Strategic Attack & Defense in Security Diffusion Games

MARCIN WANIEK, Computer Science Department, New York University, UAE and Masdar Institute, Khalifa University of Science and Technology, UAE and Institute of Informatics, University of Warsaw, Poland
TOMASZ P. MICHALAK, Institute of Informatics, University of Warsaw, Poland
AAMENA ALSHAMSI, Masdar Institute, Khalifa University of Science and Technology, UAE

Security games model the confrontation between a defender protecting a set of targets and an attacker who tries to capture them. A variant of these games assumes security interdependence between targets, facilitating contagion of an attack. So far, only stochastic spread of an attack has been considered. In this work, we introduce a version of security games, where the attacker strategically drives the entire spread of attack and where interconnections between nodes affect their susceptibility to be captured.

We find that the strategies effective in the settings without contagion or with stochastic contagion are no longer feasible when spread of attack is strategic. While in the former settings it was possible to efficiently find optimal strategies of the attacker, doing so in the latter setting turns out to be an NP-complete problem for an arbitrary network. However, for some simpler network structures, such as cliques, stars, and trees, we show that it is possible to efficiently find optimal strategies of both players. For arbitrary networks, we study and compare the efficiency of various heuristic strategies. As opposed to previous works with no or stochastic contagion, we find that centrality-based defense is often effective when spread of attack is strategic, particularly for centrality measures based on the Shapley value.

CCS Concepts: • **Security and privacy** → **Formal security models**; • **Theory of computation** → *Social networks*; • **Networks** → Network security;

Additional Key Words and Phrases: Security games, social networks, network diffusion

1 INTRODUCTION

Security games [51]—a particular type of Stackelberg games [64]—has recently attracted growing interest in the literature. These games consider a confrontation between two players: a defender,

e.g., a police force trying to protect certain resources or locations, and an attacker, e.g., a criminal organization intending to gain control over the locations or the resources. One of the milestones proving the effectiveness of such models is the ARMOR system used to protect the LAX airport [48]. Its success led to the development of many other systems based on security games [54, 60, 62].

Some security games settings consider a potential spread of the attack after hitting the initial target [32, 37]. In particular, Bachrach et al. [5] created a variation of the security game model with stochastic contagion between two targets. The idea was later expanded by adding a whole network of interdependencies between targets [1, 41]. However, so far the literature on security games has addressed only the stochastic spread of attacks. While this fits certain scenarios such as the spread of computer viruses or fires, it is not suitable for scenarios in which the attack is more likely to diffuse strategically. An example of such a setting can be an adversarial recruitment of people in social networks, which is usually not contagious but carefully planned. As we discuss below, the adversarial recruitment of people may occur in either competitive scenarios, such as customer poaching and brand switching [23], or malicious ones, such as recruitment of individuals to join criminal and covert organizations [40, 52].

An example of a competitive scenario is an archetypical Stackelberg game, i.e., the competition between two market players—an incumbent and an entrant [20, 64]. One of the ways in which such an entry can be executed, especially in the time of growing popularity of personalized marketing campaigns [42], is by poaching the customers of the incumbent [14], e.g., by offering them discounts and free samples. As a pre-emptive move, the incumbent may try to secure the loyalty of the customers by compensating them with competitive offers. In this game, the incumbent is the defender, who acts in advance to prevent the customers from switching to the entrant, who is the attacker. Furthermore, given that the decision to switch to new products or services, especially in high-tech markets [26], is often affected by social influence, the consideration of the social network is key to capture the complexity of this scenario [29, 34].

An example of a malicious scenario concerns the recruitment of individuals to join covert organizations. It is well known that various covert organizations recruit members by carefully selecting potential candidates using social ties, and they do this typically by taking advantage of the kinship and friendship ties [40]. In such scenarios, the attacker is a covert organization who intends to strategically select potential candidates from the acquaintances of the already-recruited members. However, the defender is either law enforcement or a charitable institution, whose goal is to prevent the criminal degeneration or radicalization of a local community and who has limited security resources to distribute over the social networks in an attempt to achieve this goal.

Unfortunately, the existing models are inadequate to grasp the complexity of these scenarios. On one hand, the aforementioned defender-attacker literature [1, 5, 41] assumes a stochastic contagion of the attack over the network. On the other hand, the existing models of strategic diffusion [2] do not include an optimizing defender counteracting the spread, i.e., they only model the diffusion aspect of the problem. Furthermore, there exist models of market competition where one of the competitors strategically schedules an order in which members of the network make a decision on adopting either her product or the product of her competitor based on their network neighborhood [4, 15]. However, even though these works study market competition, only one competitor in these models makes strategic decisions, while the other remains passive. Moreover, the temporal aspect is limited to selecting a sequence of nodes, with each node in the sequence making her decision immediately. In reality, more resilient nodes may take more time to be convinced by a competitor, adding another layer of complexity to the problem. Another body of literature considers strategically affecting the spread of diffusion by selecting the exact moment when the seed nodes are activated, instead of activating them all at the beginning of a marketing campaign

process [30, 31]. Nevertheless, after activating the seed nodes the remainder of the process is purely stochastic, whereas we intend to model a fully strategic diffusion process. Also, these settings do not include any competitive elements, whereas the existence of an entity intending to prevent the spread of diffusion is a key element of our approach.

Against this background, we propose and analyze the first Stackelberg game between a defender and an attacker of the network, where the attacker strategically guides the course of network diffusion and a central defender strategically attempts to counter the expected attack in advance. At the beginning of the game, the defender distributes security resources among nodes of the network. Having observed the security efforts of the defender, the attacker chooses the sequence in which the nodes are to be attacked, where the time it takes to successfully attack a certain node is inversely proportional to the number of already captured neighbors of this node. We use complex contagion to model the susceptibility level of each member of the network, as it has been shown to better describe diffusion processes in social networks [11–13]. Within this framework, we investigate two main questions: *how difficult is it for the attacker to find an optimal way of strategically attacking a network?* and *what are the effective ways of defending a network from a fully sequential strategic attack?* Our study mainly focuses on finding the best strategies for both players.

We first prove that in the general case, i.e., for an arbitrary network, the problem of finding an optimal plan of the strategic attack is NP-complete. Given this hardness result, we next analyze whether the same holds when we focus our attention on some simple network structures. In doing so, we follow the studies of the previous models in the literature (with no strategic spread of attack) for which polynomial results were obtained for some simple networks [1, 5, 41], e.g., by considering only a very limited number of nodes [5] or by considering only symmetric networks [1]. The literature shows that in such cases finding the optimal strategy of an attacker is typically relatively easy—she just selects the target offering the highest expected utility. Given this, the remaining analysis was typically focused on analyzing the strategies of the defender. Interestingly, we are still able to find optimal strategies for some some simple network structures in our more challenging model. In particular, for cliques and stars, we develop polynomial algorithms to search for the optimal strategies for both the attacker and the defender.

Next, for arbitrary but small networks, we develop a dynamic programming algorithm that finds an optimal attack more efficiently than checking all possible sequences, albeit still in exponential time. We also show an effective way of protecting the network from a fixed set of attackers' strategies, similar to that proposed by Lou et al. [41].

Our hardness result shows that considering the strategic spread of an attack, rather than a stochastic one, significantly increases the challenge faced by both players. Indeed, in the Stackelberg game, the NP-completeness of finding an optimal attack constitutes a challenge not only to the attacker but also to the defender. This is because it may be prohibitively difficult for the defender to choose her best strategy, given that this choice is based on the prediction of the best response by the attacker.

A popular way of dealing with such computational challenges efficiently, although most probably suboptimally, is to resort to heuristics. Hence, we propose a variety of heuristic strategies for both players. As for the attacker's strategies, we consider the ones found effective in strategic diffusion models [2], as well as strategies based on exploration-exploitation techniques [58]. As for the defender's strategies, the previous studies found that the optimal defender's utility is achieved when security levels of all targets are equal [5, 41]. However, our findings indicate that making all nodes equally difficult to target is not particularly effective against attack that spreads strategically. Instead, in most cases the defense based on one of the network centrality measures provides better results.

2 PRELIMINARIES

Let $G = (V, E)$ be a network, where $V = \{v_1, \ldots, v_n\}$ denotes the set of n nodes and $E \subseteq V \times V$ denotes the set of edges. Let $\mathbb{G}(V)$ be a set of all networks over the set of nodes V. We denote an edge between nodes v_i and v_j by (v_i, v_j). In this work, we consider only *undirected* networks, i.e., networks in which we do not discern between edges (v_i, v_j) and (v_j, v_i). We also assume that networks do not contain self-loops, i.e., $\forall_{v_i \in V} (v_i, v_i) \notin E$. We denote by $N_G(v_i)$ the set of *neighbors* of v_i in G, i.e., $N_G(v_i) = \{v_j \in V : (v_i, v_j) \in E\}$. Furthermore, we denote by $d_G(v_i)$ the *degree* of v_i in G, i.e., $d_G(v_i) = |N_G(v_i)|$. We will often omit the network itself from the notation when it is clear from the context, e.g., by writing $N(v)$ instead of $N_G(v)$.

We will also use a concept of a path in the graph. In particular, a path is a sequence of distinct nodes, $\langle v_l, \ldots, v_k \rangle$, such that every two consecutive nodes are connected by an edge. The length of a path is the number of edges in that path. Let $\Gamma(V)$ denote the set of all *sequences* of elements from V without repetitions. Let γ_i denote the ith element of sequence $\gamma \in \Gamma(V)$. Finally, let $|\gamma|$ denote the number of elements in $\gamma \in \Gamma(V)$.

In graph theory and social network analysis, the importance of nodes is quantified using functions of the form: $c : \mathbb{G}(V) \times V \to \mathbb{R}$, called *centrality measures*. Three such fundamental centrality measures are the *degree* centrality [53], the *closeness* centrality [7], and the *betweenness* centrality [3, 21]. Specifically, given a node $v_i \in V$ and an undirected network, we have that the formula of degree centrality is: $c_{degr}(v_i) = \frac{d(v_i)}{n-1}$. In words, degree centrality ranks the nodes simply according to their number of neighbors. Next, the formula of closeness centrality is:

$$c_{clos}(v_i) = \frac{n-1}{\sum_{v_j \in V} dist(v_i, v_j)},$$

where $dist(v_i, v_j)$ denotes the distance between v_i and v_j. Intuitively, closeness centrality focuses on distances among nodes and gives high value to the nodes that are, on average, close to all other nodes. Furthermore, betweenness centrality is defined as:

$$c_{betw}(v_i) = \frac{2}{(n-1)(n-2)} \sum_{v_j, v_k \in V \setminus \{v_i\}} \frac{|\{p \in sp(v_j, v_k) : v_i \in p\}|}{|sp(v_j, v_k)|},$$

where $sp_G(v_i, v_j)$ denotes the set of all shortest paths between any pair of nodes, v_i and v_j. Thus, in words, betweenness centrality considers all shortest paths between any two nodes in the network. The more such shortest paths a particular node belongs to, the more important it is.

Apart from the above three fundamental centrality measures, many others have been proposed in the literature. One interesting class are *game-theoretic centrality* measures. The key idea behind game-theoretic centrality measures is to treat nodes as players in a cooperative game, where the value of each coalition of nodes is determined by certain graph-theoretic properties [43, 56]. The key advantage of this approach is that nodes are ranked not only according to their individual roles in the network but also according to how they contribute to the role played by all possible subsets (or groups) of nodes. In game theory, the function that assigns a value to any subset of nodes, $u : 2^V \to \mathbb{R}$, is called the *characteristic function*. Now, the best known game-theoretic method to evaluate the role of a player in the game is the Shapley value:

$$\psi(v_i, u) = \sum_{C \subseteq V \setminus \{v_i\}} \frac{|C|!(n - |C| - 1)!}{n!} \left(u(C \cup \{v_i\}) - u(C) \right).$$

The value of $\psi(v_i, u)$ is the Shapley value of player v_i in game u. Now, in each Shapley value–based centrality measure c the centrality score of a node v_i is equal to its Shapley value in the game u

associated with this centrality measure, i.e., $c(v_i) = \psi(v_i, u)$. We consider the following centralities based on the Shapley value:

- The *Shapley value–based degree centrality* [43] of a node is its Shapley value in a game with the characteristic function:

$$u(C) = \left| C \cup \bigcup_{v_i \in C} N(v_i) \right|.$$

  In words, in this game the value of a given coalition is the size of the coalition and its neighborhood.

- The *Shapley value–based closeness centrality* [43] is defined by:[1]

$$u(C) = \left| C \cup \{v_j \in V : \exists_{v_i \in C} dist(v_i, v_j) \leq \delta\} \right|.$$

  In words, in this game the value of a given coalition is the number of its nodes and all nodes in distance at most δ. In this work, we assume $\delta = 3$.

- *Shapley value–based betweenness centrality* [59] is defined by:

$$u(C) = \sum_{v_j, v_k \in V \backslash C} \frac{|\{p \in sp(v_j, v_k) : \exists_{v_i \in C} v_i \in p\}|}{|sp(v_j, v_k)|}.$$

  In words, the value of a given coalition depends on the percentage of the shortest paths between pairs of nodes not from the coalition that the members of the coalition belong to.

3 THE MODEL OF SECURITY DIFFUSION GAMES

We define a security diffusion game by combining ideas of security games and strategic diffusion in networks.

Definition 1 (Security Diffusion Game). The security diffusion game is a Stackelberg game between two players, the defender (the leader) and the attacker (the follower). It is defined by a tuple (G, Φ, T), where $G = (V, E)$ is a given network, $\Phi \in \mathbb{R}^+$ is the amount of security resources available to the defender, and $T \in \mathbb{R}^+$ is the attacker's time limit before the defender discovers the attack and stops it completely. The game has two steps:

- Step 1—at the beginning of the game, the defender distributes security resources among nodes of the network, i.e., she chooses $\phi : V \rightarrow \mathbb{R}$ such that $\forall_{v \in V} \phi(v) \geq 0$ and $\sum_{v \in V} \phi(v) \leq \Phi$.
- Step 2— next, having observed the security efforts of the defender in Step 1, the attacker chooses the sequence, $\gamma \in \Gamma(V)$, in which nodes will be targeted for attacks. A given node in the sequence can be attacked only when all preceding nodes have been successfully captured.

The activation time of each node v, denoted $\tau(v)$, is a function of the network topology, the set of already activated nodes, and the distribution of the security resources (to improve readability, we omit the additional arguments in the notation). The utilities of the attacker and the defender are denoted u_A and u_D, respectively, and are functions of the network, the distribution of security resources, the sequence of the attack, and the time limit T:

$$u_D = f^D(G, \phi, \gamma, T);$$
$$u_A = f^A(G, \phi, \gamma, T).$$

---

[1]Note that there are many other ways in which one could define the game-theoretic extension of closeness centrality [61].

In this work, we make the following assumptions about the particular forms of the utility functions and the activation time function. As for the utility functions, we assume that $u_A = \eta(\gamma, T)$, where $\eta(\gamma, T)$ denotes the number of nodes activated within time limit T when using attack sequence γ (to simplify notation, we assume that the network and the distribution of security resources are known from the context). Furthermore, to express the competitive nature of the game, we define the utility of the defender as $u_D = -u_A = -\eta(\gamma, T)$, which makes the security diffusion game a zero-sum game.

We define the activation time of a node as a function of its assigned security resources, its degree, and the number of the activated nodes among its neighbors:[2]

$$\tau(v) = \begin{cases} d(v) + \phi(v) & \text{if v is the seed node, i.e., the first node in the sequence } \gamma \quad (1) \\ \dfrac{d(v) + \phi(v)}{|N(v) \cap I|} & \text{otherwise,} \quad (2) \end{cases}$$

where I is the set of currently activated nodes. The above form of activation time function combines notions from strategic diffusion [2] and from security games [46]. In particular:

- The more security resources assigned to v, the more time it takes to activate this node (see the term $\phi(v)$ in Equation (1) for the seed node and the nominator of Equation (2) for all the other nodes);
- The time of activation decreases with the number of activated neighbors (see the term $|N(v) \cap I|$ in the denominator of Equation (2)). In other words, the more neighbors of v have been already activated, the more this node is susceptible to become activated itself.

Furthermore, the activation time increases with the total degree of the node (see the term $\phi(v)$ in Equation (1) for the seed node and the nominator of Equation (2) for all the other nodes). This is a well established idea in the diffusion models [2, 35] and complex contagion [13] literature and it has the following intuitive interpretation. If the number of already activated nodes around v makes only a small proportion of the total number of v's neighbors, then v is less likely to become active itself.

We also note that the activation time function (Equations (1–2)) implies that, while we allow the attacker to choose any node in the network as the seed node, any other node needs at least one activated neighbor to become activated itself. Hence, the strategy of the attacker (i.e., the sequence of nodes that she chooses to attack) constitutes by definition a path in the graph.

Example 1. Figure 1 presents a sample network of $n = 9$ nodes and a possible sequence, $\gamma = (v_8, v_6, v_3, v_4, v_7)$, in which the attacker captures the nodes within the time limit $T \geq 26$, i.e.:

- Figure 1(a) depicts a not-yet-attacked network at time $t = 0$, where the number inside a node shows security resources assigned to this node by the defender;
- Figure 1(b) depicts the moment at which the attacker, having chosen to attack node v_8, captures it (at time $t = d(v_8) + \phi(v_8) = 3 + 1 = 4$);
- Figure 1(c) depicts the activation of v_6 at time $t = 4 + (d(v_6) + \phi(v_6)) / (|N(v_6) \cap I|) = 4 + (5 + 3)/1 = 12$);
- Figure 1(d) depicts the activation of v_3 at time $t = 12 + (d(v_3) + \phi(v_3)) / (|N(v_3) \cap I|) = 12 + (1 + 4)/1 = 17$);
- Figure 1(e) depicts the activation of v_4 at time $t = 17 + (d(v_4) + \phi(v_4)) / (|N(v_4) \cap I|) = 17 + (4 + 3)/1 = 24$); and

---

[2]The function in Equation (2) is a form of the Tullock (contest success) function, which is widely used in the security games literature [1, 5, 41] and has various desirable properties [5, 55]. Furthermore, we note that in cyber-security security domain a similar idea is expressed by the Byzantine generals problem [38].
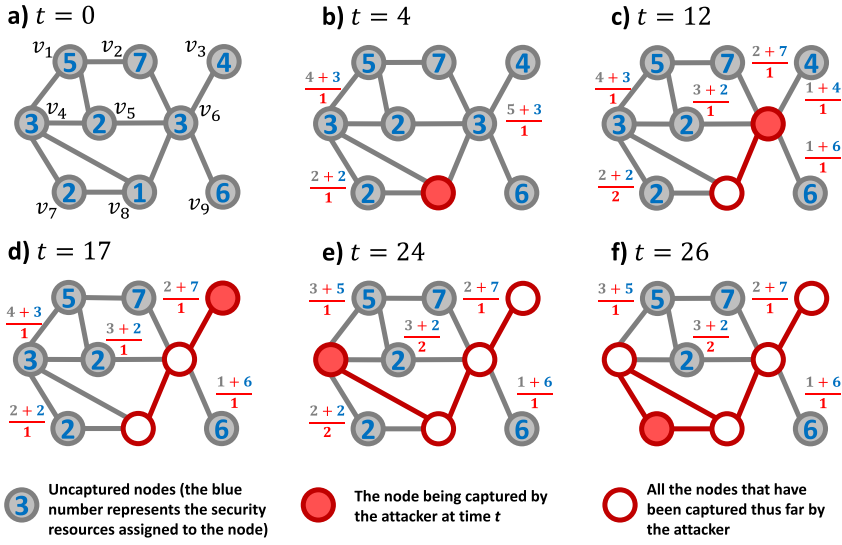
Fig. 1. Figure (a) depicts a sample network with the set of nodes $\{v_1, \ldots, v_9\}$. Figures (b)–(f) depict sequence $\gamma = (v_8, v_6, v_3, v_4, v_7)$ in which the attacker chooses to spread the attack. Note that, for the sake of clarity, we omit the labels of nodes in Figures (b)–(f).

- Figure 1(f) depicts the activation of v_7 at time $t = 24 + (d(v_7) + \phi(v_7)) / (|N(v_7) \cap I|) = 24 + (2 + 2)/2 = 26$).

We can also relate our model to both motivating scenarios presented in the introduction. Our sample competitive scenario involves two companies—an incumbent and an entrant. The former one is the defender who can invest resources into securing higher levels of loyalty of its customer base. Conversely, the entrant is the attacker whose goal is to attract as many customers of the defender as possible. The activation time for each customer (node) represents here the difficulty for the entrant of doing so. As per Equations (1–2), in our model this difficulty for any customer v is assumed to be in the inverse proportion to the share of the already attracted (activated) customers in the neighborhood of v. As we already argued, this is a well-established approach in the complex contagion literature [13]. In our context, such a definition of the activation time reflects that in many markets (especially high-tech ones) the customers are more likely to switch to a competing company (or a technology) if more of their friends already use it. We also note that the strategy of the attacker (the sequence of nodes) induces a connected subgraph in the network; hence, our model fits the settings of word-of-mouth marketing [36].

Our sample malicious scenario involves either a criminal or extremist organization who selects the candidates for new members using social ties of their existing rank-and-file. Here, the defender is either the law enforcement or a charitable institution whose goal is to protect a community. The activation time for each node represents the threshold of an individual to become attracted by the covert organization. The meaning of Equations (1–2) in this context is that any individual v is assumed to be more attracted to become a member of a covert organization if relatively many of her neighbors have already done so. This reflects many studies of illegal behaviors in the literature—for instance, Bikhchandani et al. [8, p. 994] "consider a teenager deciding whether or not to try drugs. A strong motivation for trying out drugs is the fact that friends are doing so. Conversely, seeing friends reject drugs could help persuade the teenager to stay clean." Furthermore, we note that the fact that the strategy of the attacker induces a connected subgraph is natural in this scenario, as it ensures that the covert network remains connected.

4 FINDING OPTIMAL STRATEGIES

We begin our analysis of the model by considering the problem of finding optimal strategies. Since the Stackelberg game is solved by backward induction, we first consider the strategy of the attacker (given an arbitrary strategy of the defender). In particular, we formally define the Optimal Attack Problem as follows:

Definition 2 (Optimal Attack Problem). This problem is defined by a tuple (G, ϕ, v_S, T), where $G = (V, E)$ is a given network, $\phi : V \rightarrow \mathbb{R}$ is the defender's distribution of security resources, $v_S \in V$ is the seed node of the attacker, and T is the attacker's time limit. The goal is to identify $\gamma^* \in \Gamma(V)$ such that γ^* is in $\arg\max_{\gamma \in \Gamma(V):\gamma_1=v_S} \eta(\gamma, T)$.[3]

As mentioned in the introduction, in the existing literature, where the attack spreads in the form of a stochastic process, the attacker's strategy consists only of choosing a seed node; hence, it is usually easy to pick an optimal strategy of the attacker. However, finding the optimal strategy of the attacker in our model is no longer trivial. Indeed, as we show in Section 4.1, in the general case, i.e., when we allow a network to have any structure, the problem of finding a sequence of nodes providing the largest number of nodes activated within a time limit is NP-complete. Hence, even if the attacker is given complete information about the activities of the defender, finding an optimal strategy of the attack is computationally intractable.

However, we are able to find optimal strategies of both attacker and defender for more restricted network structures. In particular, we find that, if the network under consideration is a star, the attacker should attack nodes in non-decreasing order of their activation times, with the additional constraint that the center of the star has to be one of the first two nodes in the sequence (see Section 4.2). As for the defender strategy, if it is possible not to let the attacker activate even a single node given the time limit, then the defender should try to make the time of activation equal for all nodes (if possible) or all peripheral nodes (if there are not enough security resources). Alternatively, the defender should assign all security resources to the central node (since, again, it has to be one of the first two nodes in the sequence). We also analyze the optimal strategies for cliques, where the optimal attack sequence orders the nodes non-decreasingly according to assigned security resources, while the defender should spread her security resources uniformly (see Appendix A). We can also efficiently find an optimal attack sequence for a tree (see Appendix B).

In the case we need to find an optimal attacker's strategy for an arbitrary (but small) network, we present an algorithm based on dynamic programming technique that returns solution in time $O(2^n n^2)$, where n is the number of nodes in the network. Its pseudocode can be found as Algorithm 2 in Appendix C. While it is more effective than simple exhaustive search, its exponential time complexity makes it suitable for rather small networks.

Finally, we also present a mixed-integer-linear-programming method to compute optimal defense strategies when the strategies of the attacker can be enumerated, i.e., when the set of the attacker's strategies is of reasonable size. The formulation has $O(|\hat{\Gamma}|n)$ constraints, where $\hat{\Gamma}$ is the set of strategies available to the attacker (see Section 4.3). While this method would be highly impractical if we assume that the attacker can use any strategy, it can help defend the network effectively when some additional constraints are put on the attacker, e.g., we can ensure that some ways of attack are completely implausible.

4.1 Complexity Analysis of the Optimal Attack Problem

The following result holds:

---

[3]Recall that $\eta(\gamma, T)$ denotes the number of nodes activated within time limit T when using attack sequence γ.
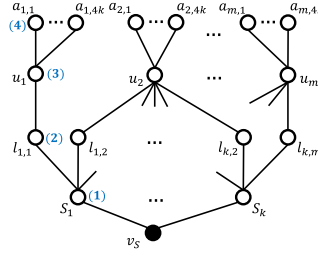
Fig. 2. The network used to reduce the 3-Set Cover problem to the Optimal Attack problem. The seed node, v_S, is marked black. Blue numbers in the parentheses show the order of nodes that have to be activated before activating node $a_{1,1}$.

THEOREM 2. *The Optimal Attack problem is NP-complete.*

PROOF. The decision version of the optimization problem is the following: Given a network $G = (V, E)$, a distribution of security resources ϕ, a seed node v_S, a time limit T, and a value $r^* \in \mathbb{N}$ expressing the number of nodes to be activated within time limit, does there exist a sequence of nodes $\gamma^* \in \Gamma(V)$ such that $\eta(\gamma^*, T) \geq r^*$?

This problem clearly is in NP. This is because, given a solution, i.e., a sequence $\gamma^* \in \Gamma(V)$, we can compute the time of activation of every node in the sequence and verify in polynomial time whether $\eta(\gamma^*, T) \geq r^*$.

To prove NP-hardness, we will show a reduction from the NP-complete 3-Set Cover problem. To this end, we will build a network that reflects the structure of a given 3-Set Cover problem instance and use it as an input for the Optimal Attack problem. Finally, we will show that an optimal solution of the Optimal Attack problem corresponds to a solution of the given instance of the 3-Set Cover problem.

More formally, an instance of the NP-complete 3-Set Cover problem is defined by a universe $U = \{u_1, \ldots, u_m\}$, a collection of sets $S = \{S_1, \ldots, S_k\}$ such that $\forall_j S_j \subset U$ and $\forall_j |S_j| = 3$, and an integer $b \leq k$. The goal is to determine whether there exist b elements of S, the union of which equals U. In what follows, let $\sigma(u_i) = |\{S_j \in S : u_i \in S_j\}|$, i.e., $\sigma(u_i)$ is the number of sets in S that contain u_i.

First, let us create a network G that reflects the structure of the instance of the 3-Set Cover problem under consideration. It is shown in Figure 2, where:

- **The set of nodes:** For every $S_i \in S$, we create a single node, denoted by S_i. For every $u_i \in U$, we create a single node, denoted by u_i, as well as $4k$ nodes $a_{i,1}, \ldots, a_{i,k}$. Moreover, for every $u_j \in U$ and every $S_i \in S$ such that $u_j \in S_i$, we create a single node $l_{i,j}$. Additionally, we create a single node v_S.
- **The set of edges:** For every node $S_i \in S$, we create an edge (S_i, v_S). For every node $a_{i,j}$, we create an edge $(a_{i,j}, u_i)$. Finally, for every node $l_{i,j}$, we create two edges, $(S_i, l_{i,j})$ and $(u_j, l_{i,j})$.

Next, let ϕ^* be a distribution of security resources such that:

- $\phi^*(v_S) = 0$,
- $\phi^*(a_{i,j}) = 0$ for every node $a_{i,j}$,
- $\phi^*(u_i) = k - \sigma(u_i)$ for every node u_i,
- $\phi^*(S_i) = 20mk$ for every node S_i,
- $\phi^*(l_{i,j}) = 20mk + 2$ for every node $l_{i,j}$.

This particular distribution of security resources has the following properties:

- it gives each node u_i the same time of activation (assuming an equal number of activated neighbors), and
- it gives each of the nodes S_i and $l_{i,j}$ the minimal time of activation that is greater than the maximal time of activation of nodes u_i, i.e., activating node S_i or $l_{i,j}$ always takes more time than activating node u_i, no matter what the number of active neighbors is.

These properties will play an important role later in the proof. Furthermore, let $T^* = (b + m)(20mk + 4) + 9mk + 1$ and $r^* = b + (4k + 2)m$. These are the time and the number of activated nodes of the strategy that corresponds to a solution to the 3-Set Cover problem.

Now, consider the instance of the Optimal Attack problem in the form of $(G, \phi^*, v_S, T^*, r^*)$. We will show next that an optimal solution to this instance corresponds to an optimal solution to the 3-Set Cover problem. We begin by observing that the following holds for the times of activation of the nodes in V:

- $\tau(a_{i,j}) = 1$ for every node $a_{i,j}$, as we have $d(a_{i,j}) + \phi^*(a_{i,j}) = 1$ and $|N(a_{i,j}) \cap I| \leq 1$;
- $\tau(u_i) \leq 5k$ for every $u_i \in U$, as we have $d(u_i) + \phi^*(u_i) = 5k$ and $|N(u_i) \cap I| \geq 1$;
- $5mk + 1 \leq \tau(S_i) \leq 20mk + 4$ for every $S_i \in S$, as we have $d(S_i) + \phi^*(S_i) = 20mk + 4$ and $1 \leq |N(S_i) \cap I| \leq 4$;
- $10mk + 2 \leq \tau(l_{i,j}) \leq 20mk + 4$ for every node $l_{i,j}$, as we have $d(l_{i,j}) + \phi^*(l_{i,j}) = 20mk + 4$ and $1 \leq |N(l_{i,j}) \cap I| \leq 2$.

We will now show that if there exists a solution $S^* \subset S$ to the given instance of the 3-Set Cover problem, then there exists a solution γ^* to the constructed instance of the Optimal Attack problem such that $\eta(\gamma^*, T^*) = r^* = b + (4k + 2)m$. Indeed, we can construct such solution by activating every node $S_i \in S^*$ (b nodes activated in time $b(20mk + 4)$), choosing for every $u_j \in U$ node $S_i \in S^*$ such that $u_j \in S_i$ and activating nodes $l_{i,j}$ and u_j ($2m$ nodes activated in time $(20mk + 4)m + 5mk$), and finally activating all nodes $a_{i,j}$ ($4mk$ nodes activated in time $4mk$).

In what follows, let γ_T^* denote the maximum prefix of γ^* such that $\tau(\gamma_T^*) < T^*$. Now, we have to show that if there exists a solution γ^* to the constructed instance of the Optimal Attack problem, then there exists also a solution $S^* \subset S$ to the given instance of the 3-Set Cover problem. To this end, we will show that for the prefix γ_T^* of any such solution γ^* it must be that, for every $u_j \in U$, there exists a node $S_i \in \gamma_T^*$ such that $u_j \in S_i$ and $|\gamma_T^* \cap S| \leq b$. Notice that when this is the case, we can obtain the solution to the given instance of the 3-Set Cover problem by taking $S^* = \gamma_T^* \cap S$.

First, we observe that any sequence γ_T^* that does not contain all nodes u_i cannot have the required number of r^* activated nodes within the time limit. Assume, to the contrary, that there exists such a sequence. Since there exists node u_i that is not activated, neither of the nodes $a_{i,j}$ are activated. Therefore, this sequence has to activate $4k + 1$ of the nodes S_i or $l_{i,j}$ instead. However, there are only $4k$ nodes S_i and $l_{i,j}$. Hence, sequence γ_T^*, which is a solution of the constructed instance of the Optimal Attack problem, must activate all nodes u_j. To activate given node u_j, we need to activate at least one node $l_{i,j}$, and to achieve that, we need to activate node S_i. Therefore, because of the way we constructed the network, for every $u_j \in U$ there exists a node $S_i \in \gamma_T^*$ such that $u_j \in S_i$.

Now, we need to show that for γ_T^* being a solution of the constructed instance of the Optimal Attack problem, we have $|\gamma_T^* \cap S| \leq b$, i.e., that any sequence being a solution cannot activate more than b nodes S_i within time limit. As shown above, γ_T^* has to activate all u_j nodes. Notice that to activate u_j, we have to activate a node $l_{i,j}$ with only one activated neighbor (its only other neighbor being u_j). Activating m such nodes takes $m(20mk + 4)$ time. Hence, to activate more than b nodes S_i within time limit, we would have to do it in time shorter (as we still have to activate nodes u_j) than

$T^* - m(20mk + 4) = b(20mk + 4) + 9mk$. Now, activating node S_i (including the time necessary to activate neighboring nodes $l_{i,j}$) takes time:

- $20mk + 4$ when activating with only one active neighbor v_S;
- $\frac{20mk+4}{2} + (20mk + 4)$ when activating with two active neighbors;
- $\frac{20mk+4}{3} + 2(20mk + 4)$ when activating with three active neighbors;
- $\frac{20mk+4}{4} + 3(20mk + 4)$ when activating with four active neighbors.

This is because, to be used to speed up the activation of node S_i, node $l_{i,j}$ has to be activated beforehand, in time $20mk + 4$. Hence, activating a single node S_i takes at least $20mk + 4$ time and it is not possible to activate more than b nodes S_i in time $b(20mk + 4) + 9mk$.

This implies that the optimal solution to the constructed instance of the Optimal Attack problem must correspond to the optimal solution to the given instance of the 3-Set Cover problem, thus concluding the proof. □

4.2 Optimal Strategies for Stars

For stars, we have the following result:

THEOREM 3. *Let network $G = (V, E)$ be a star with center a and peripheral nodes $b_1, \ldots, b_{n-1}$. Let ϕ be a particular distribution of security resources (assume that nodes b_i are ordered non-decreasingly according to $\phi(b_i)$). An optimal attack sequence is then:*

- *$\gamma^* = \langle a, b_1, b_2, \ldots, b_{n-1} \rangle$ if $\phi(b_1) \geq \phi(a) + n - 2$ (i.e., when a has the lowest time of activation),*
- *$\gamma^* = \langle b_1, a, b_2, \ldots, b_{n-1} \rangle$ otherwise.*

An optimal defense strategy against an optimal attack strategy is then:

- *$\phi^*(a) = \frac{\Phi - (n-1)(n-2)}{n}$ and $\phi^*(b_i) = \frac{\Phi + n - 2}{n}$ if $T \leq \frac{\Phi + 2(n-1)}{n}$,*
- *$\phi^*(a) = 0$ and $\phi^*(b_i) = \frac{\Phi}{n-1}$ if $\frac{\Phi + 2(n-1)}{n} < T \leq \frac{\Phi + n - 1}{n-1}$,*
- *$\phi^*(a) = \Phi$ and $\phi^*(b_i) = 0$ otherwise.*

In words, an optimal strategy of the attacker is to attack nodes in non-decreasing order of their activation times, with the additional constraint that the center of the star has to be one of the first two nodes in the sequence. As for the defender strategy, if it is possible not to let the attacker activate even a single node given the time limit, then the defender should try to make the time of activation equal for all nodes (if possible) or all peripheral nodes (if there are not enough security resources). Alternatively, the defender should assign all security resources to the central node (since, again, it has to be one of the first two nodes in the sequence).

PROOF. First, we prove our claim about the strategy of the attacker. Notice that node a has to be either first or second node in the attack sequence, as the attacker either starts with it or it is the only neighbor of the starting node (if the attacker starts with one of the peripheral nodes). Hence, it has activation time of $\tau(a) = n - 1 + \phi(a)$. Similarly, any node b_i is either first in the sequence or is activated when exactly one of its neighbors (central node a) is active. Hence, its activation time is $\tau(b_i) = 1 + \phi(b_i)$. If it is impossible to activate more than one node within time limit, then the sequence γ^* is optimal, because the first node has the lowest time of activation in the entire network.

Now consider a case where it is possible to activate at least two nodes within the time limit. Assume, to the contrary, that in an optimal attack sequence, γ, we have b_j activated before b_i for $i < j$ (as mentioned above, node a has to be either on the first or the second position in the sequence and we can swap these nodes without changing the total activation time). However, such sequence

can be improved by swapping elements b_i and b_j. The argument follows the same logic as in the proof of Theorem 2.

Now, we move to proving our claim about the strategy of the defender. First, consider a case in which it is possible not to let attacker activate even a single node, i.e., $\forall_{v \in V} \tau(v) \geq T$. It is then optimal for the defender to have the same activation time for all nodes (as the attacker will pick the one with lowest activation time as the first in sequence). Hence, we have $\tau(a) = \phi^*(a) + n - 1 = \phi^*(b_i) + 1 = \tau(b_i)$ and $\phi^*(a) + (n-1)\phi^*(b_i) = \Phi$. After solving this set of equations, we get $\phi^*(a) = \frac{\Phi - (n-1)(n-2)}{n}$ and $\phi^*(b_i) = \frac{\Phi + n - 2}{n}$. We then have $\tau(a) = \tau(b_i) = \frac{\Phi + 2(n-1)}{n}$. Since we consider a case in which it is possible not to let the attacker activate even a single node, we need to have $\tau(a) = \tau(b_i) = \frac{\Phi + 2(n-1)}{n} \geq T$.

Notice that it is possible that the defender does not have enough defense resources to make time of activation of all nodes equal (when $\Phi < (n-1)(n-2)$). She should then spread them uniformly among peripheral nodes to maximize the minimal time of activation in the network. We then have $\phi^*(a) = 0$ and $\phi^*(b_i) = \frac{\Phi}{n-1}$. Minimal time of activation in the network is then $\tau(b_i) = \frac{\Phi + n - 1}{n-1}$ and for this strategy to be optimal, we need to have $\tau(b_i) = \frac{\Phi + n - 1}{n-1} \geq T$.

Now, consider the case in which the attacker is able to activate $k \geq 1$ nodes. One of the first two nodes in the sequence has to be a. Hence, assigning all security resources to a either maximizes the time required to activate available number of nodes—if $k \geq 2$ as their total activation time is $\phi(a) + \sum_{i=1}^{k-1} \phi(b_i)$), or it maximizes the time that would be necessary to activate second node in the sequence—if $k = 1$. □

4.3 Optimal Defense against a Subset of Strategies

In this section, we present a mixed-integer linear programming method to compute optimal defense strategies when the strategies of the attacker can be enumerated, i.e., when the set of the attacker's strategies is of reasonable size.

Assume that the attacker has at her disposal only a limited set of strategies, $\hat{\Gamma} \subseteq \Gamma(V)$. We assume that each sequence in this set is a correct strategy of the attack, i.e., there is no attempt to activate nodes with no active neighbors. In such case the problem of finding the optimal distribution of defense resources can be formulated as a mixed-integer linear programming problem. The formulation is:

$$\max_{\phi_v, a_{\gamma, i}, k} k$$
$$\text{subject to} \quad \phi_v \geq 0 \qquad \forall_{v \in V}$$
$$\sum_{v \in V} \phi_v \leq \Phi$$
$$a_{\gamma, i} \in \{0, 1\} \qquad \forall_{\gamma \in \hat{\Gamma}} \forall_{i \in I}$$
$$\sum_{j \leq i} \frac{d(\gamma_j) + \phi_{\gamma_j}}{c_{\gamma, j}} \geq a_{\gamma, i} T \qquad \forall_{\gamma \in \hat{\Gamma}} \forall_{i \in I}$$
$$\sum_{i \in I} a_{\gamma, i} \geq k \qquad \forall_{\gamma \in \hat{\Gamma}},$$

where $I = \{1, \ldots, n\}$ and $c_{\gamma, j}$ is the number of active neighbors of γ_j at the moment of its activation, when using attack sequence γ (we set $c_{\gamma, 1} = 1$ for every $\gamma \in \hat{\Gamma}$).

The first two sets of constraints guarantee that we have a valid distribution of security resources. When the attacker is using attack sequence γ, we want variable $a_{\gamma, i}$ to equal 1 if and only if node γ_i is not activated within the time limit. This is guaranteed with the fourth set of constraints. In particular, we have $a_{\gamma, i} = 1$ only when the total time of activation of the first i nodes in the sequence (the left-hand side of the fourth constraint) exceeds the time limit, T. The last set of constraints guarantees that k is the minimum number of inactive nodes over all the choices of γ (the attacker minimizes this value, while the defender maximizes it). The formulation has $O(|\hat{\Gamma}|n)$ constraints.

Table 1. Formulas for the Defender's Heuristics

Equality	$\phi(v_i) = \max(0, s^* - d(v_i))$
Uniform	$\phi(v_i) = \frac{1}{n}\Phi$
Random	$\phi(v_i) = \frac{r_i}{\sum_{i=1}^{n} r_i}\Phi$
High Centrality c	$\phi(v_i) = \frac{c(v_i)}{\sum_{i=1}^{n} c(v_i)}\Phi$
Low Centrality c	$\phi(v_i) = \frac{c^{-1}(v_i)}{\sum_{i=1}^{n} c^{-1}(v_i)}\Phi$

In the next section, we explore the efficiency of various heuristic strategies that may be used by both players.

5 ATTACK AND DEFENSE STRATEGIES BASED ON HEURISTICS

Another approach that may be pursued by both the defender and the attacker is to resort to heuristic strategies. In fact, one can think of many potential heuristics, depending on the network topology and the particular setting at hand. Hence, in this section, we propose a number of different heuristics that are inspired by a variety of scenarios and the results from the literature.

5.1 Heuristic Strategies of the Defender

To find a suitable heuristic, we should take into account the information possessed by the players. In particular, the defender does not know *a priori* where the attack is going to take place. Hence, the heuristic strategy of assigning available security resources should be based on some properties of either the security game or the network, as no other information is available at this stage. The formulas of the defender's strategies are presented in Table 1. In these formulas, r_i is drawn uniformly at random from $[0, 1]$, s^* is the maximal s such that $\sum_{i=1}^{n} \max(0, s - d(v_i)) \leq \Phi$, and c denotes a given centrality measure for centrality-based strategies.

We now present an intuition behind each heuristic for the defender:

- *Equality*—the defender tries to make the activation time equal for as many nodes as possible (as it may not be possible for all nodes). This heuristic is inspired by the work of Bachrach et al. [5], who found that in their model the optimum is achieved when security levels are equal for all targets:
- *Uniform*—the security resources are distributed uniformly among all nodes. This heuristic is related to the previous one, but now the resources are simply divided equally.
- *Random*—the security resources are distributed randomly among all nodes. This heuristic is a natural benchmark for other ones.

Next, we propose 12 heuristics for the defender that are based on the topology of the network. The basic idea is to assign available security resources either proportionally or inverse proportionally to the centrality, i.e., the importance, of nodes in the network. The following set of heuristics for the defender is based on the centrality measures:

- *High Degree*—the defender focuses on defending nodes with high degrees. The inspiration for this strategy can be found in a number of works that indicate that hubs are of key importance in the diffusion process [16, 47].
- *Low Degree*—the defender focuses on defending the nodes with low degrees. This strategy is inspired by the observation that some criminal groups tend to recruit lonely, socially isolated people [10].

- *High Betweenness*—the defender focuses on defending the nodes with high betweenness. Betweenness centrality is considered particularly important in the context of financial networks, where it is arguably the most sensible measure of the systemic danger of an individual bank within the financial system [22]. In particular, banks with high betweenness centrality are able to bring down the entire financial system if consecutive occurrences of illiquidity materialize, given that there are no interventions by relevant authorities.

The next two strategies are inspired by the observation that nodes with low betweenness centrality and (typically) high closeness centrality play the role of the "pulse-takers" in their organizations [27]. They are easily accessible by other central nodes as well as to the rest of the network. The pulse-takers are key to the preservation and the development of companies [57] and as such are more likely to be a subject of various attacks, such as an advanced ransomware attack [63].

- *Low Betweenness*—the defender focuses on defending the nodes with low betweenness.
- *High Closeness*—the defender focuses on defending the nodes with high closeness.

We also consider the heuristic strategy focused on the nodes with low closeness. In the social network context, such nodes tend to be vulnerable (easier to target) as more isolated and dependent on information from few neighbors [50].

- *Low Closeness*—the defender focuses on defending the nodes with low closeness.

Finally, we consider heuristics based on the Shapley value of each node (see Section 2 for more details). As the Shapley value expresses synergy between nodes, it might prove effective in choosing which areas of the network to defend more intensively. The Shapley value of a given game can be used to define centrality measure, where the centrality score of each node is the payoff assigned to it by the Shapley value. In this article, we consider the following heuristics based on the Shapley value:

- *High/Low Shapley Value Degree*—the defender focuses on defending the nodes with high/low payoff in a game where the value of the coalition C is the number of nodes in C and those immediately reachable from C [43].
- *High/Low Shapley Value Closeness*—the defender focuses on defending the nodes with high/low payoff in a game where the value of the coalition C is the number of nodes in C and those not further than δ hops away (in our experiments, we use $\delta = 3$) [43].
- *High/Low Shapley Value Betweenness*—the defender focuses on defending the nodes with high/low payoff in a game where the value of the coalition C is the percentage of the shortest paths between the nodes from outside C controlled by the nodes in C [59].

5.2 Heuristic Strategies of the Attacker

Unlike the defender, who has to choose her moves *a priori*, the attacker knows at each step of her decision-making process which nodes in the network have been already activated and which have not, as well as she knows the distribution of security resources chosen by the defender. This immediately suggests the following two heuristics in which the attacker may use this information to her advantage:

- *Greedy*—the heuristic in which the next target is a not-yet-activated node with the lowest time of activation; and
- *Majority*—the heuristic in which the next target is a not-yet-activated node with the highest number of activated neighbors.

In comparison to the Greedy heuristic, the Majority heuristic has a tendency to activate nodes with high degrees earlier in the process. One can also mix both heuristics:

- *Mixed*—a mix of the two previous heuristics that we will denote by Mixed(p), where p is the probability of using the Greedy heuristic and $1 - p$ is the probability of using the Majority heuristic.

The above three heuristics were already studied in strategic network diffusion [2], however, not in the setting that involves a strategic defender or another player.

A parallel can be drawn between using the Greedy and Majority heuristics and the tradeoff between *exploration* and *exploitation* [33]. In particular, the Greedy strategy can be considered the exploitation component of the setting, as it maximizes short-term gains. At the same time, the Majority heuristic has a tendency to target high-degree nodes, thus exploring new potential venues of an attack for the future, although the immediate cost of such an attack is usually higher than for the Greedy heuristic. Following this trail, we propose yet another two heuristics for the attacker that are inspired by the exploration-exploitation literature [58]:

- *Epsilon-Decreasing*—a heuristic that exhibits exploratory behavior at the beginning and exploitative behavior at the end of the process. In more detail, we set the probability of selecting the target using the Greedy heuristic to $p = \frac{t}{T}$, where t is the time elapsed since the beginning of the attack process and T is the total time limit. Otherwise, we select the target using the Majority heuristic.
- *Epsilon-First*—a heuristic that at first exhibits only exploratory behavior, followed by only exploitative behavior. In more detail, we select the target using the Majority heuristic if $t \le \epsilon T$, where ϵ is the parameter of the heuristics, while t and T are defined as for the Epsilon-Decreasing heuristic. Otherwise, we select the target using the Greedy heuristic.

In the next section, we will evaluate all the above heuristic strategies by the means of numerical simulations.

6 EXPERIMENTAL ANALYSIS

We will now test the heuristics proposed in the previous section across different combinations of networks and strategies.

6.1 Experiment Design

The setting of each of our experiments is characterized by (a) a network, G, (b) a strategy of the defender, ϕ, and (c) a strategy of the attacker, γ. As for (a), we use randomly generated networks, as well as real-life datasets. As for (b)—i.e., the defender's strategy—we consider all the defense heuristics proposed in the previous section. In our experiments, we assume that the amount of security resources is $\Phi = 10n$ and the time limit is $T = n$. These parameters' values allow us to clearly present trends appearing for different settings of our experiments. As for (c)—i.e., the attacker's strategy—we again consider all attack heuristics proposed in the previous section. We benchmark them against the optimal attack strategy, which we find using the dynamic-programming algorithm. However, the dynamic-programming algorithm is time- and memory-intensive, and we are able to apply it in practice only for smaller networks. Furthermore, following the usual assumption in the security games literature [1, 5, 41], we choose the seed node of the attacker that provides the highest number of activated nodes (breaking ties uniformly at random).

Network Generation Models. We use the following models to generate the networks:

- *Preferential attachment* networks generated using the Barabási-Albert model [6]. Unless stated otherwise, in our experiments, we add $d = 3$ links with each new node, and we set the size of the initial clique to d;
- *Scale-free* networks generated using the configuration model [44]. Unless stated otherwise, in our experiments, we assume the minimal degree to be $d_{min} = 2$, the maximal degree to be $d_{max} = 10$, and the configuration model parameter to be $\lambda = 3$;
- *Random graphs* generated using the Erdős-Rényi model [19]. Unless stated otherwise, in our experiments, we assume the expected average degree to be $d = 10$;
- *Small world* networks generated using the Watts-Strogatz model [65]. Unless stated otherwise, in our experiments, we assume the expected average degree to be $d = 10$ and the probability of rewiring to be $p = \frac{1}{4}$;
- *Random trees* generated using Prüfer sequences [49]. We use sequences where each element is chosen uniformly at random from set $1, \ldots, n$.

Real-life Networks. In addition to the experiments on randomly generated networks, we also perform simulations for a number of real-life network structures. In particular:

- *Karate Club* [66]—the network of people attending karate classes in one of the American universities. The network consists of 34 nodes and 78 edges.
- *Facebook (small)* [39]—a small fragment of the social network of the users of Facebook. The network consists of 61 nodes and 272 edges.
- *Facebook (medium)* [39]—a medium fragment of the social network of the users of Facebook. The network consists of 333 nodes and 2,523 edges.
- *Facebook (large)* [39]—a large fragment of the social network of the users of Facebook. The network consists of 786 nodes and 14,027 edges.

The four datasets can be considered examples of networks appearing in different applications of our model (they are standard social networks generated based on surveys and social media data), since all nodes are inactive at the beginning.

6.2 Numerical Results

We now move to describing the results of our simulations. Results for randomly generated networks with any combination of attacker's and defender's strategies are presented in Figures 3 and 4, whereas Figure 5 presents results under the assumption that the attacker chooses the optimal strategy (in the case of smaller networks) or the best of considered heuristic strategies. Results for real-life networks are presented in Figures 6 and 7. We describe results of each type of network separately.

Small networks. We will first describe results for networks with 80 nodes. We generate networks of this size to be able to compare the performance of heuristic solutions against the optimal strategy of the attacker.

Interestingly, the Uniform strategy is a relatively effective way of defending the network. At the same time, the Equality strategy usually gets considerably worse results than the Uniform strategy. This seems to be the case, because, while the Equality strategy aims to compensate differences in the time of activation between nodes with different degrees, the Uniform strategy keeps hubs harder to activate than other nodes, discouraging the attacker from targeting them.

Out of the betweenness centrality-based defender strategies, the Low Betweenness strategy is always more effective than its High Betweenness counterpart. The efficiency of the Low
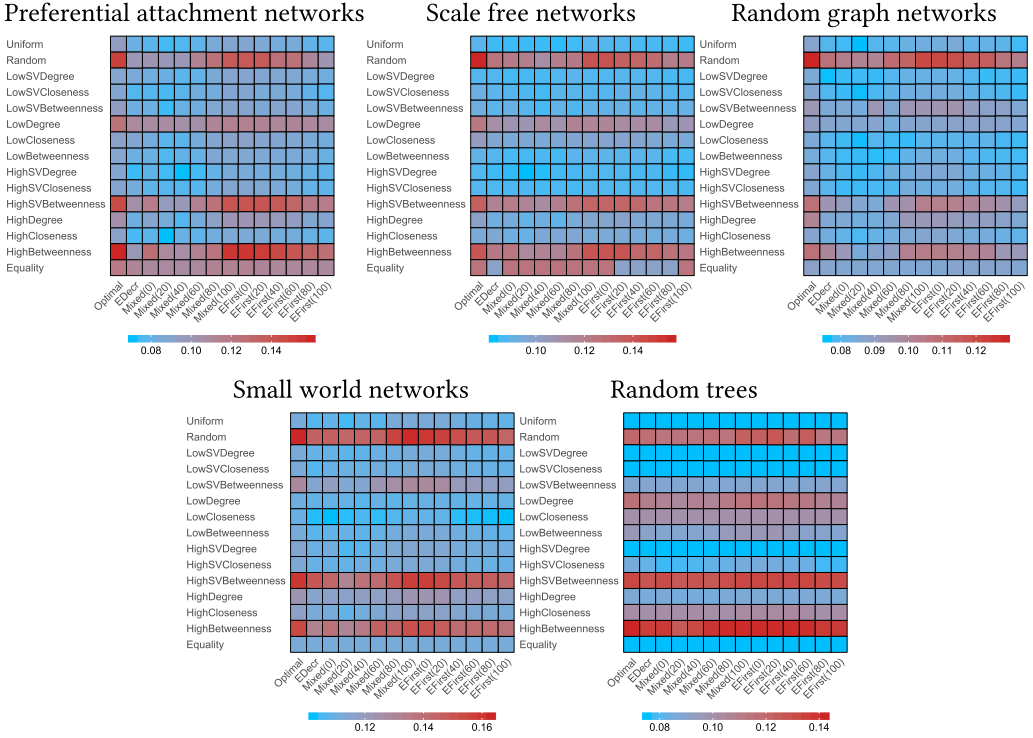
Fig. 3. Comparison of effectiveness of defender's strategies for different attacker's strategies on networks with 80 nodes. Color of each cell represents the expected percentage of nodes successfully activated by the attacker. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models.

Betweenness strategy might seem surprising at first. However, low betweenness centrality values are characteristic of not only peripheral nodes in the network, but also of the members of dense network structures (e.g., in a clique the betweenness centrality of all nodes is zero) that are typically high return targets for the attacker, as the activation of each of them affects many peers. In fact, the Low Betweenness strategy in many cases is one of the most effective ways of defending the network.

Out of the defender strategies based on closeness centrality, their relative effectiveness depends on the network model under consideration. The High Closeness strategy is more effective in the majority of the simulations. Nevertheless, they usually achieve similar performance, and choosing the right closeness-based strategy is a highly effective way of defending the network. Interestingly, these results can be connected to the role of nodes with low betweenness centrality and high closeness centrality as the "pulse-takers" of their organizations [27].

The High Degree strategy, focusing its defense efforts on hubs, is also relatively effective, as it discourages the attacker from activating high-degree nodes giving access to large parts of the network. Low Degree and Random strategies are less effective in most cases.

Out of the strategies based on the Shapley value, the High Shapley Value Betweenness is the least effective in all cases, similarly to its basic counterpart, i.e., the High Betweenness strategy. The Low Shapley Value Betweenness achieves better results, although on average it is still inferior to the rest of the strategies based on the Shapley value. Both High and Low Shapley Value Degree
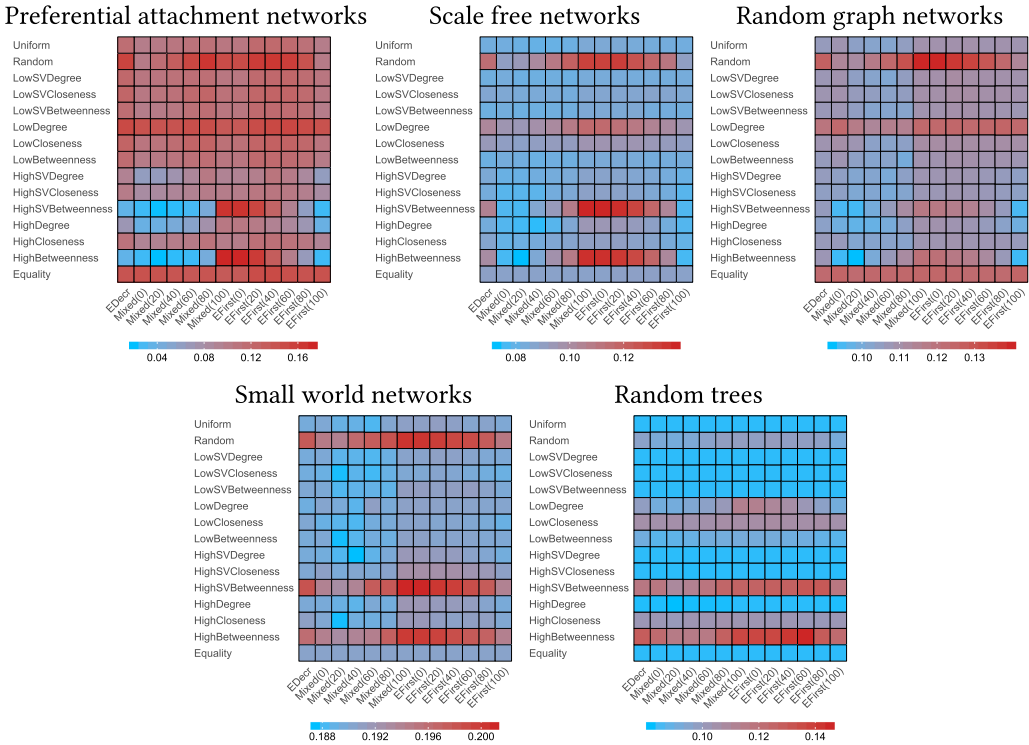
Fig. 4. Comparison of effectiveness of defender's strategies for different attacker's strategies on networks with 1K nodes. Color of each cell represents the expected percentage of nodes successfully activated by the attacker. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models.

as well as both High and Low Shapley Value Closeness are usually among the best defender's heuristic, with the exact order depending on the network structure and the strategy of the attacker. Interestingly, they are consistently more efficient than the basic centrality measures, indicating the importance of the synergy effect (expressed by the Shapley value) in planning the defense of the network.

From the attacker's point of view, if she is unable to compute the optimal solution, in most cases using the Greedy strategy with a high probability is the best choice and close to the optimal one. In most cases pure Majority and pure Greedy strategies give better results than adding a small percentage of the other strategy into the mix. The Epsilon-First strategy generally achieves better results than simple mix, while the Epsilon-Decreasing strategy is in most cases worse than both pure Majority and pure Greedy strategies.

As for the comparison between different network generation models in terms of attack potential, the small-world networks are on average the easiest to attack, as short expected distance between nodes allows to quickly reach less protected parts of the network. The networks with exponential degree distributions (i.e., preferential attachment and scale-free networks) are on average easier to attack than random graphs and trees. This may be due to the fact that hubs in networks with exponential degree distributions are more connected, and activating them gives easy access to a large part of the network. The random trees are the most difficult target for the attacker. They
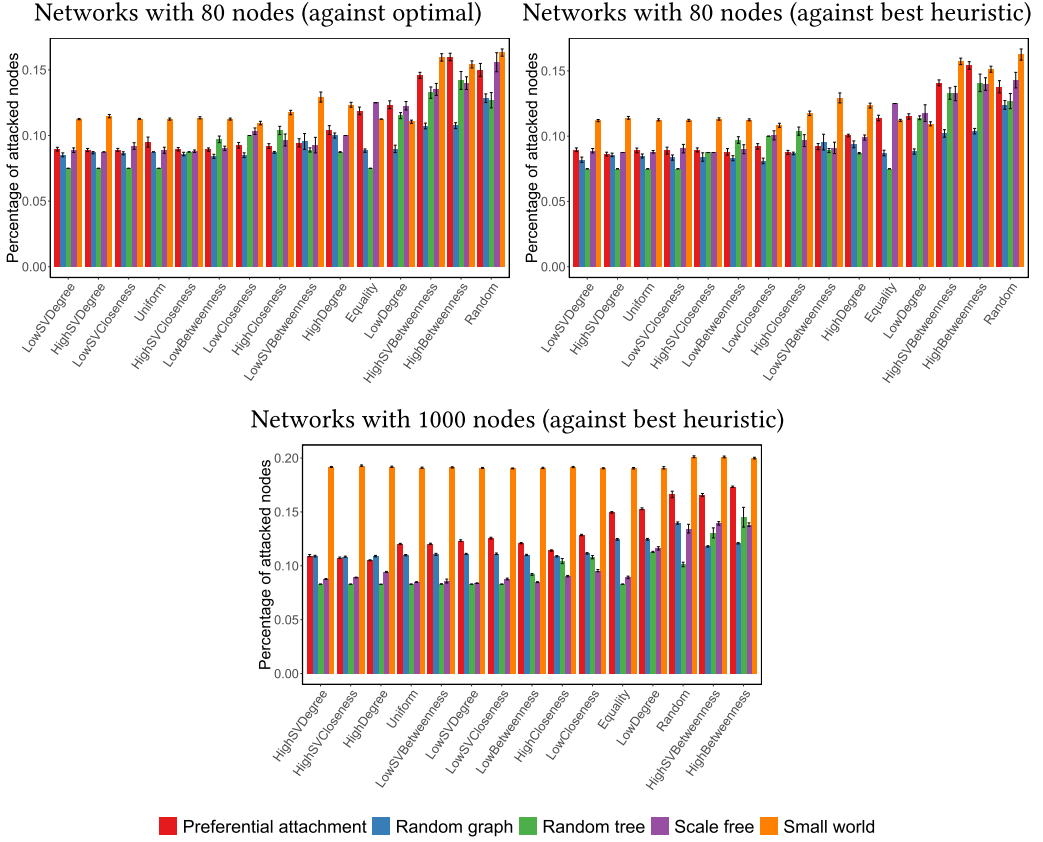
Fig. 5. Comparison of effectiveness of defender's strategies, under assumption that the attacker uses optimal strategy (for networks with 80 nodes) or the best heuristic considered by us (for networks with 1K nodes). Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models. Defender's strategies are sorted from most to least effective on average. Error bars represent 95% confidence intervals.
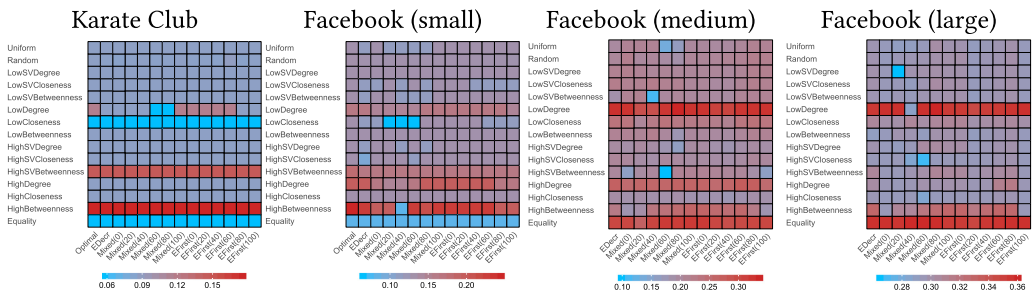

Fig. 6. Comparison of effectiveness of defender's strategies for different attacker's strategies. Color of each cell represents the expected percentage of nodes successfully activated by the attacker.
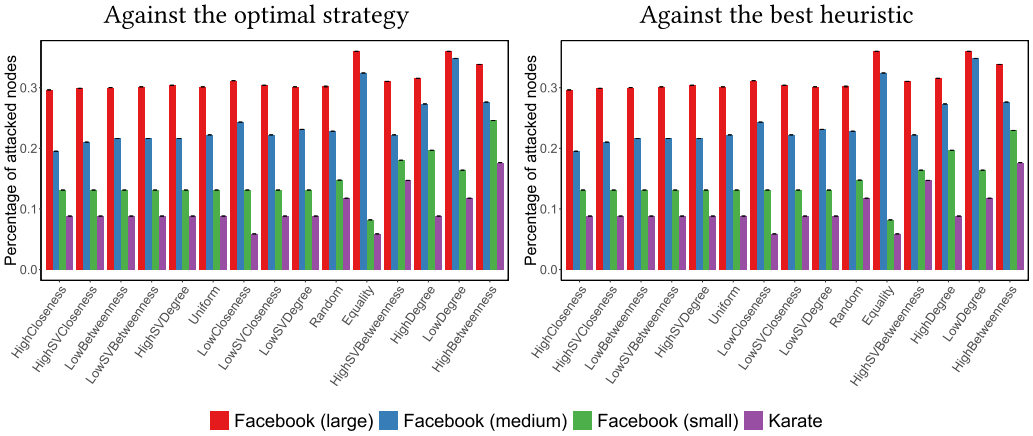
Fig. 7. Comparison of effectiveness of defender's strategies for real-life social networks, under assumption that the attacker uses optimal strategy or the best heuristic considered by us. Defender's strategies are sorted from most to least effective on average.

offer very limited choice when guiding the attack, as any two nodes are connected with but a single path.

We will focus on the choice of the best defense heuristic if the attacker utilizes her optimal strategy, which is the worst possible situation for the defender. Results for this scenario are presented in Figure 5. The best defender's choice depends on the generation model of the network. In the case of the random graphs, the Low Closeness and Low Betweenness heuristics give the best results (even though Low Closeness is on average worse than its high closeness counterpart when considering all different attacker strategies), with a number of Shapley value–based heuristics closely following. In case of the scale-free networks, the heuristics based on high scores of Shapley value degree and Shapley value closeness centrality measures provide the best results for the defender. When the preferential attachment networks are considered, the High Shapley Value Degree and the Low Shapley Value Closeness heuristics offer the best protection. For the small-world networks, Low Closeness and Low Degree heuristics are the most effective, closely followed by the Low Shapley Value Degree strategy. Finally, the random trees can be defended most efficiently by the High Shapley Value–based Degree strategy. As it can be seen, in most cases the heuristics utilizing Shapley value–based degree and closeness centralities are valid choices, albeit not always the best ones.

It might be interesting to compare the effectiveness ranking for small networks when the attacker uses the optimal strategy (top-left plot in Figure 5) and when she uses the best heuristic (top-right plot in Figure 5). As it can be seen, the ranking of the defender strategies remains almost identical, with the only difference being the Uniform strategy getting slight edge over Low Shapley Value Closeness strategy. Other trends also remain largely consistent between the two settings.

Large networks. We now discuss results for larger networks, consisting of 1K nodes. Because of time and memory complexities, we are unable to compute the optimal solution for networks of this size. However, we present results of the heuristic algorithms to observe how trends in them change with the size of the network.

Most of our observations made for the smaller networks remain valid. One notable difference in terms of defender strategies is an increase of the effectiveness of High Degree strategy. In many cases it now outperforms most of the other strategies, and the difference is especially striking in

the case of the preferential attachment networks. In this type of network, the expected degree of hubs greatly increases with size of the network, hence the attacker's advantage from activating a hub and gaining access to a large part of the network is also much higher. Because of this, focusing defender's efforts on protecting high degree nodes significantly improves effectiveness of the defense.

Another notable difference is how the effectiveness of High Betweenness strategy greatly depends on the strategy of the attacker. While it has considerable variance in effectiveness even for smaller networks, for larger networks it achieves both the best and the worst results throughout all defender strategies. It is particularly efficient to use the High Betweenness strategy against attacker strategies utilizing Majority with high probability; while we observe much worse performance against Greedy-intensive attacker strategies. A possible explanation is that in larger networks, especially these with scale-free properties, the distribution of betweenness centrality values among nodes is more heterogenous. Majority strategy, ignoring actual time of activation of the nodes, is unable to avoid attacking high betweenness, strongly defended nodes, while Greedy strategy is able to do so.

Another difference is that the Epsilon-Decreasing strategy—which usually achieves poor performance for small networks—in large networks does better in comparison to Mixed and Epsilon-First strategies. In many cases it has better results than pure Majority or pure Greedy strategies, depending on the defender's strategy. Similarly, for Mixed and Epsilon-First strategies a particular parametrization often achieves better performance than pure Greedy and Majority strategies.

While the small-world structure is the most difficult to defend for both small and large networks, the difference between it and the other models of generation is especially striking for the networks with 1K nodes. In fact, the minimal number of successfully attacked nodes for small-world networks is greater than the maximal number of successfully attacked nodes for all other types of network structure.

Since we are unable to compute the optimal strategy of the attacker for large networks, we will now analyze the choice of the defender's heuristic under assumption that the attacker will use the best response out of her strategies considered in the article (in most cases a particular parameterization of the Epsilon-First strategy). The results for this scenario are presented in Figure 5. In the case of the random graphs, a few heuristics have a very similar effectiveness: namely, High Shapley Value Closeness, High Closeness, High Shapley Value Degree, and High Degree (the difference between the best and the worst of them is smaller than a single active node). For the scale-free network, the Low Shapley Value Degree heuristic offers the best performance, with the Low Betweenness heuristic being the second. In the case of both the preferential attachment networks and the random trees, the best choice of the defender is to use the High Degree heuristic. Finally, for the small-world networks, the best performance is provided by the Low Shapley Value Closeness heuristic.

Real-life networks. Results of the simulations on real-life networks are presented in Figures 6 and 7. Notice that we computed the Optimal attack strategy only for networks with at most 80 nodes.

In general, the results for real-life networks show similar trends to those for the randomly generated datasets. In a number of situations, we observe a sharp increase or a sharp decrease in the effectiveness of a given attacker strategy for particular parametrizations (see, e.g., the Mixed(20) attacker strategy against the Low Shapley Value Degree defense in the large Facebook fragment), which is caused by the fact that, unlike for the randomly generated network, the results are not averaged over multiple runs, but rather computed for a single structure.

The comparison of results for different sizes of the networks indicates that, in general, smaller networks are easier to defend. For almost all strategies of the defender their relative effectiveness

decreases with the number of nodes (i.e., the percentage of successfully attacked nodes grows with the number of nodes).

As for the effectiveness of the defense heuristics, Random, Low Degree, High Betweenness, and High Shapley Value Betweenness are among the worst methods of defending the network, just as for the random networks.

In a situation where the total number of nodes that can be attacked is small, maximizing the cost of attacking the first node (which is the main principle behind the Equality strategy) shows its strength.

6.3 Alternative Experimental Settings

We also conducted a number of other experimental studies, the results of which are presented in the appendices. Appendix D presents the results of experiments with the mixed-integer linear programming strategy of the defender described in Section 4.3. In Appendix E, we consider a setting in which the attacker can choose multiple seeds at the beginning of the process. Appendix F presents a series of security diffusion games in which the defender updates her distribution of security resources based on the previous choices of the attacker. Finally, in Appendix G, we consider alternative forms of the function used to compute the time of activation of the network nodes.

7 DISCUSSION AND CONCLUSIONS

We considered the problem of defending the network from a sequential attack that is strategically planned and controlled by the attacker, rather than one that spreads due to a stochastic process. We proved that finding an optimal way to spread the attack is an NP-complete task in a general case, which suggests that the attacker may not always utilize the best way of diffusion. We found optimal strategies for simple network structures and formulated the problem of finding an optimal way of defending the network as a mixed-integer linear programming. For large and complex networks, we experimentally analyzed ways of defending the network from various types of attacks. We found that the highest utility of the defender is achieved by utilizing one of Shapley value–based network centrality measures. In most cases the attacker can achieve high utility by using the greedy strategy with a high probability.

One of the fundamental assumptions underlying our study is that of centralized defense, i.e., of a single authority who decides on the security level of each node. An alternative assumption that has been often considered in the network literature is that the level of defense can be chosen individually by each node in a network [25, 37]. In most of these settings, each target is autonomous and can decide whether or how to defend itself [25]. Therefore, this body of literature typically uses game-theoretic methods to identify equilibria of the game between attackers and defenders when it is possible to efficiently identify best strategies of all players. Such decentralized defense in interdependent security games fits scenarios where individuals deliberate about the incentive to invest in protection such as vaccination, airline security, or fire protection and act upon that [37]. Our study, in turn, focuses the scenarios with a central defender who is in charge of protecting all connected targets. This one defender can therefore build a resilient network or centrally assign defense resources across nodes [18, 24]. Arguably, centralized defenders can perform better in finding global defense solutions and hence can be best adopted in high-stake problems, whereas decentralized defense gives nodes the option of deciding whether to protect themselves or not and therefore unincentivized nodes might put a whole network at risk.

In a broader context, our work can be applicable and extended to other types of sequential attacks on a network [45] that are used to model, for instance, competitive adoption of a product [9] and protecting computer networks [17].

As for the ideas for future work, one possible venue is more in-depth analysis of the exploration-exploitation techniques. In this work, we considered only two simple techniques. Using more intricate learning techniques [28] might yield better results in adjusting a strategy of the attacker. Another interesting problem would be to consider a setting where, instead of playing just one game, players can face each other off multiple times on the same network, and they modify their strategies as the process goes on. Finding an equilibrium of this repeated game may lead to a better understanding of the dynamics of defending a network against a strategic continuous attack.

APPENDICES

A OPTIMAL STRATEGIES FOR CLIQUES

THEOREM 4. *Let network $G = (V, E)$ be a clique, i.e., $\forall_{v, w \in V} (v, w) \in E$. Given a particular distribution of security resources ϕ, an optimal attack sequence is γ^*, where nodes are ordered non-decreasingly according to assigned security resources, i.e., $\forall_{i < j} \phi(\gamma_i^*) \leq \phi(\gamma_j^*)$.*

An optimal defense strategy against an optimal attack strategy is to spread security resources uniformly, i.e., $\forall_{v \in V} \phi^*(v) = \frac{\Phi}{n}$.

PROOF. Since the network is a clique, the activation time of ith node in sequence γ is $\frac{n-1+\phi(\gamma_i)}{\max(1, i-1)}$. This is because γ_i has $n - 1$ neighbors, $i - 1$ of which have been already activated.

First, we prove our claim about the strategy of the attacker.

We prove it by contradiction. Assume to the contrary, that in an optimal attack sequence γ, we have $\phi(\gamma_i) > \phi(\gamma_j)$ for $i < j$ such that $i > 1$ and $j > 2$ (notice that we can always swap first and second element of the sequence, as the sum of their activation times remains the same). However, such a sequence can be improved by swapping elements i and j. Let τ_{ij} be the time of activation of γ_i and γ_j before such a swap, and τ_{ji} after the swap (note that the activation times of all other nodes remain the same after the swap). We have that (notice that $\max(1, i - 1) = i - 1$ and $\max(1, j - 1) = j - 1$ for $i > 1$ and $j > 2$):

$$
\begin{aligned}
\tau_{ji} - \tau_{ij} &= \left(\frac{n - 1 + \phi(\gamma_j)}{i - 1} + \frac{n - 1 + \phi(\gamma_i)}{j - 1} \right) - \left(\frac{n - 1 + \phi(\gamma_i)}{i - 1} + \frac{n - 1 + \phi(\gamma_j)}{j - 1} \right) \\
&= \frac{\phi(\gamma_i) - \phi(\gamma_j)}{j - 1} - \frac{\phi(\gamma_i) - \phi(\gamma_j)}{i - 1} < 0.
\end{aligned}
$$

Therefore, for $i < j$ such that $i > 1$ and $j > 2$, we always have $\phi(\gamma_i) \leq \phi(\gamma_j)$.

We will now prove our claim about the strategy of the defender. The proof is by contradiction. Assume to the contrary that in an optimal distribution of security resources ϕ, we have $i > 1$ such that $\phi(\gamma_i^*) \neq \phi(\gamma_{i+1}^*)$, where γ^* is an optimal attack sequence (again, notice that we can always swap the first and the second elements in the attack sequence, as the sum of their activation times remains the same). Since γ^* is an optimal attack sequence, we have that $\phi(\gamma_i^*) < \phi(\gamma_{i+1}^*)$. However, such ϕ can be improved by setting $\phi(\gamma_i^*)$ and $\phi(\gamma_{i+1}^*)$ to their average. Let τ_1 be time of activation of γ_i^* and γ_{i+1}^* before the change, and τ_2 after the change (the activation times of all other nodes remain the same after the change). We have:

$$
\begin{aligned}
\tau_2 - \tau_1 &= \left(\frac{n - 1 + \frac{\phi(\gamma_i^*) + \phi(\gamma_{i+1}^*)}{2}}{i - 1} + \frac{n - 1 + \frac{\phi(\gamma_i^*) + \phi(\gamma_{i+1}^*)}{2}}{i} \right) - \left(\frac{n - 1 + \phi(\gamma_i^*)}{i - 1} + \frac{n - 1 + \phi(\gamma_{i+1}^*)}{i} \right) \\
&= \frac{\phi(\gamma_{i+1}^*) - \phi(\gamma_i^*)}{2(i - 1)} - \frac{\phi(\gamma_{i+1}^*) - \phi(\gamma_i^*)}{2i} = \frac{\phi(\gamma_{i+1}^*) - \phi(\gamma_i^*)}{2i(i + 1)} > 0.
\end{aligned}
$$

Therefore, in an optimal distribution of security resources, we have $\phi(\gamma_i^*) = \phi(\gamma_{i+1}^*)$ for $i > 1$. □

B THE OPTIMAL ATTACK FOR TREES

THEOREM 5. *Let network G be a tree. The optimal strategy of the attacker can be computed in polynomial time.*

PROOF. In Algorithm 1, we present the pseudocode of the algorithm for computing the optimal attacker's strategy on a tree starting from an *a priori* given node. It performs the exhaustive search of all possible strategies of the attacker by traversing the tree in a bottom-up fashion. In the pseudocode, V_{botup} denotes the sequence of nodes in V in a bottom-up order (when v_S is the root), $c(v)$ denotes the sequence of the children of v, $c_i(v)$ denotes the ith node in this sequence, and $t(v)$ denotes the size of the subtree with v as the root. We denote the concatenation of sequences using the $\oplus$ symbol.

ALGORITHM 1: Finding the optimal attack sequence for a tree.

Input: Tree $G = (V, E)$, distribution of security resources ϕ, time limit T, starting node v_S.
Output: Optimal attack sequence starting with node v_S

1   **for** $v \in V$ **do**
2   **for** $k \in \langle 1, \ldots, n \rangle$ **do**
3         $x[v, k] \leftarrow \infty$
4   **for** $v \in V_{botup}$ **do**   `// loop over all nodes in bottom-up order`
5      $\tau[v, 0] \leftarrow 0$
6      $\gamma[v, 0] \leftarrow \emptyset$
7      $\tau[v, 1] \leftarrow d(v) + \phi(v)$
8      $\gamma[v, 1] \leftarrow \langle v \rangle$
9   **if** $d(v) > 1$ **then**   `// if v is not a leaf aggregate results from children`
10   **for** $k \in \langle 1, \ldots, t(c_1(v)) \rangle$ **do**   `// choose targets only from the subtree of the first child`
11             $y[1, k] \leftarrow d(v) + \phi(v) + \tau[c_1(v), k - 1]$
12             $q[1, k] \leftarrow \langle v \rangle \oplus \gamma[c_1(v), k - 1]$
13         $l \leftarrow t(c_1(v)) + 1$
14   **for** $m \in \langle 2, \ldots, |c(v)| \rangle$ **do**   `// choose targets from the subtrees of the first m children`
15             $l \leftarrow l + t(c_m(v))$
16   **for** $k \in \langle 1, \ldots, l \rangle$ **do**   `// loop over the number of attacked nodes`
17                $y[m, k] \leftarrow \infty$
18   **for** $i \in \langle 1, \ldots, k \rangle$ **do**   `// loop over the number of attacked nodes, we select from the subtrees of the first m − 1 children`
19                    $\tilde{y} \leftarrow y[m - 1, i] + \tau[c_m(v), k - i]$
20   **if** $\tilde{y} < y[m, k]$ **then**
21                       $y[m, k] \leftarrow \tilde{y}$
22                       $q[m, k] \leftarrow q[m - 1, i] \oplus \gamma[c_m(v), k - i]$
23   **for** $k \in \langle 2, \ldots, t(v) \rangle$ **do**   `// transfer results to tables τ and γ`
24             $\tau[v, k] \leftarrow y[|c(v)|, k]$
25             $\gamma[v, k] \leftarrow q[|c(v)|, k]$
26   **return** $\max_{k: \tau[v_S, k] < T} \gamma[v_S, k]$

One can notice that, in the case of a tree, every node other than the seed node has exactly one active neighbor at the moment of activation. Hence, its time of activation is always $d(v) + \phi(v)$. In what follows, we will call this value the *weight* of node v. The task of finding the optimal strategy of the attacker boils down to finding the largest subtree with the sum of weights lower than T. Algorithm 1 achieves this goal by filling tables τ and γ.

Entry $\tau[v, k]$ contains the weight of the lightest subtree with k nodes rooted at v, while $\gamma[v, k]$ contains the nodes in this subtree (in the order of activation). Computing values of $\tau[v, 0]$ and $\tau[v, 1]$ (as well as $\gamma[v, 0]$ and $\gamma[v, 1]$) is trivial. If node v is not a leaf, then we compute $\tau[v, k]$ and $\gamma[v, k]$ for $k > 1$ by filling tables y and q.

Entry $y[m, k]$ contains the weight of the lightest subtree with k nodes rooted at v, constructed by using only the descendants of the first m children of v. Entry $q[m, k]$ contains nodes in this subtree (in the order of activation). In lines 18–22, we check all possible values of $y[m, k]$ by taking i nodes from the first $m - 1$ children of v (and v itself) and adding to them $k - i$ nodes from the subtree of the mth child of v.

Instructions in lines 11–12 as well as instructions in lines 25–26 are run $O(n^2)$ times, since the loop in line 4 is run exactly n times and, for any v, we have $t(v) \leq n$. One can notice that the body of the loop in line 14 is run exactly $n - 1$ times in total (because it is a loop over children of node v and every node other than v_S has exactly one parent). Since both $l \leq n$ and $k \leq n$, instructions in lines 19–22, driving the time complexity of the algorithm, are run $O(n^3)$ times. Hence, the time complexity of Algorithm 1 is $O(n^3)$. The memory complexity of Algorithm 1 is $O(n^2)$ if we represent sequences in tables γ and q as trees visited in a prefix order. □

C DYNAMIC PROGRAMMING ALGORITHM FOR FINDING OPTIMAL ATTACK

In this section, we present a dynamic-programming algorithm to find the optimal strategy of the attacker. Its pseudocode can be found in Algorithm 2. The algorithm computes the strategy that yields the highest possible utility of the attacker against a given distribution of security resources. The $\oplus$ symbol denotes the concatenation of sequences.

ALGORITHM 2: A dynamic-programming algorithm for the optimal strategy of the attacker.

> **Input**: The network (V, E), the defender's distribution of security resources ϕ, the seed node $v_S \in V$, and the attacker's time limit T.
>
> **Output**: Attacker's sequence of activation starting with v_S of maximal length given the time limit.

1 **for** $C \subseteq V$ **do**
2      $\tau^*[C] \leftarrow \infty$
3 $\tau^*[\{v_S\}] \leftarrow 0$
4 $\gamma^*[\{v_S\}] \leftarrow \langle v_S \rangle$
5 **for** $k = 1, \ldots, n - 1$ **do**
6   **for** $C \subset V : (|C| = k) \wedge (\tau^*[C] < \infty)$ **do**
7   **for** $v \in V : (v \notin C) \wedge (N(v) \cap C \neq \emptyset)$ **do**
8              $\Delta \tau \leftarrow \frac{d(v) + \phi(v)}{|N(v) \cap C|}$
9   **if** $(\tau^*[C] + \Delta \tau < \tau^*[C \cup \{i\}]) \wedge (\tau^*[C] + \Delta \tau < T)$ **then**
10                 $\tau^*[C \cup \{i\}] \leftarrow \tau^*[C] + \Delta \tau$
11                 $\gamma^*[C \cup \{i\}] \leftarrow \gamma^*[C] \oplus \langle i \rangle$
12   **if** $\neg \exists_{C \subseteq V} |C| = k + 1 \wedge \tau^*[C] < \infty$ **then**
13   **return** $\gamma^*[\arg \min_{C \subseteq V : |C| = k} \tau^*[C]]$
14 **return** $\gamma^*[V]$

In lines 1–4, we initialize data structures, i.e., array τ^* holding the minimal time necessary to activate a given set of nodes and array γ^* holding the sequence allowing to activate a given set of nodes in this minimal time. In kth run of loop in line 5, we fill both arrays for these subsets of size $k + 1$ that can be activated within time limit. We do it by iterating over the subsets of size k that can be activated within time limit in loop in line 6 and trying to activate next node in loop

in line 7. In line 8, we compute the time necessary to active node v when previously we activated nodes in C in the best possible way. In lines 9–11, we update the way of activating nodes $C \cup \{v\}$ if activating nodes in C first and node v after that is better than the currently known best solution. We end the algorithm either when it is no longer possible to activate any set of the size $k + 1$ (in lines 12–13) or when we find a way of activating the entire network (in line 14).

Time complexity of the algorithm is $O(2^n n^2)$, as for every of the $O(2^n)$ subsets of the set of nodes, we might need to iterate over $O(n)$ of its neighbors and for each of them compute their time of activation in time $O(n)$. Notice that arrays can be implemented using hash tables. Given the exponential time complexity of the algorithm, it is only suitable for small networks.

D EXPERIMENTAL RESULTS WITH MIXED-INTEGER LINEAR PROGRAMMING

To test the effectiveness of mixed-integer linear programming in finding the optimal strategy of the defender, we perform experiments on randomly generated networks. To avoid selecting arbitrary set of strategies $\hat{\Gamma}$ available to the attacker, we allow the attacker to choose any strategy. However, under these assumptions, we are able to solve mixed-integer linear programming instance efficiently only for very small networks. A potential idea for future work is studying whether there exists a way of finding a set $\hat{\Gamma}$ of limited size that is still guaranteed to always contain the optimal attacker strategy.

Figures 8 and 9 present our results for networks with 6 nodes and average degree 2, taken as an average over 100 simulations. Mixed-integer linear programming was solved using *lp_solve* solver version 5.5.2.5.

As it can be seen, distribution of security resources computed using mixed-integer linear programming is considerably more successful in mitigating the attack than most of the other defense strategies, with centrality-based heuristics closely following. This suggests that, at least in the case of simple structures of very small networks, centrality measures may be close to capturing the complexity of the problem of finding the optimal defense.

Another difference in comparison to results for larger networks is that all considered strategies of the attacker achieve very similar results. Very small number of nodes results in a fairly limited variety of valid strategies available to the attacker.
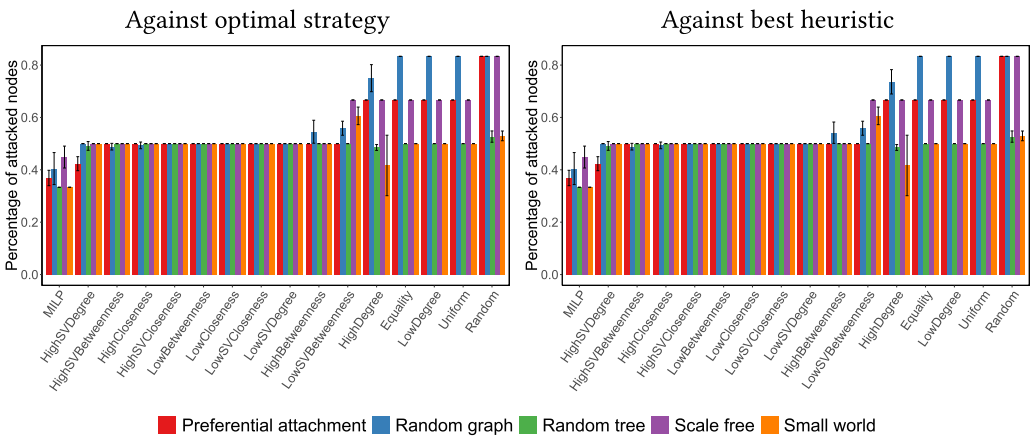

Fig. 8. Comparison of effectiveness of defender's strategies, under assumption that the attacker uses optimal strategy or the best heuristic considered by us. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models. Defender's strategies are sorted from most to least effective on average. Error bars represent 95% confidence intervals.
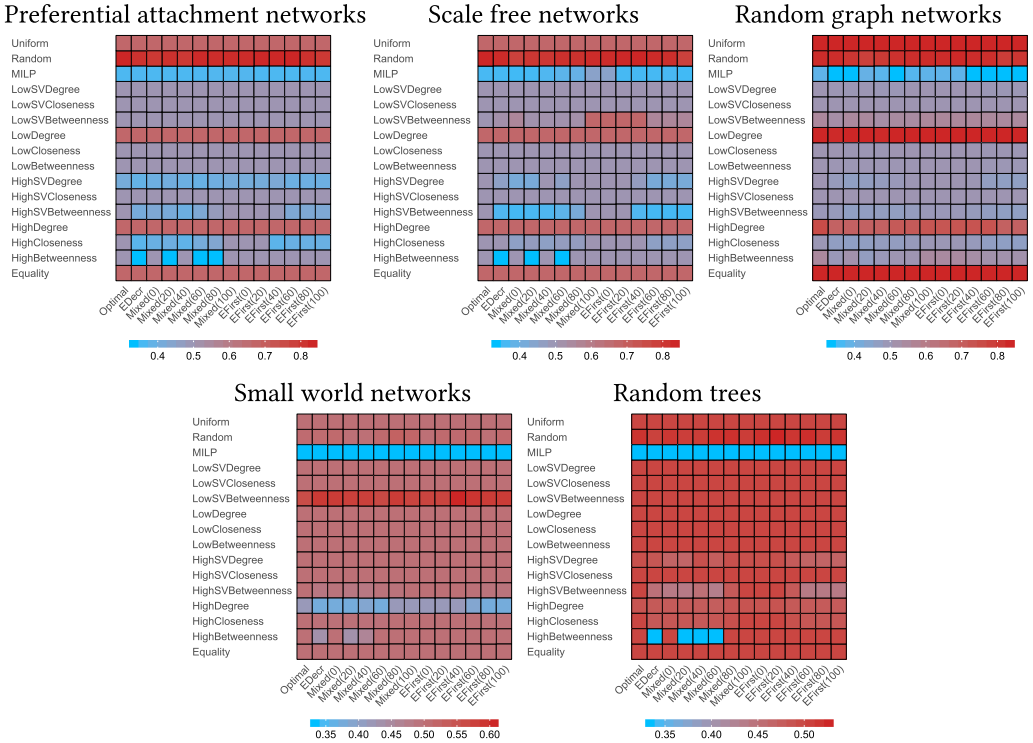
Fig. 9. Comparison of effectiveness of defender's strategies for different attacker's strategies. Color of each cell represents the expected percentage of nodes successfully activated by the attacker. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models.

E EXPERIMENTAL RESULTS WITH MULTIPLE SEEDS

The model of strategic diffusion [2] allows only one seed node. We now investigate the potential advantages of activating multiple seeds at the beginning of the process. We assume the time of activating each subsequent seed to be computed using the same formula as for the first seed in the basic version of our model, i.e., $\tau(v) = d(v) + \phi(v)$ for v that is a seed node. We choose the subsequent seeds using the same method as when choosing the first seed in the basic version of the model, i.e., selecting those that provide the highest number of activated nodes (breaking ties uniformly at random). In the remainder of the process, we follow the same procedure as in the basic version of the model, i.e., we choose the subsequent targets from the nodes with at least one active neighbor.

Figures 10 and 11 present the results of our experiments. As it can be seen, forcing the attacker to pick multiple seeds increases the effectiveness of the attack in most cases, as the expected percentage of successfully attacked nodes is greater when two or three seeds are activated (color of the heatmap cells is closer to red). An interesting exception to this rule are the random graph networks generated using Erdős-Rényi model. The uniform structure of these networks does not offer much advantage in terms of activating additional seeds.

It is also worth noting that the general trends in data are preserved when changing the number of seeds, e.g., strategies of the defender that are effective in experiments with single seeds tend to also be effective in a setting with multiple seeds. Finally, although the difference in results when using multiple seeds is notable, the magnitude of the change is not particularly substantial, e.g.,
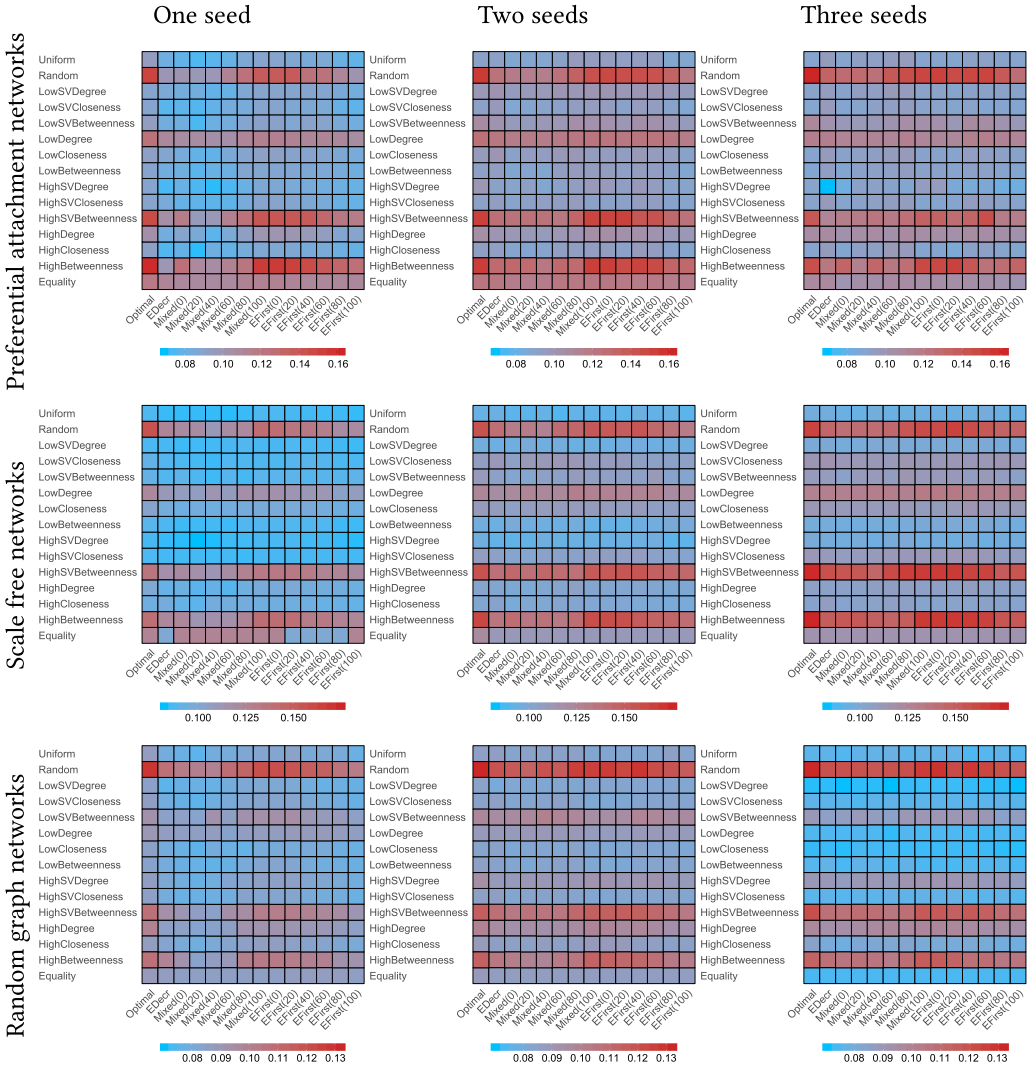
Fig. 10. Comparison of effectiveness of defender's strategies for different attacker's strategies on networks with 80 nodes, when attacker uses one, two, and three seeds. Color of each cell represents the expected percentage of nodes successfully activated by the attacker. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models. Scales are fixed for each network type to allow comparison between the number of seeds.

the difference between the percentage of successfully attacked nodes with only one seed and with two seeds is on average less than 0.01.
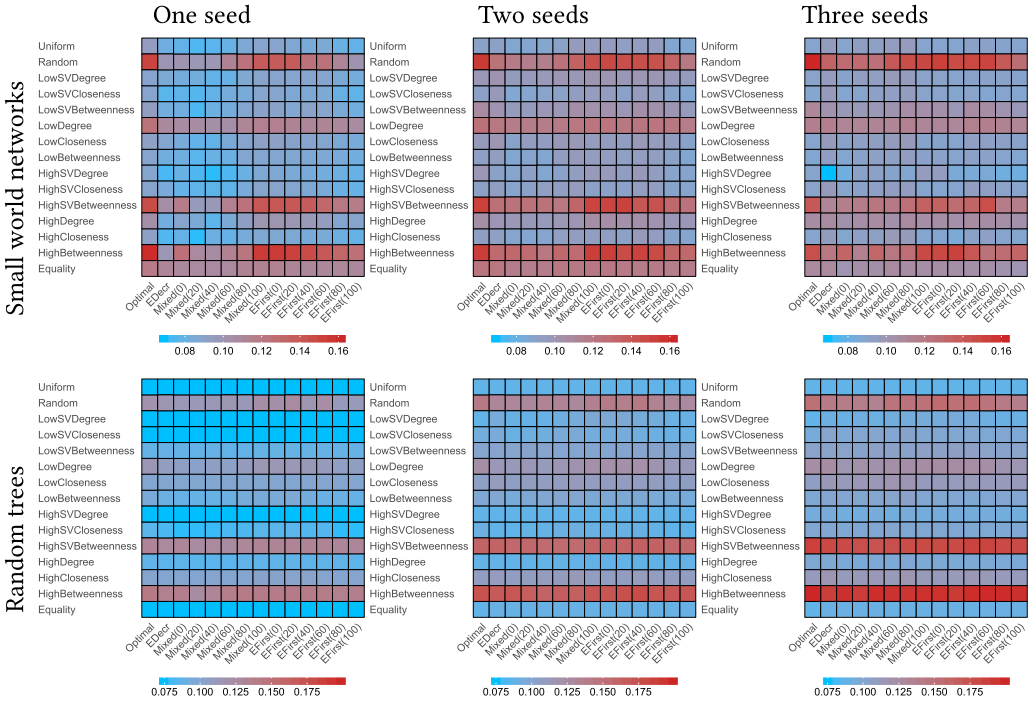
Fig. 11. Comparison of effectiveness of defender's strategies for different attacker's strategies on networks with 80 nodes, when attacker uses one, two, and three seeds. Color of each cell represents the expected percentage of nodes successfully activated by the attacker. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models. Scales are fixed for each network type to allow comparison between the number of seeds.

F REPEATED GAME SETTING

We now consider a process in which the security diffusion game is played multiple times between the same two players. We investigate whether the knowledge about the attacker's activities can help the defender achieve better performance.

We consider a process consisting of rounds. In each round, the defender and the attacker play a single instance of the security diffusion game on the same network. We assume that the attacker always utilizes the Optimal strategy, while the defender updates her distribution of security resources according to results of the previous round. The total amount of security resources remains the same.

In what follows, let $\phi_j(v_i)$ denote the amount of security resources assigned to node v_i in round j, and let I_j denote the set of nodes activated in round j.

The value of ϕ_1 is given by the strategy used by the defender in the first round, which is one of the strategies described for the single game setting. For every subsequent round $j > 1$ the defender updates the security resources assignment as follows:

$$\phi_j(v_i) = \frac{\omega_j(v_i)}{\sum_{i=1}^{n} \omega_j(v_i)} \Phi,$$

where:

$$\omega_j(v_i) = \begin{cases} r_{i,j} \phi_{j-1}(v_i) & \text{if } v_i \in I_{j-1}, \\ \phi_{j-1}(v_i) & \text{otherwise,} \end{cases}$$
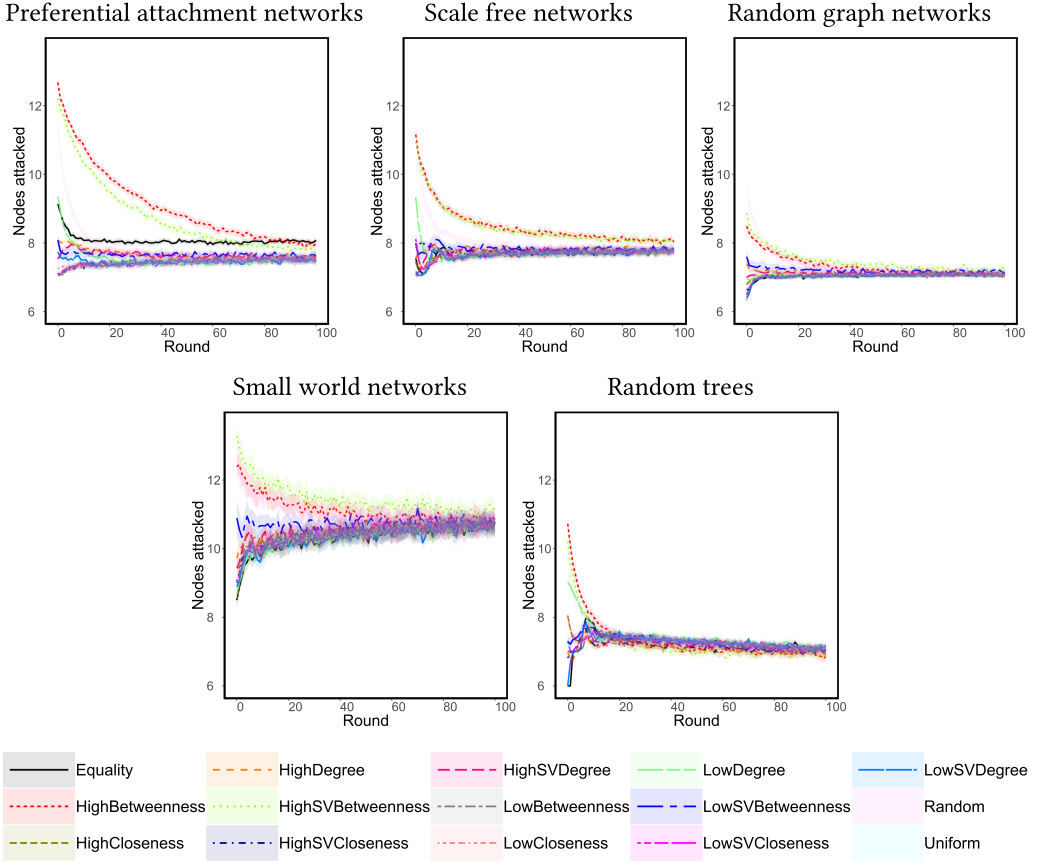
Fig. 12. Results of our experiments for repeated game setting. The x-axis represents the round, i.e., the consecutive instance of the game. The y-axis represents the number of activated nodes after the process finished. Every line represents different initial strategy of the defender. The scale is fixed to make plots easier to compare.

with $r_{i,j}$ being drawn uniformly at random from interval $[1, 2]$. In other words, the defender increases the amount of security resources assigned to nodes that were activated in the previous round.

Figure 12 presents results for our repeated game setting. Each data point is taken as an average over 100 simulations. Colored areas represent 95% confidence interval.

As it can be seen, updating the strategy that initially is not particularly successful leads to reducing the number of nodes captured by the attacker. However, in case of the defender strategies that are the most effective in the first round, moving the security resources to the nodes that got successfully attacked not only does not help, but it actually makes the network more vulnerable, i.e., in the following rounds the average number of captured nodes increases. By moving the security resources to nodes that were attacked in the previous round, the defender makes other nodes more vulnerable, which is then exploited by the attacker.

Interestingly, in most cases almost all initial strategies converge to similar efficiency. It suggests that after a high enough number of rounds an equilibrium state is reached.

G EXPERIMENTAL RESULTS WITH ALTERNATIVE DIFFUSION MODELS

We now consider alternative settings of our experiments, where we modify the formula of the activation time of a given node. In particular, following the strategic diffusion model [2], we introduce the parameter $\alpha \in \mathbb{R}^+$, which allows to adjust the strength of path dependencies. In a setting with a given value of α, we have that:

- The first node in the sequence chosen by the attacker (the seed node) is activated in time $\tau(v) = (d(v) + \phi(v))^\alpha$.
- The activation time for all other nodes is $\tau(v) = (\frac{d(v)+\phi(v)}{|N(v) \cap I|})^\alpha$ where I is the set of currently activated nodes.

Notice that for $\alpha = 1$ this is equivalent to the basic model described in Section 3 of the main article.

The results of our experiments with $\alpha = 0.8$ and with $\alpha = 1.25$ are presented in Figures 13, 14, and 15.

As for the relative effectiveness of the different strategies of the attacker, the general trends are similar to those in experiments with the basic model with $\alpha = 1$. The parameters of the attacker's strategies that proved to be successful in the basic version of the experiments are usually also effective when the function determining the time of activation changes.
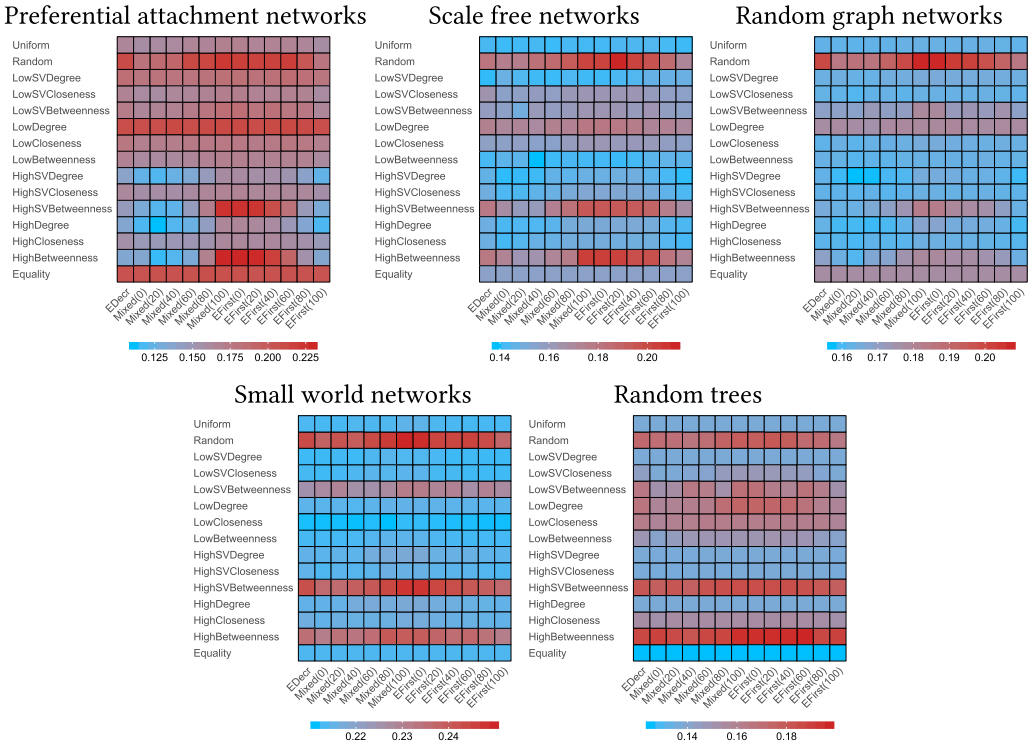

Fig. 13. Comparison of effectiveness of defender's strategies for different attacker's strategies on networks with 80 nodes in model with $\alpha = 0.8$. Color of each cell represents the expected percentage of nodes successfully activated by the attacker. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models.
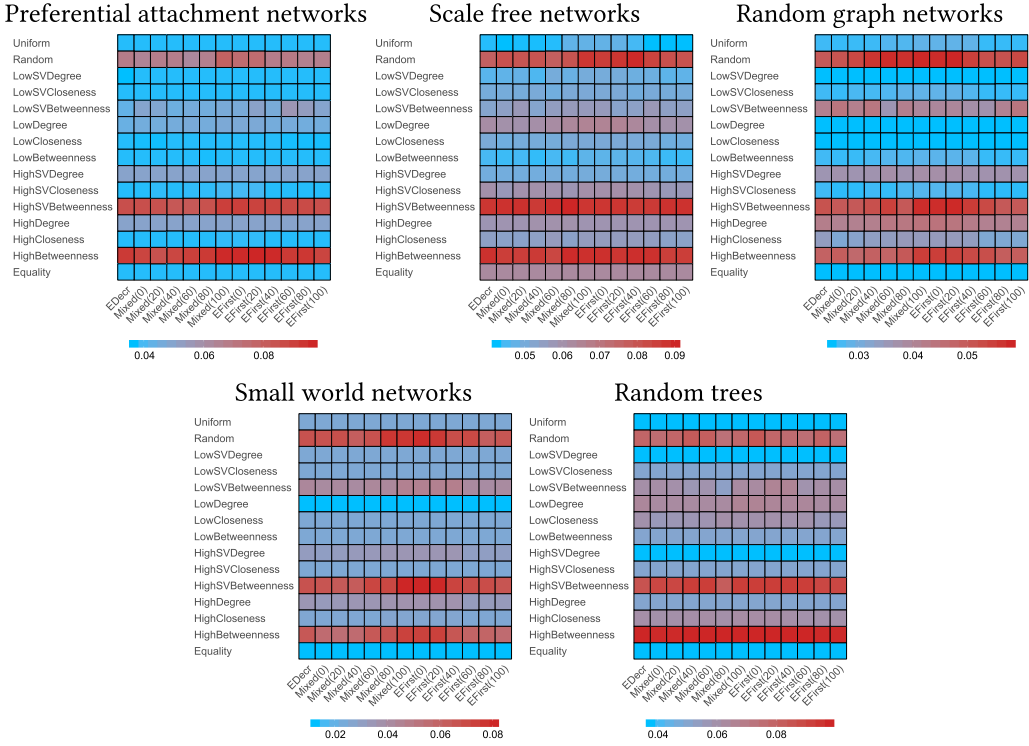
Fig. 14. Comparison of effectiveness of defender's strategies for different attacker's strategies on networks with 80 nodes in model with $\alpha = 1.25$. Color of each cell represents the expected percentage of nodes successfully activated by the attacker. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models.
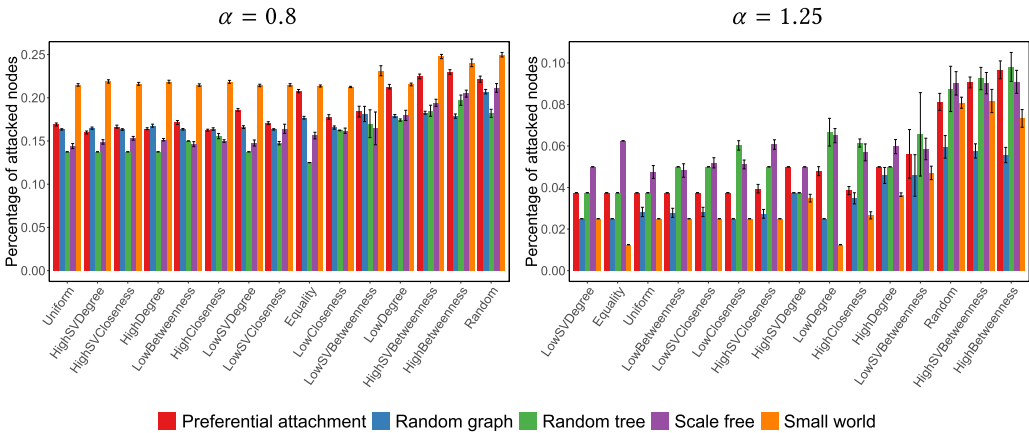

Fig. 15. Comparison of effectiveness of defender's strategies, under assumption that the attacker uses the best heuristic considered by us, for models with different values of α. Results are taken as an average over 100 simulations, with a new network generated for each simulation using one of the models. Defender's strategies are sorted from most to least effective on average. Error bars represent 95% confidence intervals.

As for the relative effectiveness of the defender's strategies, results for the setting with $\alpha = 0.8$ show similar tendencies to those for $\alpha = 1$. However, increasing the value of α to 1.25 brings some interesting insights. The Equality strategy, in other cases not particularly successful, suddenly becomes one of the best ways of defending the network. In a situation where capturing the nodes becomes significantly harder (which is the case for $\alpha > 1$), withholding the initial attack also becomes more crucial (which is the idea behind the Equality strategy).

As for the comparison between different network models, while results for $\alpha = 0.8$ present similar trends to the results for the basic setting, again an interesting change can be noticed when $\alpha = 1.25$. While in other cases the small-world networks were the easiest to attack, they are more resilient when $\alpha > 1$. Increasing the value of α puts more pressure on activating nodes with many active neighbors and reduces the advantage of being able to quickly reach all parts of the network.

REFERENCES

[1] Daron Acemoglu, Azarakhsh Malekian, and Asu Ozdaglar. 2016. Network security and contagion. *J. Econ. Theor.* 166 (2016), 536–585.

[2] Aamena Alshamsi, Flávio L. Pinheiro, and Cesar A. Hidalgo. 2018. Optimal diversification strategies in the networks of related products and of related research areas. *Nat. Commun.* 9, 1 (2018), 1328.

[3] Jac M. Anthonisse. 1971. The rush in a directed graph. *Stichting Mathematisch Centrum. Mathematische Besliskunde* BN 9/71 (1971), 1–10.

[4] W. Brian Arthur. 1989. Competing technologies, increasing returns, and lock-in by historical events. *Econ. J.* 99, 394 (1989), 116–131.

[5] Yoram Bachrach, Moez Draief, and Sanjeev Goyal. 2013. Contagion and observability in security domains. In *Proceedings of the 51st Allerton Conference on Communication, Control, and Computing.* IEEE, Piscataway, NJ, 1364–1371.

[6] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

[7] Murray A. Beauchamp. 1965. An improved index of centrality. *Behav. Sci.* 10, 2 (1965), 161–163.

[8] Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. 1992. A theory of fads, fashion, custom, and cultural change as informational cascades. *J. Polit. Econ.* 100, 5 (1992), 992–1026.

[9] Matthias Broecheler, Paulo Shakarian, and V. S. Subrahmanian. 2010. A scalable framework for modeling competitive diffusion in social networks. In *Proceedings of the IEEE 2nd International Conference on Social Computing (SocialCom'10).* IEEE, Piscataway, NJ, 295–302.

[10] Rosario Ceballo. 2000. The neighborhood club: A supportive intervention group for children exposed to urban violence. *Amer. J. Orthopsych.* 70, 3 (2000), 401–407.

[11] Damon Centola. 2010. The spread of behavior in an online social network experiment. *Science* 329, 5996 (2010), 1194–1197.

[12] Damon Centola. 2011. An experimental study of homophily in the adoption of health behavior. *Science* 334, 6060 (2011), 1269–1272.

[13] Damon Centola and Michael Macy. 2007. Complex contagions and the weakness of long ties. *Amer. J. Sociol.* 113, 3 (2007), 702–734.

[14] Yongmin Chen and Jason Pearcy. 2010. Dynamic pricing: When to entice brand switching and when to reward consumer loyalty. *RAND J. Econ.* 41, 4 (2010), 674–685.

[15] Flavio Chierichetti, Jon Kleinberg, and Alessandro Panconesi. 2014. How to schedule a cascade in an arbitrary graph. *SIAM J. Comput.* 43, 6 (2014), 1906–1920.

[16] Zoltán Dezső and Albert-László Barabási. 2002. Halting viruses in scale-free networks. *Phys. Rev. E* 65, 5 (2002), 055103.

[17] Karel Durkota and Viliam Lisỳ. 2014. Computing optimal policies for attack graphs with action failures and costs. In *Proceedings of the European Starting AI Researcher Symposium (STAIRS'14).* IOS Press, 101–110.

[18] Marcin Dziubiński and Sanjeev Goyal. 2017. How do you defend a network? *Theoret. Econ.* 12, 1 (2017), 331–376.

[19] Paul Erdős and Alfréd Rényi. 1959. On random graphs I. *Publ. Math. Debrecen* 6 (1959), 290–297.

[20] Federico Etro. 2007. Stackelberg competition and endogenous entry. In *Competition, Innovation, and Antitrust: A Theory of Market Leaders and Its Policy Implications.* Springer, Berlin, 91–129.

[21] Linton C. Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry* 40 (1977), 35–41.

[22] Xavier Freixas and Bruno Parigi. 1998. Contagion and efficiency in gross and net interbank payment systems. *J. Finan. Intermed.* 7, 1 (1998), 3–31.

[23] Drew Fudenberg and Jean Tirole. 2000. Customer poaching and brand switching. *J. Econ.* 31 (2000), 634–657.

[24] Sanjeev Goyal and Adrien Vigier. 2014. Attack, defence, and contagion in networks. *Rev. Econ. Stud.* 81, 4 (2014), 1518–1542.

[25] Sanjeev Goyal and Adrien Vigier. 2015. Interaction, protection and epidemics. *J. Pub. Econ.* 125 (2015), 64–69.

[26] Bronwyn H. Hall and Beethika Khan. 2003. *Adoption of New Technology*. Working Paper 9730. National Bureau of Economic Research. https://doi.org/10.3386/w9730

[27] Charles Henderson and Kathleen Quardokus. 2013. Department-level change: Using social network analysis to map the hidden structure of academic departments. *AIP Conf. Proc.*, Vol. 1513. AIP, Melville, NY, 170–173.

[28] Junling Hu and Michael P. Wellman. 1998. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the International Conference on Machine Learning (ICML'98)*, Vol. 98. Morgan Kaufmann Publishers Inc., Burlington, MA, 242–250.

[29] Raghuram Iyengar, Christophe Van den Bulte, and Thomas W Valente. 2011. Opinion leadership and social contagion in new product diffusion. *Market. Sci.* 30, 2 (2011), 195–212.

[30] Jarosław Jankowski, Piotr Bródka, Przemysław Kazienko, Boleslaw K. Szymanski, Radosław Michalski, and Tomasz Kajdanowicz. 2017. Balancing speed and coverage by sequential seeding in complex networks. *Sci. Rep.* 7, 1 (2017), 891.

[31] Jarosław Jankowski, Marcin Waniek, Aamena Alshamsi, Piotr Bródka, and Radosław Michalski. 2018. Strategic distribution of seeds to support diffusion in complex networks. *PloS One* 13, 10 (2018), e0205130.

[32] Benjamin Johnson, Jens Grossklags, Nicolas Christin, and John Chuang. 2010. Uncertainty in interdependent security games. In *Proceedings of the International Conference on Decision and Game Theory for Security*. Springer, Berlin, 234–244.

[33] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. 1996. Reinforcement learning: A survey. *J. Artific. Intell. Res.* 4 (1996), 237–285.

[34] Márton Karsai, Gerardo Iniguez, Kimmo Kaski, and János Kertész. 2014. Complex contagion process in spreading of online innovation. *J. Roy. Soc. Interf.* 11, 101 (2014), 20140694.

[35] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 137–146.

[36] Robert V. Kozinets, Kristine De Valck, Andrea C. Wojnicki, and Sarah J. S. Wilner. 2010. Networked narratives: Understanding word-of-mouth marketing in online communities. *J. Market.* 74, 2 (2010), 71–89.

[37] Howard Kunreuther and Geoffrey Heal. 2003. Interdependent security. *J. Risk Uncert.* 26, 2–3 (2003), 231–249.

[38] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine generals problem. *ACM Trans. Progr. Lang. Syst.* 4, 3 (1982), 382–401.

[39] Jure Leskovec and Julian J. Mcauley. 2012. Learning to discover social circles in ego networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'12)*. 539–547.

[40] E. Rutger Leukfeldt, Edward R. Kleemans, and Wouter P. Stol. 2017. Cybercriminal networks, social ties, and online forums: Social ties versus digital ties within phishing and malware networks. *Brit. J. Criminol.* 57, 3 (2017), 704–722.

[41] Jian Lou, Andrew M. Smith, and Yevgeniy Vorobeychik. 2017. Multidefender security games. *IEEE Intell. Syst.* 32, 1 (2017), 50–60.

[42] Joanna Matloka and Dimitrios Buhalis. 2010. Destination marketing through user personalised content (UPC). *Inform. Commun. Technol. Tour.* 2010 (2010), 519–530.

[43] Tomasz P. Michalak, Karthik V. Aadithya, Piotr L. Szczepanski, Balaraman Ravindran, and Nicholas R. Jennings. 2013. Efficient computation of the Shapley value for game-theoretic network centrality. *J. Artific. Intell. Res.* 46 (2013), 607–650.

[44] Mark E. J. Newman. 2003. The structure and function of complex networks. *SIAM Rev.* 45, 2 (2003), 167–256.

[45] Kien C. Nguyen, Tansu Alpcan, and Tamer Basar. 2009. Stochastic games for security in networks with interdependent nodes. In *Proceedings of the International Conference on Game Theory for Networks (GameNets'09)*. IEEE, Piscataway, NJ, 697–703.

[46] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. 2008. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *Proceedings of the International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS'08)*. IFAAMAS, Columbia, SC, 895–902.

[47] Romualdo Pastor-Satorras and Alessandro Vespignani. 2001. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.* 86, 14 (2001), 3200.

[48] James Pita, Manish Jain, Janusz Marecki, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. 2008. Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles International Airport. In *Proceedings of the International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS'08)*. IFAAMAS, Columbia, SC, 125–132.

[49] H. Prüfer. 1918. Neuer beweis eines satzes über permutationen. *Arch. Math. Phys. 3. Reihe* 27 (01 1918), 142–144.

[50] Shaheen Rana and Nicole E. Allen. 2015. Centrality measures to identify key stakeholders in Family Violence Councils. *Psychosoc. Interv.* 24, 3 (2015), 167–176.

[51] Sankardas Roy, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Vivek Shandilya, and Qishi Wu. 2010. A survey of game theory as applied to network security. In *Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS'10)*. IEEE, Piscataway, NJ, 1–10.

[52] Alsaleh Saad and Alwaleed Alhumaid. 2018. How terrorist groups use social networking media to attract and recruit new members. In *Proceedings of the International Conference on Innovations in Information Technology (IIT'18)*. IEEE, Piscataway, NJ, 129–134.

[53] Marvin E. Shaw. 1954. Group structure and the behavior of individuals in small groups. *J. Psychol.* 38, 1 (1954), 139–149.

[54] Eric Anyung Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. 2012. PROTECT: An application of computational game theory for the security of the ports of the United States. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'12)*. The AAAI Press, Palo Alto, CA, 2173–2179.

[55] Stergios Skaperdas. 1996. Contest success functions. *Econ. Theor.* 7, 2 (1996), 283–290.

[56] Oskar Skibski, Tomasz P. Michalak, and Talal Rahwan. 2018. Axiomatic characterization of game-theoretic centrality. *J. Artific. Intell. Res.* 62 (2018), 33–68.

[57] Karen Stephenson. 2004. *The Quantum Theory of Trust: The Secret of Mapping and Managing Human Relationships*. Financial Times Prentice Hall, Upper Saddle River, NJ.

[58] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. Vol. 1. The MIT Press, Cambridge, MA.

[59] Piotr L. Szczepański, Tomasz P. Michalak, and Talal Rahwan. 2016. Efficient algorithms for game-theoretic betweenness centrality. *Artific. Intell.* 231 (2016), 39–63.

[60] Milind Tambe. 2011. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, Cambridge, MA.

[61] Mateusz K. Tarkowski, Piotr L. Szczepański, Tomasz P. Michalak, Paul Harrenstein, and Michael Wooldridge. 2018. Efficient computation of semivalues for game-theoretic network centrality. *J. Artific. Intell. Res.* 63 (2018), 145–189.

[62] Jason Tsai, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Shyamsunder Rathi. 2009. IRIS—A tool for strategic security allocation in transportation networks. In *Proceedings of the International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS'09)*. IFAAMAS, Columbia, SC, 37–44.

[63] Eric Vanderburg. 2017. The 6 Stages of an Advanced Ransomware Threat Attack. *Carbonite*. www.carbonite.com/blog/article/2017/08/the-6-stages-of-an-advanced-ransomware-threat-attack/.

[64] Heinrich Von Stackelberg. 1934. *Marktform und Gleichgewicht*. J. Springer, Berlin.

[65] Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of small-world networks. *Nature* 393, 6684 (1998), 440–442.

[66] Wayne W. Zachary. 1977. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* 33 (1977), 452–473.