optimization problem,

$$\min_{(\bar{\mathbf{q}}_1,\bar{\mathbf{q}}_2)\in\Delta^d_{N_d}} \max_{(\bar{\mathbf{p}}_1,\bar{\mathbf{p}}_2)\in\Delta^a_{N_a}} \bar{\mathbf{p}}_1^T \begin{bmatrix} \mathbf{B}_r^a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{q}}_1 + \bar{\mathbf{p}}_2^T \begin{bmatrix} \mathbf{L}_s^a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{q}}_2.$$

Moreover, considering the fact that only the first r elements in vector $\bar{\mathbf{p}}_1$ and $\bar{\mathbf{q}}_1$, and the first s elements in $\bar{\mathbf{p}}_2$ and $\bar{\mathbf{q}}_2$ have non-zero coefficients in the above optimization model, we can further simplify the above optimization problem as

$$\min_{(\bar{\mathbf{q}}_1,\bar{\mathbf{q}}_2)\in H_d} \max_{(\bar{\mathbf{p}}_1,\bar{\mathbf{p}}_2)\in H_a} \bar{\mathbf{p}}_1^T \mathbf{B}_r^a \bar{\mathbf{q}}_1 + \bar{\mathbf{p}}_2^T \mathbf{L}_s^a \bar{\mathbf{q}}_2, \tag{2.9}$$

where the H_a and H_d is obtained by projecting the polytope $\Delta^a_{N_a}$ and $\Delta^d_{N_d}$ to those coordinates belonging to the non-zero blocks.

The basic observation in the above example is that the number of variables in the optimization model (2.9) is equal to the sum of rank $r+s$ of payoff matrices. Based on the rank inequality that the rank of a matrix is less than its dimension, we have that $r, s \leq \min\{N_a, N_d\}$. Since the number of attacker's pure strategies is $N_a = O(n^c) = \text{poly}(n)$. Therefore, there exists at most poly(n) variables in the optimization model (2.9).

The above illustrative derivation provides a possible path to compactly represent the game. However, there exists a significant technical challenge: the elementary matrices $\mathbf{F}_1$, $\mathbf{F}_2$ and their inverse matrices may have an exponential size due to the exponentially large defender's pure strategy space. Hence, the key question is whether we can **find both these elementary matrices** efficiently? To tackle this problem, we first show that payoff matrices $\mathbf{B}^a$ and $\mathbf{L}^a$ can be decomposed as the product of the several simple matrices.

Theorem 2.1 (Decomposition of the Payoff Matrix)
The payoff matrix $\mathbf{M}^a = \mathbf{B}^a + \mathbf{L}^a$ *can be decomposed as*

$$\mathbf{M}^a = \mathbf{E}(\mathbf{D}^b\mathbf{J} + \mathbf{D}^l\mathbf{K}), \tag{2.10}$$

where $\mathbf{D}^b, \mathbf{D}^l \in \mathbb{R}^{N_a \times N_a}$ *are the diagonal matrices with*

$$\mathbf{D}^b_{\sigma(A),\sigma(A)} = B^c(A), \mathbf{D}^l_{\sigma(A),\sigma(A)} = L^c(A), \forall A \in \mathscr{A}.$$

The $\mathbf{E} \in \mathbb{R}^{N_a \times N_a}$ *and* $\mathbf{J}, \mathbf{K} \in \mathbb{R}^{N_a \times N_d}$ *are binary matrices:*

(continued)

[4]The notation $(\cdot, \cdot)$ denotes the concatenation operator of vector.

$$\mathbf{E}_{\sigma(A),\sigma(U)} = \mathbf{1}\{U \subseteq A\}, \forall A, U \in \mathscr{A}$$
$$\mathbf{J}_{\sigma(A),\mu(D)} = \mathbf{1}\{A \subseteq D^c\},$$
$$\mathbf{K}_{\sigma(A),\mu(D)} = \mathbf{1}\{A \subseteq D\}, \forall A \in \mathscr{A}, D \in \mathscr{D}.$$

The common utility is defined as the Möbius transformation [19, 20] of the benefit and loss function $B(U)$ and $L(U)$ for all $U \in 2^{[n]}$,

$$\begin{aligned} B^c(U) &= \sum_{V \subseteq U} (-1)^{|U \setminus V|} B_a(V) \\ L^c(U) &= \sum_{V \subseteq U} (-1)^{|U \setminus V|} L_a(V). \end{aligned} \tag{2.11}$$

As can be seen in Theorem 2.1, we decompose the original exponentially large payoff matrix $\mathbf{M}^a$ into the summation and the product of several simple matrices including binary matrices $\mathbf{E}$, $\mathbf{J}$, $\mathbf{K}$ and two polynomial-sized diagonal matrices $\mathbf{D}^b$ and $\mathbf{D}^l$. Moreover, such a decomposition has a closed-form expression and the elements in those simple matrices can be implicitly represented.

Based on the above decomposition results, we can let the elementary matrices $\mathbf{E}_1 = \mathbf{E}_2 = \mathbf{E}$, $\mathbf{F}_1 = \mathbf{J}$ and $\mathbf{F}_2 = \mathbf{K}$, and the corresponding affine transformation $f(\mathbf{p}) = \mathbf{E}^T\mathbf{p}$ and $g_1(\mathbf{q}) = \mathbf{J}\mathbf{q}$, $g_2(\mathbf{q}) = \mathbf{K}\mathbf{q}$ to yield two polytopes: $\Delta^a_{N_a} = \{f(\mathbf{p}) | \mathbf{p} \in \Delta_{N_a}\}$ and $\Delta^d_{N_d} = \{(g_1(\mathbf{q}), g_2(\mathbf{q})) | \mathbf{q} \in \Delta_{N_d}\}$. Then we can represent the minimax problem (2.7) as

$$\min_{(\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2) \in \Delta^d_{N_d}} \max_{\bar{\mathbf{p}} \in \Delta^a_{N_a}} \bar{\mathbf{p}}^T (\mathbf{D}^b \bar{\mathbf{q}}_1 + \mathbf{D}^l \bar{\mathbf{q}}_2), \tag{2.12}$$

The following definitions are often used in our next step theoretical development.

Definition 2.1 (Support Set) The support set of the non-additive security game is defined as

$$S = \{A \in \mathscr{A} | B^c(A) \neq 0 \text{ or } L^c(A) \neq 0\}. \tag{2.13}$$

and the support index set $\sigma(S) = \{\sigma(A) | A \in S\}$.

Definition 2.2 (Projection Operator) The projection operator $\pi_S : \mathbb{R}^N \to \mathbb{R}^{|S|}$ is

$$\pi_S((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)) = (\ldots, \mathbf{x}_i, \ldots)_{i \in \sigma(S)}, \tag{2.14}$$

and projection of polytope: $\Pi_S(\Delta_N) \triangleq \{\pi_S(\mathbf{x}) | \mathbf{x} \in \Delta_N\}$.

Based on the definition of our support set S and matrices $\mathbf{D}^b$, $\mathbf{D}^l$, only the variables with indices belonging to $\sigma(S)$ have non-zero coefficients. Therefore, we can eliminate those variables with zero coefficients in (2.12) and project the polytopes $\Delta^a_{N_a}$ and $\Delta^d_{N_d}$ into the coordinates with indices belonging to $\sigma(S)$. The further simplified model can be expressed as

Compact Minimax Problem

$$\min_{(\bar{\mathbf{q}}_1,\bar{\mathbf{q}}_2)\in H_d} \max_{\bar{\mathbf{p}}\in H_a} \bar{\mathbf{p}}^T(\widetilde{\mathbf{D}}^b\bar{\mathbf{q}}_1 + \widetilde{\mathbf{D}}^l\bar{\mathbf{q}}_2), \tag{2.15}$$

where[5] $H_a = \Pi_S(\Delta^a_{N_a})$, $H_d = \Pi_S(\Delta^d_{N_d})$, matrix $\widetilde{\mathbf{D}}^b$ and $\widetilde{\mathbf{D}}^l$ is obtained by extracting the non-zero columns and rows of matrix $\mathbf{D}^b$ and $\mathbf{D}^l$.

Since the size of the support set $|S| \leq N_a$, and $N_a = \text{poly}(n)$, we arrive at a compact representation of the non-additive security game with only poly(n) variables. Note that in the above compact representation framework, the affine transformation f_1 and f_2 are the same as in our compact representation. The following theorem guarantees the correctness of our compact representation.

Theorem 2.2 (Compact Representation) *$(\mathbf{p}^*, \mathbf{q}^*)$ is a Nash equilibrium of zero-sum non-additive security game if and only if $(\pi_S(f(\mathbf{p}^*)), (\pi_S(g_1(\mathbf{q}^*)), \pi_S(g_2(\mathbf{q}^*)))$ is the optimal solution of compact minimax problem (2.15).*

2.4 Oracle-Based Algorithmic Framework

In the previous section, we develop a compact representation technique such that one can equivalently represent the original NASG by a minimax problem with a polynomial number of variables, which can be further solved by the following linear programming model,

Compact Linear Programming

$$\begin{aligned} \min \quad & u \\ s.t. \quad & \mathbf{v}^T(\widetilde{\mathbf{D}}^b\bar{\mathbf{q}}_1 + \widetilde{\mathbf{D}}^l\bar{\mathbf{q}}_2) \leq u, \forall \mathbf{v} \in I_a, \\ & (\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2) \in H_d, \end{aligned} \tag{2.16}$$

where I_a denotes the set of vertices of the convex polytope H_a. The above linear programming problem has poly(n) number of variables and potentially exponential number of constraints (due to the membership constraint $(\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2) \in H_d$). This motivates us to utilize the ellipsoid method to solve the problem.

[5] Note that each vector in $\Delta^d_{N_d}$ is consists of two parts $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$. Here the corresponding low-dimensional point is $(\pi_S(g_1(\mathbf{q}), \pi_S(g_2(\mathbf{q}))$.

2.4.1 Preliminaries

Let H be a non-empty convex polytope in $\mathbb{R}^n$. Given a vector $\mathbf{w} \in \mathbb{R}^n$, one wants to find a solution to $\max_{\mathbf{x} \in H} \mathbf{w}^T \mathbf{x}$. By "linear optimization over H", we mean solving the problem $\max_{\mathbf{x} \in H} \mathbf{w}^T \mathbf{x}$ for any $\mathbf{w} \in \mathbb{R}^n$. A separation problem for H is that, given a vector $\mathbf{x} \in \mathbb{R}^n$, decide if $\mathbf{x} \in H$, and if not, find a hyperplane which separates $\mathbf{x}$ from H. The following results are due to Grötschel et al. [21].

Theorem 2.3 (Separation and Optimization) *Let $H \in \mathbb{R}^n$ be a convex polytope. There is a* poly *(n) time algorithm to solve the linear optimization problem over H if and only of there is a* poly *(n) time algorithm to solve the separation problem for H.*

Theorem 2.4 (Separation and Convex Decomposition) *Let $H \in \mathbb{R}^n$ be a convex polytope. If there is a* poly *(n) time algorithm to solve the separation problem for H, then there is a* poly *(n) time algorithm that, given any $\mathbf{x} \in H$, yields $(n+1)$ vertices $\mathbf{v}^1, \ldots, \mathbf{v}^{n+1} \in H$ and convex coefficients $\lambda_1, \ldots, \lambda_{n+1}$ such that $\mathbf{x} = \sum_{i=1}^{n+1} \lambda_i \mathbf{v}^i$.*

2.4.2 Reduction Between NASG and Combinatorial Optimization

The main result in this subsection is captured in the following theorem.

Theorem 2.5 (NE Computation and Defender Oracle Problem) *There is a* poly *(n) time algorithm to compute the defender's Nash equilibrium (strong Stackelberg equilibrium),* ***if and only if*** *there is a* poly *(n) time algorithm to compute the defender oracle problem: for any given vector $\mathbf{w} \in \mathbb{R}^{2|S|}$, determine,*

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in I_d} \mathbf{w}^T \mathbf{x}. \tag{2.17}$$

To obtain above reduction, we adopt the following path: we first show how the compact problem and the defender oracle problem can be reduced to each other in poly(n) time; then we exploit the geometric structure of polytope H_a and H_d to construct two poly(n) time vertex mapping algorithms to obtain the reduction between the equilibrium computation and the compact problem. This whole procedure also produces an algorithmic framework to the solve the NASG.

The polynomial time reduction between the defender oracle problem and the compact linear programming problem can be easily obtained by the ellipsoid method. The key lies in how to obtain the reduction between the equilibrium

computation (2.7) and the compact linear programming problem. Actually, there exist two issues: first, how to transform the input instance of each problem to the other one in poly(n) time; second, how to map the optimal solution of each problem to the other in poly(n) time. Since the input of the equilibrium computation problem are the utility functions $\{B(U)\}$ and $\{L(U)\}$ and the input of compact problem are the common utilities $\{B^c(U)\}$ and $\{L^c(U)\}$ (all the elements of matrices $\mathbf{D}^b$ and $\mathbf{D}^l$ are the common utilities), such transformation can be completed in $O(2^c n^c) = \text{poly}(n)$ time based on the definition of common utilities.

To resolve the second issue, we first consider how to map the optimal solution of compact problem to the defender's optimal mixed strategies. Based on Theorem 2.4, we obtain that if the separation problem of LP (2.16) can be solved in poly(n) time, we can decompose any feasible point $\mathbf{x}$ into a convex combination of at most $(2|S|+1)$ vertices of the polytope defined by those constraints. Note that this is precisely the DOP required for above reduction. Applying this result to the optimal solution $(\mathbf{q}_1^*, \mathbf{q}_2^*)$ of the LP (2.16), we can get a convex decomposition that

$$(\mathbf{q}_1^*, \mathbf{q}_2^*) = \sum_{i=1}^{2|S|+1} \lambda_i (\mathbf{v}_1^i, \mathbf{v}_2^i), \tag{2.18}$$

where $(\mathbf{v}_1^i, \mathbf{v}_2^i) \in I_d$. The basic fact is that the defender's mixed strategy can be regarded as a convex combination of its pure strategies, each of which corresponds to a vertex of simplex Δ_{N_d}. If we can map the vertices $(\mathbf{v}_1^i, \mathbf{v}_2^i)$ back to the vertices (pure strategy) of the original game, denoted by $h((\mathbf{v}_1^i, \mathbf{v}_2^i))$, the mixed strategies of the defender can be expressed as

$$\mathbf{q}^* = \sum_{i=1}^{2|S|+1} \lambda_i h((\mathbf{v}_1^i, \mathbf{v}_2^i)). \tag{2.19}$$

Thus, the key lies in how to compute $h((\mathbf{v}_1^i, \mathbf{v}_2^i))$ in poly(n) time.

To tackle this problem, we need to investigate the geometric structure of polytope H_d. First, considering an arbitrary defender's pure strategy $D \in \mathscr{D}$, the corresponding vertex in Δ_{N_d} is a unit vector $\mathbf{e}^D \in \mathbb{R}^{N_d}$ with only one non-zero element $\mathbf{e}^D_{\mu(D)} = 1$. Based on the definition of the transformation $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$, the corresponding point of polytope H_d is

$$(g_1(\mathbf{e}^D), g_2(\mathbf{e}^D)) = (\mathbf{J}\mathbf{e}^D, \mathbf{K}\mathbf{e}^D) = (\mathbf{J}_{\mu(D)}, \mathbf{K}_{\mu(D)}), \tag{2.20}$$

where $\mathbf{J}_{\mu(D)}$ and $\mathbf{K}_{\mu(D)}$is the $\mu(D)$th column of matrix $\mathbf{J}$ and $\mathbf{K}$. Then the corresponding point $\mathbf{v}^D$ of the projected polytope H^d is

$$\mathbf{v}^D = \left(\pi_S(\mathbf{J}_{\mu(D)}), \pi_S(\mathbf{K}_{\mu(D)})\right), \tag{2.21}$$

which is the sub-vector of $\mathbf{J}_{\mu(D)}$ and $\mathbf{K}_{\mu(D)}$. The problem is that the vertex in the high-dimensional polytope may not project to a vertex of its low-dimensional image. However, the following lemma will provide a positive result.

Lemma 2.1 (Geometric Structure of H_d)
For any support set $[n] \in S \in \mathscr{A}$, the vertices of the polytope H_d are the columns of the sub-matrix of $\begin{bmatrix} \mathbf{J} \\ \mathbf{K} \end{bmatrix}$, which is formed by extracting the row whose index belongs to $\sigma(S)$.

Since we have a closed-form expression of the matrix $\mathbf{J}$ and $\mathbf{K}$, we can construct a vertex mapping algorithm from low-dimensional vertex to the defender's pure strategy. The efficiency and the correctness of Algorithm 1 is justified by following lemma.

Algorithm 1: Vertex mapping from vertex to pure strategy

input : Vertex $(\mathbf{v}_1, \mathbf{v}_2) \in I^d$.
output: Defender's pure strategy D.

$T = \emptyset$;
for *each* $i \in [n]$ **do**
  Examine each coordinate of vertex:
  if $\mathbf{v}_{1,\sigma(\{i\})} \neq 0$ **then**
    $T = T \cup \{i\}$;
  end
end
$D = T^c$;

Lemma 2.2 (Correctness of Vertex Mapping Algorithm) *The vertex mapping Algorithm* 1 *runs in $O(n)$ time and maps each vertex of H_d to a unique pure strategy.*

Note that our vertex mapping algorithm only examines n instead of all the coordinates of each vertex of H_d to recover a defender's pure strategy. The reason behind this result is that there exists a one-to-one correspondence between each pure strategy and those n coordinates of each vertex of polytope H_d. Intuitively, those n coordinates of each vertex of H_d is binary and therefore there exists possibly 2^n possibilities, each of which corresponds to a pure strategy.

The other direction follows from the following argument. Suppose that the problem of equilibrium computation is solved in poly (n) time and the optimal defender's mixed strategy is denoted by $\mathbf{q}^*$. Invoking a known result in game theory (Theorem 4 in [22]), the support size, i.e., number of strategies with nonzero probability, of the Nash equilibrium is less than the rank of the payoff matrix. Since the rank of payoff matrix $\mathbf{M}^a$ is $O(n^c)$, the number of non-zero coordinates in $\mathbf{q}^*$ is at most $O(n^c) =$ poly(n) and $\mathbf{q}^*$ can be expressed as

$$\mathbf{q}^* = \sum_{i=1}^{\text{poly}(n)} \lambda_i \mathbf{e}^i. \tag{2.22}$$

Therefore, we can determine the optimal solution of the compact problem in poly(n) time by constructing the following poly(n) time vertex mapping algorithm from a pure strategy $\mathbf{e}^i$ to a vertex of H_d.

Algorithm 2: Vertex mapping from pure strategy to vertex

input : Defender's Pure Strategy D.
output: Vertex $\mathbf{v}^D \in I_d$

$T = \emptyset$;
for *each* $V \in \mathscr{A}$ **do**
  if $V \subseteq D^c$ **then** $\mathbf{v}^D_{1,\sigma(V)} = 1$;
  else $\mathbf{v}^D_{2,\sigma(V)} = 0$;
end
Output vertex $\mathbf{v}^D = (\mathbf{v}^D_1, \mathbf{v}^D_2)$;

The intuition behind this result is similar to the previous vertex mapping algorithm and the correctness of Algorithm 2 is guaranteed by the following lemma.

Lemma 2.3 (Correctness of Vertex Mapping Algorithm)
Vertex mapping Algorithm 2 *runs in* $O(n^c)$ *time and maps each defender's pure strategy* D *to a unique vertex of* H_d.

Combining all the above results together, we provide a general algorithmic framework shown next.

Algorithm 3: General algorithmic framework for non-additive security game

1. **Utility transformations:** Transform the original utility functions $\{B(U)\}$ and $\{L(U)\}$ to the corresponding common utilities $\{B^c(U)\}$ and $\{L^c(U)\}$ based on Möbius transformation;
2. **Solve the compact problem:** Solve the linear program (2.16) to obtain the optimal compact strategy $\mathbf{t}^*$ by ellipsoid method;
3. **Convex decomposition:** Decompose optimal compact strategy $\mathbf{t}^*$ into the convex combination: $\mathbf{t}^* = \sum_{i=1}^{n+1} \lambda_i \mathbf{v}^i$ by exactly solving the defender oracle problem;
4. **Vertex mapping:** Map each vertex $\mathbf{v}^i$ to a defender pure strategy D_i by Algorithm 1, output the defender's NE strategy:
   play pure strategy D_i with probability λ_i, $1 \le i \le n+1$;

2.4.3 Applications

In this subsection, we will discuss the applications of our developed algorithmic framework to several security domain problems.

2.4.3.1 Network Security Game

The network security game [14, 23] is given by the following definitions.

Definition 2.3 A network security game is given by the tuple $(G, T, \mathbf{F_a}, c)$, where $G = (V, E)$ with node set V, edge set E, T is the network value function, $\mathbf{F_a}$ is the failure operator, c is the maximum number of nodes the attacker can choose, while the defender can protect any target.

The network value function $T : G \to \mathbb{R}$ is a security measure assessing the utility of a network, and failure operator $\mathbf{F_a} : 2^G \to 2^G$ is to generate a new network via a specific failure mode after removing some nodes. For example, Shakarian et al. [23] adopt the number of connected load nodes as T, and edge cascading failure model as $\mathbf{F_a}$. We next discuss several classical network security games *that can be solved in polynomial time*.

Example 2.3 (Security Game in a Tree Network) In cybersecurity, the sensor network often exhibits a tree topology. The game is such that the attacker attempts to invade some nodes to destroy the connectedness of the network and the IT manager is required to deploy anti-virus software in some nodes. Suppose that the network G consists of m connected components: $V_1, V_2, \ldots, V_m$ and both players adopt the following network value functions

$$T(G) = \max_{1 \le i \le m} |V_i|. \tag{2.23}$$

In practice, we assume that the attacker can simultaneously invade at most two nodes, i.e., $c = 2$. Then, if node i is attacked, the tree G is divided into 2 sub-trees: G_{i1} and G_{i2}, and the benefit is given by

$$B(\{i\}) = n - \max\{|G_{i1}|, |G_{i2}|\} = \min\{n - |G_{i1}|, n - |G_{i2}|\}.$$

Similarly, if node j is attacked, the tree G is divided into 2 sub-trees: G_{j1} and G_{j2}, and the benefit is given by

$$B(\{j\}) = n - \max\{|G_{j1}|, |G_{j2}|\} = \min\{n - |G_{j1}|, n - |G_{j2}|\}.$$

Without loss of generality, suppose $j \in G_{i2}$, then if nodes i, j are simultaneously attacked, the tree G is divided into 3 subtrees: G_{i1}, G_{i21} and G_{i22}, where the latter two are obtained by dividing G_{i2}. The corresponding benefit is given by

$$B(\{i,j\}) = n - \max\{|G_{i1}|, |G_{i21}|, |G_{i22}|\} = \min\{n - |G_{i1}|, n - |G_{i21}|, n - |G_{i22}|\}.$$

Then one can easily show that the following holds true for any $i, j \in [n]$,

$$B(\{i, j\}) \leq B(\{i\}) + B(\{j\})$$

and $B^c(\{i, j\}) \leq 0$. Combining this result with Theorem 2.5, one can easily show that the defender oracle problem is a submodular minimization problem, which can be solved in polynomial time. Further, we can use Algorithm 1 to determine an equilibrium strategy in polynomial time.

Example 2.4 (Security Game in a Sparse Network) As can be seen in (Fig. 2.4), the real world network is extremely sparse and the largest connect component is always small compared to the network scale, i.e, $O(\log(n))$. In this case, we have the following result.

Lemma 2.4 *A network security game* $(G, T, \mathbf{F_a}, c)$ *can be solved in* poly (n) *time if the largest connected component of* G *is* $\Theta(\log(n))$.

The basic intuition is that, when the network is extremely sparse such that the largest connected component of G is $\Theta(\log(n))$, the common utility functions defined in (2.11) will satisfy a separable condition

$$U = \bigcup_{i=1}^{m} U_i, \forall U_i \subset V_i, U_i \neq \emptyset$$
$$\Longrightarrow B^c(U) = C_a^c(U) = C_d^c(U^c) = 0.$$

Then, one can easily show that the defender oracle problem can be separated into $O(n)$ subproblems, each of which can be solved in polynomial time. Combining this result with Theorem 2.5, we can solve this network security game in polynomial time.

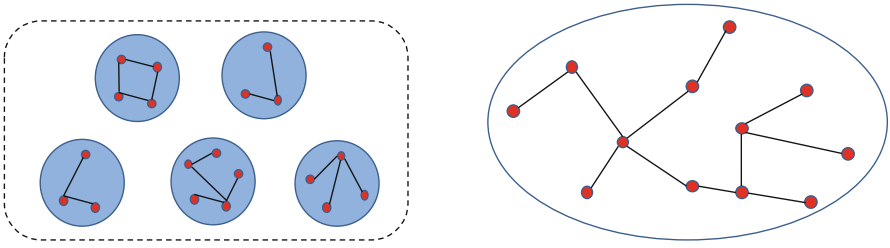


Fig. 2.4 Security game in a sparse network and tree network

2.4.3.2 Security Game with Multiple Attacker Resources

There exists several other important applications of our developed algorithmic framework.

LAX Airport Checkpoint Placement Problem [24] This problem is one of the earliest applications of security games. In this setting, the security force has k police officers that are to be deployed across n (where $k<n$) checkpoints. Each police officer can be deployed at any given check point. Therefore, any subset of $[n]$ of size at most k is a defender pure strategy. Korzhyk et al. [25] extends this game model into the multiple attacker resources and shows that this problem can still be solved in poly(n) time by a state transition algorithm [25]. In our framework, the DOP is the linear optimization over a uniform matroid.

$$\begin{aligned} &\max \quad \mathbf{w}^T\mathbf{x} \\ &s.t. \quad \textstyle\sum_{i=1}^{n} \mathbf{x}_i \le k, \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{2.24}$$

The above problem can be solved in polynomial time by summing the k largest elements of vector $\mathbf{w}$. Thus, it verifies previous results.

In the following three cases, the defender's resources are heterogeneous such that there exists some practical constrains in the set system ε.

Geographic Constrained Patrolling Problem In the patrolling problem, due to geographic constraints, the police officer can only patrol the area around the station. In this case, the resources of different defenders (police) can defend different groups of targets. In our framework, we can construct a weighted bipartite graph as follows: (1) two disjoint sets U, V , where U represents all the nodes, and V represents all the resources; (2) there exists an edge between the node u in U and node v in V if the resource v can cover node u; (3) associate each edge (u,v) with a weight $\mathbf{w}_u$ ($\mathbf{w}$ is the vector in the DOP). Then the DOP is a weighted bipartite matching problem, which can be solved in polynomial time by Hungarian algorithm.

Federal Air Marshal Scheduling Problem [10] In such applications, one air marshal is assigned to protect several sequential flights with the constraint that any destination of the previous flight is the departure of the next flight. The objective is to cover all current flights. In [26], the authors investigate this problem under single attacker resources and shows the polynomial solvability in some cases and NP-hardness in other cases. However, attackers may initiate simultaneous attacks (e.g., the flights of 911) and there still does not exist any efficient algorithm. In our framework, we can construct the following weighted set cover problem: let the node set $[n]$ be the universe and all the air marshals constitute the collection S of subsets of $[n]$; then associate the weight $\mathbf{w}$ to each element of the universe. Then, the DOP is a weighted set cover problem and our results show that when the attacker has multiple resources, the problem is generally NP-hard but we can still solve this problem in some cases. For example, if each air marshal can protect at most two

flights (a pair of round trip flights), the set system ε indeed encodes the weighted 2-cover, which can be solved in poly(n) time.

Spatio-Temporal Security Game [12,27] In many applications of security games, an important class is the spatio-temporal security game. This kind of game is used to model the games played in the spatio-temporal spaces such as planning patrol boats of the US Coast Guard [12], wildlife protection [27]. The current solution technique of this game is to discretize the space and time and build 2-D gird, in which the security force patrol the points. Combining the results in [28], we can show that spatio-temporal security game with multiple attacker resources are indeed a min-cost flow problem, which can be solved in poly(n) time.

There exist other applications that can be cast in our framework such as passenger screening for the Transportation Security Administration [11]. Indeed, based on our general framework in Algorithm 3, all the results under the single attacker resources can be directly extended to the scenario of multiple attacker resources.

2.5 Approximated Equilibrium Computation by Low Rank Decomposition

In the previous section, we have developed a compact representation technique and algorithmic framework such that one can reduce the problem of determining the equilibrium point of NASG to a combinatorial optimization problem. However, one pessimistic result is that the defender oracle problem in general is NP-hard, which is high-complexity to be solved in practice. A natural question is the following: in practical network security games, can we still efficiently solve an equilibrium point. Actually, one crucial observation is that *the common utility in realistic networks is concentrated around zero.*

In Fig. 2.5, we examine the distributions of the benefit function and its common utility function in the following two kinds of network: Erdös-Renyi network $G(n, p)$ and scale-free network $G(n, \alpha)$, where n is the number of nodes, p is the probability that any two nodes are connected, α is the parameter of degree distribution of the scale-free network. Suppose that the network G consists of m connected components: $V_1, V_2, \ldots, V_m$ and we adopt the following two kinds of network value functions,

$$T_1(G) = \max_{1 \le i \le m} |V_i|, T_2(G) = \sum_{i=1}^{m} |V_i|^2.$$

The different form of network value functions have different assessment of the network. The detailed comparison can be found in [14]. As can be seen in Fig. 2.5, in both Erdös-Renyi and scale-free networks, although the distribution of the benefit function is random, the distribution of the common utility function is concentrated around zero and 90% of them are less than 0.05. In particular, when the number of

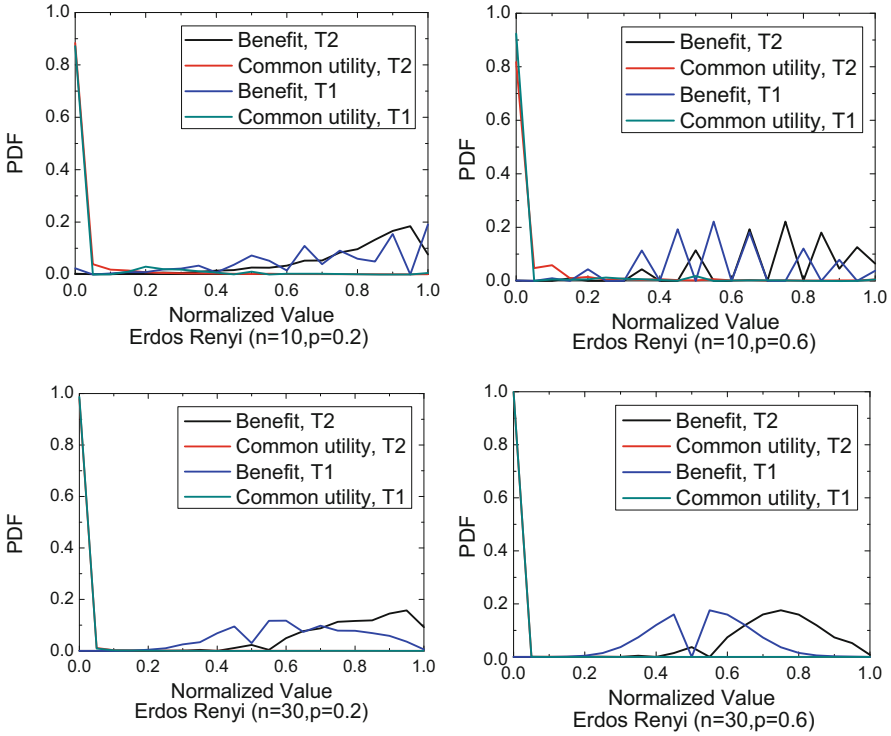


Fig. 2.5 The distributions of common utility function and benefit function. All their value are absolute value and normalized in [0, 1]

nodes increases, this phenomenon is amplified such that almost 99% of the common utility functions are less than 0.05.

Based on the above observation, we can let most of the common utility functions equal to 0 according to a given threshold ε_c. Formally, let $\tilde{B}^c(\cdot)$ denote the new common utility function generated by Algorithm 2, then the corresponding approximate benefit function satisfies

$$|\tilde{B}(U) - B(U)| = \left| \sum_{W \subseteq U} \tilde{B}^c(W) - \sum_{W \subseteq U} B^c(W) \right|$$
$$\leq \sum_{W \subseteq U} \left| \tilde{B}^c(W) - B^c(W) \right| \leq 2^{|U|} \epsilon_c.$$

Since $|U| \leq c$, the maximum error between the original benefit functions and new generated benefit functions is less than $2^c \varepsilon_c$. A classic result of game theory is that, if the maximum difference between the elements of two payoff matrices is bounded by ε, the difference of the optimal game values yielded by these two payoff matrices

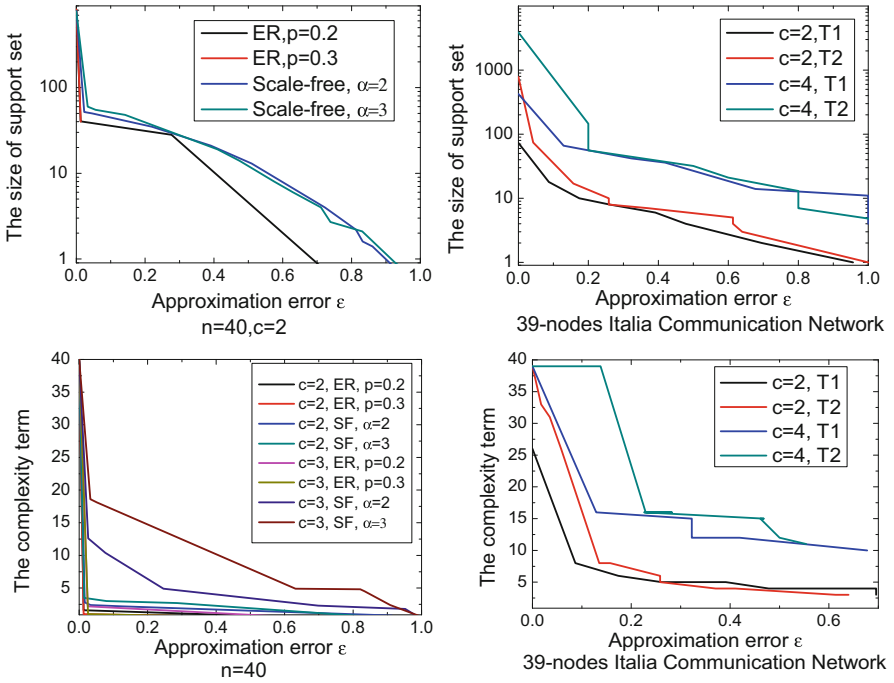


Fig. 2.6 Top: the size of support set $|S|$ versus approximation error ε; Bottom: the complexity term $\max_i |U_i|$ versus approximation error ε. Remark that the ε represents the approximation error of the game value. Note that SF denotes the scale-free network

are bounded by 2ε [22]. Therefore, the approximation error of our game value is bounded by $2^{c+1}\varepsilon_c$.

As shown at the top of Fig. 2.6, for the Erdös Renyi, scale-free and Italian communication network, the size of support set will be reduced 90% by an extremely small approximation error 0.05. Moreover, this process also leads to a separable structure of S, and the resulting complexity of solving the NASG is $\text{poly}(n)O(2^{\max_i |U_i|})$. For example, in the bottom of Fig. 2.6, the complexity term $\max_i |U_i|$ can be greatly reduced to the order of $\Theta(\log(n))$ with an approximation error of 1%, regardless of the size and density of the network, and how many targets the attacker can choose. More comprehensive numerical results can be found in [15]. In summary, our approximation framework can reduce the complexity term $\max_i |U_i|$ to order $\Theta(\log(n))$ by only 10% approximation error in most networks including Erdös-Renyi, scale-free network and a 39-nodes Italian communication network. Therefore, using our theoretical framework, we can **approximately and compactly represent a realistic network security game and solve it in poly(n) time with high accuracy**.

2.6 Future Research Directions

In this section, we outline several future research directions.

2.6.1 Learning-Based Proactive Network Defense

In our proposed NASG, we suppose that probability distribution of attacker type is known to the defender and regarded as a prior belief of defense group, and can be formed by the Bayesian rule. However, in practical settings, some of the information might be unknown to the defense group. This problem can be investigated by incorporating a learning framework into our Non-additive Security Game based on the following two scenarios: (1) full information setting: both the attacker type and action is known in each time slot t. We need to construct an online learning algorithm to form the belief of attacker type distribution; (2) partial information setting: only the attacker action is known in each time slot t. This kind of problem can be cast into a multi-arm bandits setting. The key challenge here will be in designing algorithms that provide a small (sublinear) regret.

2.6.2 Game-Theoretical Network Defense with Boundedly Rational Players

In our proposed NASG, we suppose that all the players are fully rational. However, in real situations, the players such as civilians will have bounded rationality. To model this behavior, the quantal response equilibrium is a more appropriate solution concept. The challenge is that, due to the introduction of the quantal response model, such an optimization problem has a non-convex fractional objective function, which is generally hard to solve. The goal lies in how to transform such an problem into a sequence of convex optimization problem and solve each sub-problem efficiently.

2.6.3 Multi-Scale Proactive Network Defense

In the previous sections, we have already discussed the general model and algorithmic framework of game-theoretical proactive network defense. However, in the future battlefields, there exists multiple factors that will greatly change the current

game structure. For example, internet of things make the network structure highly dynamic. Other key factors includes:

Multi-Party Games In this game there exist multiple players in both the defense and adversarial groups. In addition, in practice, there may also exist "neutral players" that could potentially be influenced by the strategies of the attackers and defenders. These kinds of dependency are sometimes characterized by the underlying social networks formed among all the players. For example, in the battle with adversarial group Lashkar-e-Taiba, the players in the defense group would include the US and Indian Governments, as well as other peaceful nationals. They share the defense resources and cooperate with each other. In contrast, the players in the opposing groups include training camps, military bases, and get political supports from diaspora and foreign states. The players in the neutral group can be regarded as civilians or the weak peaceful groups in Pakistan.

Multi-Genre Networks In real scenarios, there exists some linkage structure among different infrastructures due to the effect of the underlying multi-genre networks. One well-known example is the interdependence network formed by power grid and communication systems [29]. Due to the dependencies among different targets, attacking one target will influence other targets. For instance, an attacker attempts to destroy the connectivity of a network and the defender aims to protect it. The strategy for both players is to choose the nodes of the network to either protect or attack. If there are two nodes that constitute a bridge in this network or inter-dependent network, successfully attacking both of them will split the network into two parts and incur a huge damage, while attacking any one of them may have limited impact.

Actually, as shown in Fig. 2.7, we can generalize NASG to Multi-Stage Multi-Party Bayesian Security Game with considering the interaction between multi-genre networks, multi-parties and uncertain attacker behavior. It contains a time horizon $T = \{1, 2, \ldots, t\}$ and runs Multi-Party Bayesian Security Games in each time slot t, which contains three kinds of players: defenders, attackers and neutrals. Social links could exist among some of the players in different groups such that the decision making of different players are dependent on each other. Each player i in the adversarial group is from a set of possible types θ_i (multiple adversary types trying to infiltrate security). The defender has a belief $p[t]$ of the attacker's uncertainty, which is a probability distribution over all the adversarial players' types. The belief $p[t]$ is a prior of defender before playing the game in time slot t, and can be formed by a Bayesian rule and learning the actions of all the agents in the previous time slots. The objective of the MMBSG is to calculate the mixed strategy (a probability distribution over each pure strategies) Nash Equilibrium (NE) in each time slot t, and the key lies in how the solve this game efficiently based on our previously developed technique.

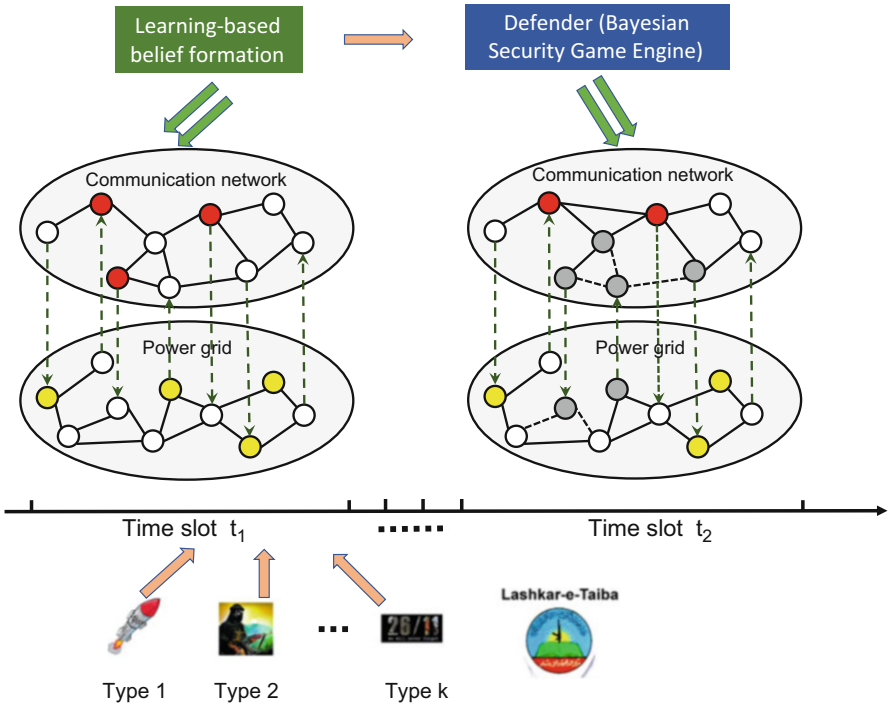


Fig. 2.7 Overview of our proposed compositional game theory framework consisting of an interdependent network between power grid and communication network. The attacker has multiple attacking types. The defender needs a learning-based belief formation regarding attacking types, and then determine an equilibrium strategy

2.7 Conclusion

In this Chapter, we have aimed to illustrate that game theory can provide a sound mathematical approach to combat attacks across a wide range of applications. However, to do this one most go beyond the existing game theoretic models that typically assume additive utility functions, or that the attacker can attack only one target. While such assumptions have lead to tractable analyses, they miss key inherent dependencies that exist among different targets in current complex networks. In this chapter, we generalize the traditional security game model to the network scenario capturing network dependencies and the possibilities of a coordinated multi-resource attacks. We show that each security game is equivalent to a combinatorial optimization problem over a set system, which consists of defender's pure strategy space. The key technique we use is based on projection of a polytope based transformation, and the ellipsoid method. While in its most generality, capturing the equilibria under such an intricate model, is computationally hard, we provide several important classes of real-life problems for which our

techniques can be used to develop optimal defense mechanisms. Based on our new mathematical framework, we outline a number of important future directions that can be investigated. The area of game theory coupled with reinforcement learning is fertile ground for solving many important security related problems.

Acknowledgement This work has been funded in part by a grant from the Army Research Office W911NF-15-1-0277.

References

1. R. Anderson, Why information security is hard - an economic perspective, in *Proceedings of ACSAC*, 2001
2. T. Moore, R. Anderson, Economics and internet security: a survey of recent analytical, empirical and behavioral research (2011). ftp://ftp.deas.harvard.edu/techreports/tr-03-11.pdf
3. T. Alpcan, T. Basar, *Network Security: A Decision and Game-Theoretic Approach* (Cambridge University Press, Cambridge, 2010)
4. L. Buttyan, J.P. Hubaux, *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing* (Cambridge University Press, Cambridge, 2007)
5. H. Kunreuther, G. Heal, Interdependent security. J. Risk Uncertain. **26**(2–3), 231–249 (2003)
6. M.H. Manshaei, Q. Zhu, T. Alpcan, T. Basar, Game theory meets network security and privacy. ACM Comput. Surv. **45**, 25 (2012)
7. A. Laszka, G. Horvath, M. Felegyhazi, L. Buttyan, Flipthem: modeling targeted attacks with flipit for multiple resources, in *Proceedings of GameSec*, 2014
8. M. Zhang, Z. Zheng, N.B. Shroff, A game theoretic model for defending against stealthy attacks with limited resources, in *GameSec 2015*, November 2015, London (Springer, Cham, 2015)
9. Z. Zheng, N.B. Shroff, P. Mohapatra, When to reset your keys: optimal timing of security updates via learning, in *AAAI'17*, San Francisco, CA, February 2017
10. J. Tsai, C. Kiekintveld, F. Ordonez, M. Tambe, S. Rathi, IRIS-a tool for strategic security allocation in transportation networks, in *Eighth International Joint Conference on Autonomous Agents and Multiagent Systems (Industry Track)*, May 2009
11. M. Brown, A. Sinha, A. Schlenker, M. Tambe, One size does not fit all: a game-theoretic approach for dynamically and effectively screening for threats, in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016
12. F. Fang, A. Xin Jiang, M. Tambe, Optimal patrol strategy for protecting moving targets with multiple mobile resources, in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* International Foundation for Autonomous Agents and Multiagent Systems (2013), pp. 957–964
13. M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned* (Cambridge University Press, Cambridge, 2011)
14. A. Gueye, V. Marbukh, J.C. Walrand, Towards a metric for communication network vulnerability to attacks: a game theoretic approach, in *International Conference on Game Theory for Networks* (Springer, Berlin, 2012)
15. F.L. Sinong Wang, N.B. Shroff, Non-additive security games, in *AAAI*, 2017
16. S. Wang, N. Shroff, Security game with non-additive utilities and multiple attacker resources, in *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1 (2017), p. 13
17. H. Von Stackelberg, *Marktform und gleichgewicht* (J. Springer, Berlin, 1934)

18. B. Von Stengel, S. Zamir, Leadership with commitment to mixed strategies, vol. 38. Technical report LSE-CDAM-2004-01, CDAM research report (2004)
19. F.V. Fomin, D. Kratsch, *Exact Exponential Algorithms* (Springer, Berlin, 2010)
20. R. Kennes, P. Smets, Computational aspects of the Mobius transformation, in *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 401–416 (Elsevier Science Inc., Amsterdam, 1990)
21. M. Grotschel, L. Lovasz, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2), 169–197 (1981)
22. R.J. Lipton, E. Markakis, A. Mehta, Playing large games using simple strategies, in *Proceedings of the 4th ACM Conference on Electronic Commerce (EC)*, pp. 36–41 (ACM, New York, 2003)
23. P. Shakarian, H. Lei, R. Lindelauf, Power grid defense against malicious cascading failure, in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems (2014)
24. J. Pita, M. Jain, J. Marecki, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, S. Kraus, Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport, in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track*, pp. 125–132. International Foundation for Autonomous Agents and Multiagent Systems (2013)
25. D. Korzhyk, V. Conitzer, R. Parr, Security games with multiple attacker resources, in *IJCAI Proceedings - International Joint Conference on Artificial Intelligence*, vol. 22 (2011), pp. 273–279. Citeseer
26. D. Korzhyk, V. Conitzer, R. Parr, Complexity of computing optimal Stackelberg strategies in security resource allocation games, in *AAAI*, 2010
27. F. Fang, T.H. Nguyen, R. Pickles, W.Y. Lam, G.R. Clements, B. An, A. Singh, M. Tambe, A. Lemieu, Deploying PAWS: field optimization of the protection assistant for wildlife security, in *Proceedings of the Twenty-Eighth Innovative Applications of Artificial Intelligence Conference*, 2016
28. H. Xu, F. Fang, A.X. Jiang, V. Conitzer, S. Dughmi, M. Tambe, Solving zero-sum security games in discretized spatio-temporal domains, in *AAAI* (2014), pp. 1500–1506, Citeseer
29. S.V. Buldyrev et al., Catastrophic cascade of failures in interdependent networks. Nature **464**(7291), 1025 (2010)