Knowl Inf Syst (2017) 52:657–685
DOI 10.1007/s10115-017-1023-7

REGULAR PAPER

Monitoring stealthy diffusion

Nika Haghtalab[1] · **Aron Laszka**[2] · **Ariel D. Procaccia**[1] · **Yevgeniy Vorobeychik**[2] · **Xenofon Koutsoukos**[2]

Received: 14 November 2015 / Revised: 28 November 2016 / Accepted: 28 January 2017 /
Published online: 13 February 2017

Abstract A broad variety of problems, such as targeted marketing and the spread of viruses and malware, have been modeled as maximizing the reach of diffusion through a network. In cyber-security applications, however, a key consideration largely ignored in this literature is stealth. In particular, an attacker who has a specific target in mind succeeds only if the target is reached before the malicious payload is detected and corresponding countermeasures deployed. The dual side of this problem is deployment of a limited number of monitoring units, such as cyber-forensics specialists, to limit the success of such targeted and stealthy diffusion processes. We investigate the problem of optimal monitoring of targeted stealthy diffusion processes. While natural variants of this problem are NP-hard, we show that if stealthy diffusion starts from randomly selected nodes, the defender's objective is submodular and can be approximately optimized. In addition, we present approximation algorithms for the setting where the choice of the starting point is adversarial. We further extend our results to settings where the diffusion starts at multiple-seed nodes simultaneously, and where there is an inherent delay in detecting the infection. Our experimental results show that the proposed algorithms are highly effective and scalable.

Keywords Diffusion in networks · Security · Stealthy diffusion · Monitoring diffusions · Malware detection

The preliminary version of this work appeared in the Proceedings of the 15th IEEE International Conference on Data Mining [12].

✉ Nika Haghtalab
nhaghtal@cs.cmu.edu

1 Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

2 Vanderbilt University, Nashville, TN 37235, USA

1 Introduction

In recent years, diffusion processes in social networks have been the focus of intense study [9,18,19,22,27]. Much of the work in this space considers diffusion as a desirable process, motivated by the study of viral marketing strategies, and seeks to maximize its reach by choosing the (near) optimal set of influential nodes. However, the same mathematical framework can also be applied to malicious diffusion processes. Indeed, the spread of computer worms—perhaps the most prominent malicious diffusion process—has been studied extensively using epidemic models [21,25]. Even though these models have been successfully used to analyze the spread of some real-world worms, such as the Code Red worm from 2001 [33], they do not consider a key aspect of malware: *stealth*. In practice, once a worm is detected, we can implement a number of effective countermeasures. For example, if we acquire a sample of a worm, we can use it to implement signature-based antivirus software. As another example, if we learn of the vulnerabilities exploited for propagation, we can patch them and effectively stop the worm. In the case of nontargeted worms, which try to infect as many computers as possible, stealth does not always play a crucial role, since it may be in conflict with the primary goal of maximizing impact. For example, the Code Red worm defaced the websites hosted by the webservers that it had infected, thereby immediately revealing its presence.

In contrast, recent years have seen the rise of highly targeted worms. For example, the Stuxnet worm targeted uranium-enrichment infrastructure in Iran, reportedly destroying one-fifth of the uranium centrifuges at the Natanz facility [17], while the Gauss worm was designed to spy on Lebanese banks, including Bank of Beirut and EBLF, but it also targeted users of Citibank and PayPal in the Middle East [16]. Even though these worms propagated in a non-deterministic manner, typically via USB flash drives and local area networks, they had very specific (sets of) targets (Fig. 1). In the case of these worms, stealth plays a key role, as the worm must remain covert until reaching its target in order to succeed.

Worms that can propagate over local networks and removable drives pose a serious threat to systems that are meant to be secured by the "air gap," that is, by not connecting them to the Internet or other public networks. In order to keep these systems safe, it is imperative that we detect worms *before* they reach their target. Consequently, systems must be continuously monitored for suspicious activities and anomalies. For example, we can monitor network connections originating from a system to detect when a worm connects to a remote command-and-control server, or use entropy analysis to find encrypted malware payload. However, since thorough monitoring of a system requires substantial resources and experts' time, we cannot monitor every system. Hence, we are faced with the problem of determining *which* systems to monitor.

1.1 Approach

We introduce a new model of *stealthy* diffusion with the goal of choosing a collection of nodes to monitor so as to maximize the probability that the malicious diffusion is detected before some high value asset is affected. We analyze the problem of monitoring stealthy diffusion as a game between two players, the *attacker* and the *defender*; we take the side of the defender. The game is defined on a known graph, with a distinguished *target node*. The attacker chooses a single seed node, and the defender selects k monitor nodes. Both the defender's and attacker's choices are restricted to subsets of network nodes (i.e., only nodes that are under their direct control, or, for the attacker, that could be directly attacked).

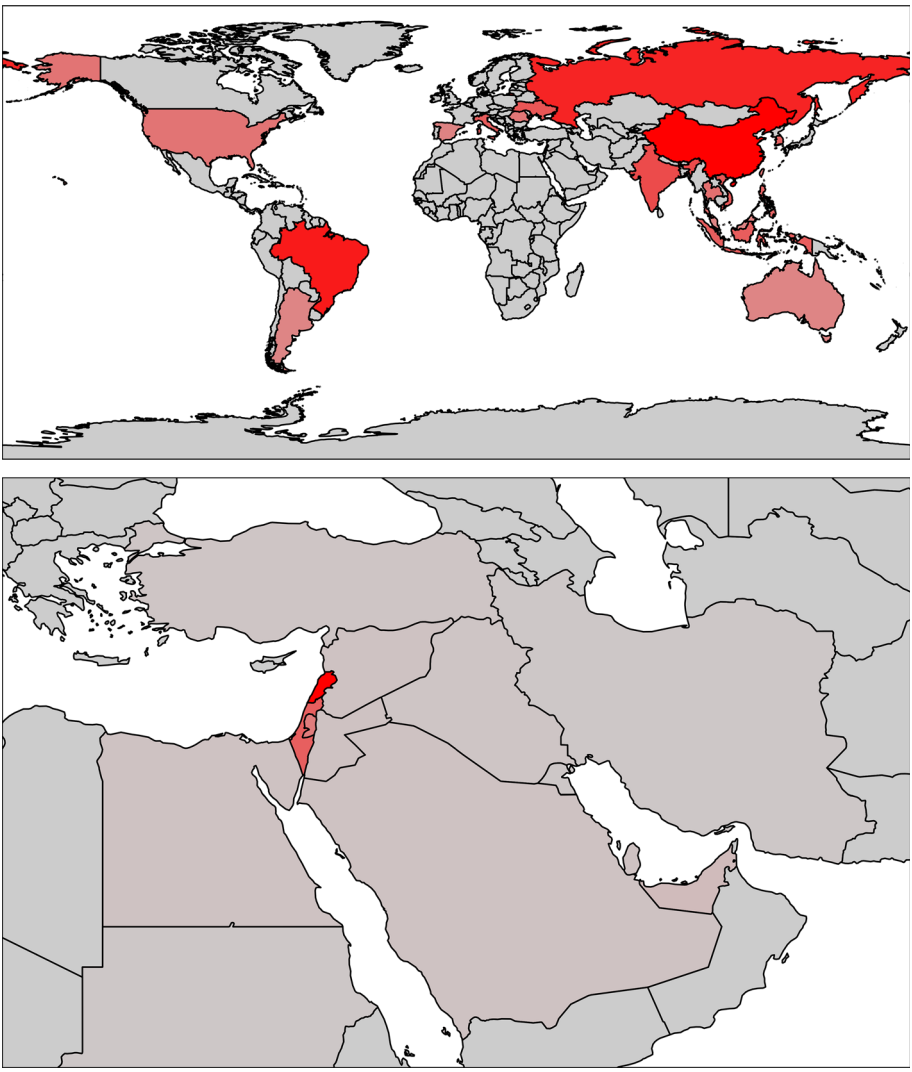


Fig. 1 Many worms, such as Conficker (*top*), spread so as to maximize the number of infections. Others, like Gauss (*bottom*), aim at specific targets, and deliberately try to avoid being detected, so that their spread is highly localized

The defender's utility is the probability that the diffusion process hits a monitor node before reaching the target.

Our model bears resemblance to recent work on competitive influence maximization [4,5,7,13,28,29,31]. However, our model is distinct in two respects: first, because it accounts for stealth in the attacker's primary objective, and second, because of the defender's focus on malware detection, rather than blocking.

We consider two design choices, with two options each:

1. *Diffusion process model* The two options here are the *independent cascade model* as described by Kempe et al. [18], and a variant of the independent cascade model where

each infected node repeatedly tries to infect its neighbors, until they are all infected. The latter model, which we call *repeated independent cascade*, provides a more realistic model for malicious diffusion, such as the spread of computer worms. We also find the repeated variant to be exciting on a conceptual level, since it considerably enriches the problem of monitoring the diffusion process in our setting, whereas it does not lead to meaningful problems in the classic influence maximization setting, as it is inevitable that all nodes will be infected eventually.

2. *Attacker power* In the *distributional* setting, the attacker does not respond to the defender's choice of monitors: We are given upfront a probability distribution over his choice of seed nodes. In the *maximin setting*, the (more powerful) attacker best-responds to any choice of monitors by minimizing the defender's utility, and the defender's goal is to maximize the minimum utility.

1.2 Results

Our theoretical results focus on choosing an approximately optimal set of monitors in polynomial time. Structure-wise, the results are split according to the attacker model (item 2 above), as this is the more significant factor. All the results below hold for both diffusion models.

In Sect. 3, we study the distributional setting. We present a polynomial-time algorithm that approximates the optimal solution to a factor of $1 - 1/e - o(1)$. We also show that this result is tight, by proving that it is NP-hard to approximate the problem to a factor of $1 - 1/e + o(1)$. These results are reminiscent of the classic results for influence maximization [18].

In Sect. 4, we study the maximin version of the problem, which turns out to be much more challenging. In fact, the problem is NP-hard to approximate to any factor, even when the defender's monitor budget is increased by a factor of $\ln |\mathcal{S}|$, where $\mathcal{S}$ is the set of possible seed nodes. On the positive side, we show that with an additional increase in the number of monitors—$|S|k \ln(1/\epsilon)$—we can achieve a $1 - \epsilon$ fraction of the optimum for k monitors, in polynomial time. We also establish a stronger result when the diffusion process is deterministic: $k \ln |\mathcal{S}|$ monitors suffice to do as well as the k-monitor optimum.

In Sect. 5, we discuss a generalization of our model and results to a setting where, like the defender, the attacker also has a budget b and selects b seed nodes to start the diffusion. We discuss the extent to which our results extend to this setting. In particular, we show that all of our results about the distributional setting readily extend to the case of multiple seed nodes. Whereas some of our guarantees for the maximin setting deteriorate when $b > 1$ is considered. We also discuss the problem of selecting the initial seed nodes from the attacker's point of view and demonstrate the hardness of such optimization. In Sect. 6, we discuss another generalization of our model that takes into account possible detection delays associated with monitors.

In Sect. 7, we test several algorithms on random graphs and the autonomous system relationship graph. We find that our approximation algorithm for the distributional setting is essentially optimal in practice. For the maximin setting, while our approximation algorithm is not far from optimal, we present two algorithms that are closer to optimal in practice, albeit without providing worst-case guarantees.

1.3 Related work

Multiple models have been suggested for studying diffusion processes [3,9,18,19,22,27,32]. One of the most well-studied models in this space is the *independent cascade model* of [18],

where the infection starts at one node and at every time step $t = 1, 2, \ldots$, any newly infected node gets a chance at infecting its neighbors. Diffusion can also be modeled as a *continuous-time* process, such as in the influential work of Bass [3] who used differential equations to describe a diffusion process over a continuous-time horizon. More recently, [32] introduced the *linear influence model* that models the global influence of any node on the rate of diffusion through an implicit network.

Previous papers have also used diffusion models to study epidemics and security aspects of a network structure. The *susceptible-infected-susceptible (SIS)* and *susceptible-infected-removed (SIR)* models describe the spread of viruses in a network under infecting and curing processes [11,30]. Diffusion models are also used in crime models for the purpose of physical security [23]. Another related setting is the deterministic and randomized pursuit-evasion games, where one or more cops move on a network in order to catch moving robbers [1,14,26]. Our work addresses inherently different problems and models than these works.

After the publication of the conference version of our results, we were made aware of a related existing work in the space of robust submodular optimization by Krause et al. [20]. In this work, the authors consider maximizing the minimum of n monotone submodular functions and show how one can recover OPT using an $O(\log(n))$ multiplicatively larger budget. As we will describe further in Sect. 4, we can reduce the problem of optimal monitoring in the maximin setting (more powerful attacker) to the robust optimization framework of [20].

2 Model

Our starting point is a model of diffusion (of viruses or malware) through a network from an initial set S of affected nodes. Importantly, in our theoretical results in Sects. 3 and 4, we assume that S is a singleton; we discuss the generalization to any number of seed nodes in Sect. 6.

Let $G = (V, E)$, with $|V| = n$ be a graph with a set of nodes V, and for simplicity assume that this graph is undirected. Each edge $(v, w) \in E$ is associated with a probability p_{vw} which captures the likelihood of direct diffusion from node v to its neighbor w. For two nodes $v, w \in V$, we use $d(v, w)$ to denote their shortest path distance in the graph. For a node $v \in V$ and integer d we use $\Gamma_d(v) = \{w \mid d(v, w) \leq d\}$ to denote the set of all nodes that are within distance d from v.

One natural model of diffusion that has commonly been considered in the past is known as the *independent cascade (IC)* model [18]. A set of *seed* nodes $S \subseteq V$ are infected at the beginning of the diffusion process. In each subsequent round, when a node first becomes infected it is active for exactly one round. Each active node $v \in V$ passes the infection to its uninfected neighbor $w \in V$ with probability p_{vw}, independently of previous rounds or neighbors. Note that in the independent cascade model, the diffusion process dies out after at most n rounds. In the context of cyber malware spread, the notion that an infected node can only spread malware to its neighbors once seems too limiting. We therefore also consider a natural extension, which we term the *repeated independent cascade (RIC)* model, in which infected nodes remain active in all subsequent rounds. Thus, every infected node v attempts to pass the infection to each uninfected neighbor w with probability p_{vw} *in every round*. We assume that for any edge $e \in E$, either $p_e = 0$ or $p_e \geq \rho$ for some $\rho \in \Omega(\frac{1}{\text{poly}(n)})$.

In most of the literature to date, given a diffusion process, the problem has been to choose a set of initial seed nodes $S \subseteq V$ so as to maximize the expected total number of nodes

infected in the network.[1] In cyber security, on the other hand, the attacker often has specific targets in mind, and it is crucial for the attacker to avoid detection. These two objectives are typically in conflict: greater spread of an infection increases the likelihood of reaching the target, but also increases the likelihood of being detected *before the target is reached*. To formalize this trade-off, let $M \subset V$ be a set of monitored nodes, which we call simply *monitors*, let $\mathcal{S} \subseteq V$ be a set of potential seed nodes (for example, nodes that can be reached by the attacker directly), and let $t \notin \mathcal{S}$ be the target of attack. The restriction that $t \notin \mathcal{S}$ is natural in cyber security: For example, sensitive data are often not located on workstations in regular use, but on servers available only behind a firewall (and usually not susceptible to direct phishing attacks); as another example, critical cyber-physical system infrastructure is often separated from the internet by the *air gap*, so that it cannot be attacked directly, but is susceptible to indirect infection (for example, through software updates).

In our model, the attacker seeds a single node $s \in \mathcal{S}$; see Sect. 5 for a generalization to the case of multiple seeds. For a given seed node s and a collection of monitors M, we define the attacker's utility as the probability that the target node t is infected before any monitoring node detects an infection. More formally, the attacker's utility is the probability that the infection reaches the target t before or at the same time as when the first monitor is infected. The defender's utility is the converse: the probability that an infection is detected prior to reaching the target t. We denote the corresponding defender's utility function by $U(M, s)$.

We consider two models of attacker behavior. In the first model, the attacker chooses $s \in \mathcal{S}$ using a known distribution $\mathcal{D}$ over $\mathcal{S}$. In this case, we are interested in the expected utility of the defender, that is, the probability that there exists $m \in M$ that is infected before t, where the probability is taken over the edge probabilities of G and the choice of S. We denote this by

$$\mathcal{U}(M) = \mathbb{E}_{s\sim\mathcal{D}}[U(M, s)],$$

where $\mathcal{U}(\cdot)$ denotes the utility function when seeds are chosen randomly.

In the second model, the attacker first observes the choice of monitors M, and then chooses a seed node $s \in \mathcal{S}$ that minimizes the defender's utility. We call this model the *maximin* model and denote the defender's utility by

$$\mathcal{V}(M) = \min_{s\in\mathcal{S}} U(M, s).$$

where $\mathcal{V}(\cdot)$ denotes the utility function when seeds are chosen in an adversarial way. In both attack models, the defender's goal is to choose a set of monitor nodes $M \subseteq \mathcal{M}$ to maximize the defender's utility, where $\mathcal{M}$ is the set of feasible monitoring locations and $|M| \leq k$ for a given budget k. We use OPT_k to denote an optimal selection of M for a given model and budget k.

3 Weak attackers: the distributional setting

In this section, we study the weaker attacker model, where a known distribution over seeds is given. This section's main result is a tight $1 - \frac{1}{e}$ approximation for the case where the attacker's seed node is drawn from a known distribution. Our algorithm proceeds by greedily choosing a set of k monitors based on their marginal gains, $\mathcal{U}(M \cup \{m\}) - \mathcal{U}(M)$. However,

[1] This goal is actually meaningless in the RIC model if a graph is connected, since all nodes will eventually be infected.

since the diffusion process is stochastic and can be unbounded, we cannot compute the exact value of $\mathcal{U}(M)$ directly—this problem is indeed #P-hard for the independent cascade model using a similar reduction to that of [6]. Instead, we estimate $\mathcal{U}(M)$ in two steps by $\mathcal{U}^\tau(M)$ and $\hat{\mathcal{U}}^\tau(M)$. Define $\mathcal{U}^\tau(M)$ to be the utility measured over the first τ time steps, i.e., the probability that the target is not reached before at least one monitor is infected, measured over the first τ time steps. We in turn estimate $\mathcal{U}^\tau(M)$ via $\hat{\mathcal{U}}^\tau(M)$ by running ℓ copies of the diffusion process up to time τ, and taking the average over the outcomes.

Algorithm 1 DISTRIBUTIONAL MONITORING

Input: $G, \mathcal{M}, k, \mathcal{S}, t$, attacker distribution $\mathcal{D}$ over choice of seeds $\mathcal{S}$, and $\delta, \epsilon > 0$.

1. Let $\ell \leftarrow \frac{8k^2}{\epsilon^2} \ln(\frac{2k|\mathcal{M}|}{\delta})$ and $\tau \leftarrow \frac{n}{\rho} \ln(\frac{4kn}{\epsilon})$.
2. Start with $M \leftarrow \emptyset$.
3. For $i = 1, \ldots, k$ do
   (a) Let $m \in \mathcal{M}$ be a node that maximizes the marginal gain $\hat{\mathcal{U}}^\tau(M \cup \{m\}) - \hat{\mathcal{U}}^\tau(M)$, where the simulation is taken over ℓ samples.
   (b) Set $M \leftarrow M \cup \{m\}$.

Output: Set of monitors M.

Like [18], to establish the approximation guarantee of this algorithm, we rely on the celebrated result of [24] on optimizing *monotonically non-decreasing submodular* functions. A function F defined over a set S is said to be *submodular* if $F : 2^S \to \mathbb{R}^+$ satisfies a natural diminishing returns property: The marginal gain from adding an element to $T \subset S$ is at least the marginal gain from adding that element to any superset of T. More formally, for any $T \subset T' \subset S$, and any $s \notin T'$,

$$F(T \cup \{s\}) - F(T) \geq F(T' \cup \{s\}) - F(T').$$

Function F is furthermore *monotonically non-decreasing*, if for all s and $T \subseteq S$, $F(T \cup \{s\}) \geq F(T)$. Consider the problem of choosing $T \subseteq S$ with k elements that maximizes the value of $F(\cdot)$. While this problem is NP-hard in general, Nemhauser et al. [24] show that a simple greedy algorithm that builds T by repeatedly adding an element with the maximum marginal gain achieves a $(1 - \frac{1}{e})$ approximation. We use this result to prove the main theorem of this section.

Theorem 1 *For any $\epsilon, \delta > 0$, Algorithm 1 runs in time* $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\rho}, \log(\frac{1}{\delta}))$ *and returns a set $M \subseteq \mathcal{M}$, such that $|M| = k$, and with probability $1 - \delta$*

$$\mathcal{U}(M) \geq \left(1 - \frac{1}{e}\right) \mathcal{U}(OPT_k) - \epsilon.$$

This guarantee holds under both the IC and RIC models.

Below we prove the theorem for the RIC model. A similar (and slightly simpler) approach with different parameters also works for the IC model. We omit the modified proof due to space constraints.

The next lemmas first show that $\mathcal{U}(\cdot)$ is a monotonically non-decreasing submodular function, and furthermore, for the choice of parameters in the algorithm, $\hat{\mathcal{U}}^\tau(\cdot) \approx \mathcal{U}(\cdot)$. Putting these together, we show that the greedy algorithm finds a set that has utility at least $(1 - \frac{1}{e})\,\mathcal{U}(OPT_k) - \epsilon$.

Lemma 2 *$\mathcal{U}(\cdot)$ is monotonically non-decreasing and submodular over the set of monitor nodes.*

Proof Consider the outcome of the infection process to be a partial ordering between the set of nodes in the order that they are infected. For ordered partition σ, let $\Pr(\sigma)$ indicate the probability of partition σ occurring, taken over the choice of seed node from $\mathcal{D}$ and the outcomes of edge activations. For a given partial ordering σ and choice of monitor nodes M, let $f_\sigma(M) = 1$ if there is a monitor $m \in M$ that is infected in σ before t. Then,

$$\mathcal{U}(M) = \sum_\sigma f_\sigma(M) \Pr(\sigma).$$

Since a nonnegative linear combination of submodular functions is also submodular, to show that $\mathcal{U}(\cdot)$ is submodular it suffices to show that for any σ, $f_\sigma(\cdot)$ is submodular over set monitor nodes. Take any partial ordering σ , $M_1 \subset M_2$, and $m' \notin M_2$. There are two cases.

Case 1: There exists $m \in M_2$ that is infected before t in σ. Then, adding m' to M_2 does not produce any gain. So, $f_\sigma(M_1 \cup \{m'\}) - f_\sigma(M_1) \geq 0 = 1 - 1 = f_\sigma(M_2 \cup \{m'\}) - f_\sigma(M_2)$.
Case 2: No $m \in M_2$ exists that is infected before t. Then, adding m' to M_1 and M_2 has the same effect. So, $f_\sigma(M_1 \cup \{m'\}) - f_\sigma(M_1) = f_\sigma(M_2 \cup \{m'\}) - f_\sigma(M_2)$.

As shown above, the marginal gain of each element is nonnegative; therefore, $\mathcal{U}(\cdot)$ is also monotonically non-decreasing. □

The next lemma shows that for the choice of parameter τ in the algorithm, $\mathcal{U}^\tau(\cdot) \approx \mathcal{U}(\cdot)$. At a high level, to prove this we first show that after a large enough number of time steps, every edge in an s-t path is activated and t is infected with a high probability. Since the probability that t is not infected by time step τ is small, one can ignore the utility of the player in this case while only introducing a small change in the overall utility. More details of this analysis are given below.

Lemma 3 *For any ϵ, let $\tau = \frac{n}{\rho} \ln(\frac{n}{\epsilon})$. Then, $|\mathcal{U}(M) - \mathcal{U}^\tau(M)| \leq \epsilon$.*

Proof Any s-t path has at most n edges, each succeeding with probability at least ρ. For each edge, after $\tau' = \frac{1}{\rho} \ln(\frac{n}{\epsilon})$ time steps, the probability that the edge is not activated is equal to the probability that τ' independent attempts fail to activate the edge, which is at most $(1-\rho)^{\tau'} \leq e^{-\rho\tau'} = \frac{\epsilon}{n}$, where the first inequality comes from the fact that $1 - x \leq e^{-x}$ for all $x \in [0, 1]$. Then, t will be activated in the first $\tau = n\tau'$ time steps, with probability at least $1 - \epsilon$.

Let A be the event that t is infected by round τ, and $\bar{A}$ to be its complement. By the above argument, $\Pr(\bar{A}) \leq \epsilon$. Let $\mathcal{U}(M|A)$ indicate the utility $\mathcal{U}(M)$ of the set M conditioned on the event A. That is, $\mathcal{U}(M|A)$ is the probability that a monitor is infected before the target, given that the target is infected in the first τ steps. Define $\mathcal{U}^\tau(M|A), \mathcal{U}(M|\bar{A})$ and $\mathcal{U}^\tau(M|\bar{A})$ similarly. By this definition, $\mathcal{U}^\tau(M|A) = \mathcal{U}(M|A)$. On the other hand, if the target is not reached within the first τ steps, then $\mathcal{U}^\tau(M|\bar{A}) = 1$. So, $\mathcal{U}^\tau(M|\bar{A}) \geq \mathcal{U}(M|\bar{A})$. It follows that

$$\begin{aligned}\mathcal{U}^\tau(M) &= \mathcal{U}^\tau(M|A)\Pr(A) + \mathcal{U}^\tau(M|\bar{A})\Pr(\bar{A}) \\ &\geq \mathcal{U}(M|A)\Pr(A) + \mathcal{U}(M|\bar{A})\Pr(\bar{A}) \\ &= \mathcal{U}(M),\end{aligned}$$

and

$$\begin{aligned}\mathcal{U}^\tau(M) &= \mathcal{U}^\tau(M|A)\Pr(A) + \mathcal{U}^\tau(M|\bar{A})\Pr(\bar{A}) = \mathcal{U}(M|A)\Pr(A) + \Pr(\bar{A}) \\ &\leq \mathcal{U}(M) + \epsilon.\end{aligned}$$

Putting the above two inequalities together we have $|\mathcal{U}(M) - \mathcal{U}^\tau(M)| \leq \epsilon$. □

The next lemma shows that $\mathcal{U}^\tau(M) \approx \hat{\mathcal{U}}^\tau(M)$ when the estimation is done by running ℓ copies of the diffusion process, for a large enough value of ℓ.

Lemma 4 *For any $\epsilon, \delta > 0$ and M, let $\hat{\mathcal{U}}^\tau(M)$ be the average of $\ell = \frac{1}{2\epsilon^2}\ln(\frac{2}{\delta})$ simulations of $\mathcal{U}^\tau(M)$. With probability at least $1-\delta$,*

$$\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}^\tau(M)\right| \leq \epsilon.$$

Proof We estimate the probability that the target is not reached before a monitor is infected, in the first τ time steps, using $\ell = \ln(\frac{2}{\delta})\frac{1}{2\epsilon^2}$ simulations. The outcome of each simulation is a random variable X_i with expectation $\mathcal{U}^\tau(M)$. Using Hoeffding's inequality we have

$$\begin{aligned}\Pr\left[\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}^\tau(M)\right| \geq \epsilon\right] &= \Pr\left[\left|\frac{1}{\ell}\sum_{i=1}^{\ell} X_i - \mathbb{E}\left[\frac{1}{\ell}\sum_{i=1}^{\ell} X_i\right]\right| \geq \epsilon\right] \\ &\leq 2e^{-2\ell\epsilon^2} \leq \delta.\end{aligned}$$

□

We are now ready to prove the theorem.

Proof of Theorem 1 Recall from Algorithm 1 that $\ell = \frac{8k^2}{\epsilon^2}\ln(\frac{2k|\mathcal{M}|}{\delta})$ and $\tau = \frac{n}{\rho}\ln(\frac{4kn}{\epsilon})$.

The algorithm takes k rounds, and at each round estimates the utility of $O(|\mathcal{M}|)$ monitors. By Lemma 4, for each of these estimates, with probability $1 - \frac{\delta}{k|\mathcal{M}|}$, we have $\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}^\tau(M)\right| \leq \epsilon/(4k)$. So, with probability $1-\delta$, all the estimates $\hat{\mathcal{U}}^\tau(\cdot)$ used in the algorithm are within $\epsilon/4$ of their respective $\mathcal{U}^\tau(\cdot)$. Using Lemma 3, this is within $\epsilon/(4k)$ of $\mathcal{U}(\cdot)$. Therefore, $|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}(M)| \leq \epsilon/(2k)$ for all M considered by the greedy algorithm.

The $(1-\frac{1}{e})\,\mathcal{U}(OPT_k) - \epsilon$ guarantee then follows by applying the result of [24] (described above) for optimizing submodular functions, and observing that at each of the k steps of Algorithm 1, which uses estimates of the utilities, the true marginal utility of the chosen monitor differs from the choice the exact greedy algorithm would have made *at this round* by at most ϵ/k. So, at each step the true contribution of the node chosen at that step is close to the contribution of node with the best marginal gain. We conclude that after k estimated greedy choices the outcome has a utility that differs from the exact greedy solution, which has value $(1-\frac{1}{e})\,\mathcal{U}(OPT_k)$, by at most ϵ.[2] □

Next we provide a matching hardness result to complement Theorem 1. This hardness result is obtained through a reduction from the MAX- COVER problem.

Theorem 5 *Finding a $(1-\frac{1}{e}+o(1))$-approximately optimal monitor set is NP-hard under the IC and RIC models. That is, it is NP-hard to find a set $M \subseteq \mathcal{M}$ such that $|M| \leq k$ and*

$$\frac{\mathcal{U}(M)}{\mathcal{U}(\mathrm{OPT}_k)} > 1 - \frac{1}{e}.$$

This is true even if $\mathcal{D}$ has singleton support.

Proof We present a reduction from the search version of the MAX- COVER problem: Given a set of elements U, a collection of its subsets $A \subseteq 2^U$, and a budget k such that there exists

[2] Proof of Theorem 7 formalizes this argument for a more general optimization problem discussed in the future section.

a subset of A with size k that covers all the elements U, it is NP-hard to find a subset of A of size k that covers more than $1 - \frac{1}{e}$ fraction of U [15].

We create a graph $G = (V, E)$ as follows. V includes one vertex per $a \in A$, one vertex per $u \in U$, the deterministic seed node s (which has probability 1 under $\mathcal{D}$), the target t, and two additional vertices v_1 and v_2 (see Fig. 2). The set of edges and their corresponding probabilities are as follows.

$$E = \left\{ \begin{array}{lll} e : au & \forall a \in A, u \in U, \quad \text{s.t. } u \in a & p_e = 1 \\ e : su & \forall u \in U & p_e = \frac{1}{|U|^2} \\ e : sv_2, v_1v_2, v_1t & & p_e = 1 \end{array} \right\}$$

This graph is an instance of the targeted diffusion problem with monitor set $\mathcal{M}$ corresponding to nodes in A, s being the attacker seed node, and t being the target node.

Let M' be the choice of monitor nodes that correspond to a k-cover of (U, A) and OPT_k be the optimal set of k monitors. Since there is a path of length 3 between s to t that consists of edges with probability 1, target t is certainly infected at time step 3 if a monitor is not infected earlier. So, the utility of M' is the probability that at least one of the nodes in U is infected in the first time step (and as result one monitor becomes infected in the second time step). Then, the utility of M' is the probability of the complement of the event where none of the members of U are infected in the first step. Letting $|U| = m$, we have

$$\mathcal{U}(\text{OPT}_k) \geq \mathcal{U}(M') = 1 - \left(1 - \frac{1}{m^2}\right)^m.$$

Let $M \subseteq \mathcal{M}$ be any monitor set and let α be the fraction of the elements of U that are adjacent to some member of M, i.e., $|\Gamma(M)| = \alpha m$ is the size of the neighborhood of M in U. The utility of the defender for choosing M is the probability that at least one of the nodes in $\Gamma(M)$ is infected in the first time step. Therefore,

$$\mathcal{U}(M) = 1 - \left(1 - \frac{1}{m^2}\right)^{\alpha m}.$$

We have

$$\begin{aligned} \lim_{m\to\infty} \frac{\mathcal{U}(M)}{\mathcal{U}(M')} &= \frac{1 - \left(1 - \frac{1}{m^2}\right)^{\alpha m}}{1 - \left(1 - \frac{1}{m^2}\right)^{m}} \\ &= \lim_{m\to\infty} \frac{-\left(1 - \frac{1}{m^2}\right)^{\alpha m} \left(\frac{2\alpha}{\left(1-\frac{1}{m^2}\right)m^2} + \alpha \log\left(1 - \frac{1}{m^2}\right)\right)}{-\left(1 - \frac{1}{m^2}\right)^{m} \left(\frac{2}{\left(1-\frac{1}{m^2}\right)m^2} + \log\left(1 - \frac{1}{m^2}\right)\right)} \\ &= \lim_{m\to\infty} \frac{\alpha \log\left(1 - \frac{1}{m^2}\right)}{\log\left(1 - \frac{1}{m^2}\right)} = \alpha, \end{aligned}$$

where the second equality follows by the application of L'Hospital's rule. So, if $\frac{\mathcal{U}(M)}{\mathcal{U}(M')} > 1 - \frac{1}{e}$, then $|\Gamma(M)| > (1 - \frac{1}{e})m$. This implies that a polynomial-time algorithm produces a $(1 - \frac{1}{e})$-approximation for any MAX- COVER instance, which contradicts the hardness of $(1 - \frac{1}{e})$-approximation for MAX- COVER. □

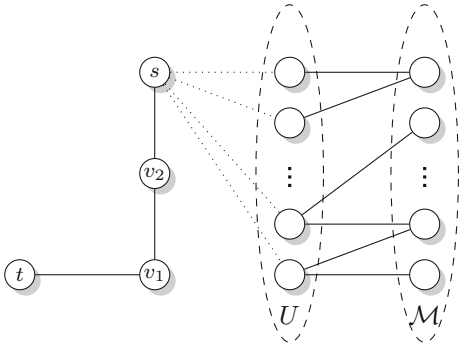


Fig. 2 Illustration of the construction used in the proof of Theorem 5. All *solid edges* have probability 1, and all *dotted edges* have probability $1/|U|^2$

4 Powerful attackers: the maximin setting

We next tackle more powerful attackers that observed the defender's choice of monitors (for example, when such a choice is made public) and best-respond to it. The defender's goal is then to choose a set of monitors M that maximizes $\mathcal{V}(M) = \min_{s \in \mathcal{S}} U(M, s)$.

Our first result is negative: We show that it is NP-hard to find a set of $(1 - o(1))k \ln(|\mathcal{S}|)$ monitor nodes with nonzero utility even when OPT_k has utility 1. That is, the targeted diffusion problem is hard to approximate to any factor even when the given budget is significantly larger.

This hardness result follows by a reduction from the MIN- SET- COVER problem. At a high level, we embed a MIN- SET- COVER instance between the set of possible monitor nodes $\mathcal{M}$, and possible attacker seed nodes $\mathcal{S}$, such that the optimal solution covers $\mathcal{S}$ fully and achieves utility 1 (see Fig. 3). All possible seed nodes are connected to the target, so a seed node that is not covered by a monitor will infect the target before the infection is detected by other monitors. Therefore, any suboptimal choice of monitors leads to a utility of 0. The details of this approach are described below.

Theorem 6 *For any $\epsilon > 0$, it is NP-hard under the IC and RIC models to find a set $M \subseteq \mathcal{M}$ such that $|M| \leq (1 - \epsilon) \ln(|S|)k$, and*

$$\frac{\mathcal{V}(M)}{\mathcal{V}(\text{OPT}_k)} > 0.$$

This is true even if the diffusion process is deterministic, that is, $\rho = 1$.

Proof We reduce from the search version of the MIN- SET- COVER problem: Given a set of elements U, a collection of its subsets $A \subseteq 2^U$, and k such that we are promised that there exists a subset of A with size k that covers all the elements of U, for any $\epsilon > 0$, it is NP-hard to find a subset of A of size $(1 - \epsilon)k \ln(|U|)$ that covers U [8].

Let (U, A) be an instance of MIN- SET- COVER with the promise that there exists a subset of A of size k that covers all the elements U. We create a graph $G(V, E)$ as shown in Fig. 3. V includes one vertex per $a \in A$, one vertex per $u \in U$, the target t, and an additional vertex v. E includes one edge as for every $a \in A$ and $u \in U$ such that $u \in a$. Furthermore, E has an edge vu for all $u \in U \cup \{t\}$. All edges have probability 1 (so the IC and RIC models are equivalent in the context of this construction).

Consider the maximin targeted diffusion problem with the set of possible monitors $\mathcal{M}$ corresponding to the set of nodes in A, set of possible attacker seed nodes $\mathcal{S}$ corresponding to the set of nodes in U, and t being the target node. Let OPT_k denote the optimal set cover for

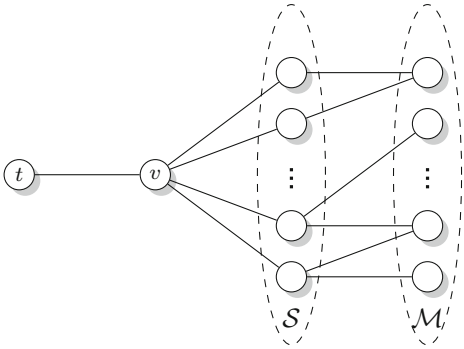


Fig. 3 Illustration of the construction used in the proof of Theorem 6. All edges have probability 1

(U, A). Then $\mathcal{V}(\text{OPT}_k) = 1$, because whichever node in $\mathcal{S}$ the attacker chooses, it is covered by some monitor, which is reached in one step (whereas it takes two steps to reach t).

Assume on the contrary that there is a polynomial-time algorithm for finding a set $|M| \leq (1-\epsilon)\ln(|S|)k$ such that $\mathcal{V}(M) > 0$. Since, all the edge probabilities are 1, this implies that $\mathcal{V}(M) = 1$. If $\Gamma(M) \subsetneq \mathcal{S}$, then the attacker could choose any $u \in \mathcal{S} \backslash \Gamma(M)$ as the seed node and successfully attack the target with probability 1, leading to $\mathcal{V}(M) = 0$. Therefore, $\Gamma(M) = \mathcal{S}$. But, this shows that there is a polynomial-time algorithm that approximates set cover within $(1-\epsilon)\ln(|U|)$, which contradicts the hardness result stated above. □

Next, we show that it is possible to achieve $1-\epsilon$ multiplicative factor approximation of $\mathcal{V}(\text{OPT}_k)$ using at most $|\mathcal{S}|k\ln(1/\epsilon)$ monitors. For a seed node s, let $\mathcal{U}_s(\cdot)$ represent the utility function *when the attacker deterministically selects* s. Algorithm 2 informally proceeds as follows: For each seed node s, individually, choose $k\ln(1/\epsilon)$ monitors greedily based on their estimated marginal gain with respect to $\mathcal{U}_s(\cdot)$ and store them in a set $M(s)$. The algorithm then returns $\bigcup_{s\in\mathcal{S}} M(s)$.

Algorithm 2 MAXMIN MONITORING

Input: $G, \mathcal{M}, k, \mathcal{S}, t$ and $\delta, \epsilon, \gamma > 0$.

1. Let $\ell \leftarrow \frac{36k^2\ln^2(1/\epsilon)}{\gamma^2}\ln\left(\frac{\delta}{2|\mathcal{S}|\cdot|\mathcal{M}|k\ln(1/\epsilon)}\right)$ and $\tau \leftarrow \frac{n}{\rho}\ln(\frac{8nk\ln(1/\epsilon)}{\gamma})$.
2. For all $s \in \mathcal{S}$, do
   (a) Set $M(s) \leftarrow \emptyset$.
   (b) For all $i = 1, \ldots, k\log(\frac{1}{\epsilon})$: Let $m_i \in \mathcal{M}$ be a node that maximizes the estimated marginal gain $\hat{\mathcal{U}}_s^{\tau}(M(s)\cup\{m_i\}) - \hat{\mathcal{U}}_s^{\tau}(M(s))$, where the simulation is taken over ℓ tries up to τ time steps. Set $M(s) \leftarrow M(s)\cup\{m_i\}$.
   (c) $M \leftarrow M \cup M(s)$.

Output: Set of monitors M.

Theorem 7 *For any maximin targeted diffusion instance, any* $k, \epsilon > 0, \gamma > 0$ *and* $\delta > 0$, *Algorithm 2 runs in time* $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\gamma}, \frac{1}{\rho}, \log(\frac{1}{\delta}))$ *and finds a set* $|M| \leq |\mathcal{S}|k\ln(1/\epsilon)$ *such that with probability* $1-\delta$, $\mathcal{V}(M) \geq (1-\epsilon)\,\mathcal{V}(OPT_k) - \gamma$. *This guarantee holds under both the IC and RIC models.*

As before, we prove the theorem for the more difficult RIC model; modifying the proof for the IC model is an easy exercise.

Proof Let OPT_k represent the optimal set of k monitor nodes for the maximin utility $\mathcal{V}(\cdot)$. For a seed node s, let $\text{OPT}_k(s)$ represent the optimal set of k monitors *when the attacker deterministically selects* s. Then for all $s \in \mathcal{S}$, $\mathcal{V}(\text{OPT}_k) \leq \mathcal{U}_s(\text{OPT}_k(s))$.

To prove the claim, it suffices to show that for any s, when we choose $M(s)$ using $k\ln(1/\epsilon)$ greedy selections of monitors, we have,

$$\mathcal{U}_s(M(s)) \geq (1-\epsilon)\,\mathcal{U}_s(\text{OPT}_k(s)) - \gamma, \tag{1}$$

and as a result,

$$\begin{aligned}\mathcal{V}\left(\bigcup_s M(s)\right) &\geq \min_s\ \mathcal{U}_s(M(s)) \geq \min_s\ (1-\epsilon)\,\mathcal{U}_s(\text{OPT}_k(s)) - \gamma \\ &\geq (1-\epsilon)\,\mathcal{V}(\text{OPT}_k) - \gamma.\end{aligned}$$

Hereinafter, we focus on establishing Eq. (1). For ease of notation, we suppress s in $\mathcal{U}_s(\cdot)$ and $M(s)$ and represent them, respectively, by $\mathcal{U}(\cdot)$ and M. Let $\xi = \frac{\gamma}{2k\ln(1/\epsilon)}$.

For a fixed M and

$$\ell = \frac{8}{\xi^2}\log(\delta/(2|\mathcal{S}|\cdot|\mathcal{M}|k\ln(1/\epsilon)))$$

simulations up to time step $\tau = \frac{n}{\rho}\ln(4nk\log(1/\epsilon)/\epsilon)$, using Hoeffding's inequality we have

$$\Pr\left[\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}^\tau(M)\right| \geq \frac{\xi}{4}\right] \leq 2e^{-\ell\xi^2/8} \leq \frac{\delta}{|\mathcal{S}|\cdot|\mathcal{M}|k\ln(1/\epsilon)}.$$

A total of $|\mathcal{S}|\cdot|\mathcal{M}|k\ln(1/\epsilon)$ sets are considered by the algorithm, so with probability $1-\delta$, for any M considered by the algorithm, we have $\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}^\tau(M)\right| \leq \xi/4$. Additionally, by Lemma 3, $|\mathcal{U}^\tau(M) - \mathcal{U}(M)| \leq \xi/4$. Therefore, with probability $1-\delta$, for any M considered by the algorithm, we have $\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}(M)\right| \leq \xi/2$.

Let us introduce additional notations to help with the proof. For any set M and monitor m, let $g_M(m) = \mathcal{U}(M\cup m) - \mathcal{U}(M)$ be the marginal utility of m with respect to the set M. Similarly, let $\hat{g}^\tau_M(m) = \hat{\mathcal{U}}^\tau(M\cup m) - \hat{\mathcal{U}}^\tau(M)$. Then, with probability $1-\delta$, for any M and m considered by the algorithm, we have $|\hat{g}^\tau_M(m) - g_M(m)| \leq \xi$.

Next, for any $i \leq k\ln(1/\epsilon)$, let $M_i = \bigcup_{j\leq i} m_j$ be the set of monitors that have been chosen by the greedy algorithm up to and including step i for the seed node s. We prove by induction that

$$\mathcal{U}(\text{OPT}_k(s)) - \mathcal{U}(M_i) \leq \left(1-\frac{1}{k}\right)^i \mathcal{U}(\text{OPT}_k(s)) - 2i\xi.$$

For the case of $i=0$, the claim holds trivially. Assume that this claim holds for $i-1$. At step i, m_i is chosen such that $m_i = \arg\max_m \hat{g}^\tau_{M_{i-1}}(m)$. So in particular, m_i has higher estimated marginal utility than any monitor in the set $\text{OPT}_k(s)\backslash M_{i-1}$. If $\text{OPT}_k(s)\backslash M_{i-1} = \emptyset$, then we have already achieved utility of at least $\text{OPT}_k(s)$ and the claim holds trivially. If not, then $0 < |\text{OPT}_k(s)\backslash M_{i-1}| \leq k$. So,

$$\hat{g}^\tau_{M_{i-1}}(m_i) \geq \frac{\sum_{m\in\text{OPT}_k(s)\backslash M_{i-1}} \hat{g}^\tau_{M_{i-1}}(m)}{|\text{OPT}_k(s)\backslash M_{i-1}|}.$$

Therefore,

$$g_{M_{i-1}}(m_i) \geq \frac{1}{k}\sum_{m\in\text{OPT}_k(s)\backslash M_{i-1}} g_{M_{i-1}}(m) - 2\xi. \tag{2}$$

On the other hand, using submodularity, we have that

$$\mathcal{U}(\mathrm{OPT}_k(s)) - \mathcal{U}(M_{i-1}) \leq \sum_{m \in \mathrm{OPT}_k(s) \setminus M_{i-1}} g_{M_{i-1}}(m),$$

So, using this in conjunction with Eq. (2), we get

$$g_{M_{i-1}}(m_i) \geq \frac{1}{k} \left(\mathcal{U}(\mathrm{OPT}_k(s)) - \mathcal{U}(M_{i-1})\right) - 2\xi.$$

It follows that

$$\begin{aligned}
\mathcal{U}(\mathrm{OPT}_k(s)) - \mathcal{U}(M_i) &= \mathcal{U}(\mathrm{OPT}_k(s)) - \mathcal{U}(M_{i-1}) - g_{M_{i-1}}(m_i) \\
&\leq \left(1 - \frac{1}{k}\right) \left(\mathcal{U}(\mathrm{OPT}_k(s)) - \mathcal{U}(M_{i-1})\right) + 2\xi \\
&\leq \left(1 - \frac{1}{k}\right)^i \mathcal{U}(\mathrm{OPT}_k(s)) + 2(i-1)\xi + 2\xi \\
&\leq \left(1 - \frac{1}{k}\right)^i \mathcal{U}(\mathrm{OPT}_k(s)) + 2i\xi.
\end{aligned}$$

Therefore, after $i = k \ln(1/\epsilon)$ rounds and replacing $\xi = \frac{\gamma}{2k \ln(1/\epsilon)}$, we get $\mathcal{U}_s(M(s)) \geq (1-\epsilon)\, \mathcal{U}_s(\mathrm{OPT}_k(s)) - \gamma$. So, with probability $1 - \delta$, $\mathcal{V}(M) \geq (1-\epsilon)\, \mathcal{V}(\mathrm{OPT}_k) - \gamma$. □

Our final theoretical result states that if the diffusion process is deterministic (case of $\rho = 1$), then $k \ln(|\mathcal{S}|)$ monitor nodes are sufficient to find the optimal solution. Note that by the $(1-\epsilon) \ln(|\mathcal{S}|)k$ lower bound of Theorem 6, which holds even for the $\rho = 1$ case, this is the smallest number of monitors needed to guarantee a nonzero utility.

The idea behind our Algorithm, presented below as Algorithm 3, is to choose monitors in a way as to "cover" the set of all possible seed nodes. Specifically, for each possible seed node $s \in \mathcal{S}$ and candidate monitor node $m \in \mathcal{M}$, we say that *m covers s* if m is successful at monitoring the diffusion process starting from s, i.e., the deterministic diffusion process starting at s infects m before it infects the target. Our algorithm then constructs an equivalent set cover instance for an instance of a deterministic diffusion problem and greedily finds a set cover of size $k \ln(|\mathcal{S}|)$.

Algorithm 3 MAXIMIN MONITORING WITH $\rho = 1$

Input: $G, \mathcal{M}, k, \mathcal{S}, t$.

1. For all $s \in \mathcal{S}$ create the set $\Gamma_{d(s,t)-1}(s)$.
2. Create a set cover instance $(\mathcal{S}, \mathcal{M})$, where for the element corresponding to $s \in S$ and the set corresponding to $m \in \mathcal{M}$, $s \in m$ if and only if $m \in \Gamma_{d(s,t)-1}$. See Fig. 4 for an example.
3. Greedily find a set cover $M \subseteq \mathcal{M}$ for $(\mathcal{S}, \mathcal{M})$.

Output: Set of monitors M.

Theorem 8 *For any maximin targeted diffusion instance with $\rho = 1$ and for any k, Algorithm 3 runs in polynomial time in n and finds a set $|M| \leq k \ln(|\mathcal{S}|)$ such that $\mathcal{V}(M) = \mathcal{V}(\mathrm{OPT}_k)$.*

Proof Since $\rho = 1$, all edges in the instance have probability 1 and the diffusion process is deterministic. Therefore, for any k, $\mathcal{V}(\mathrm{OPT}_k) \in \{0, 1\}$. In the case of $\mathcal{V}(\mathrm{OPT}_k) = 0$, the theorem holds trivially. Hence, we focus on the case of $\mathcal{V}(\mathrm{OPT}_k) = 1$.

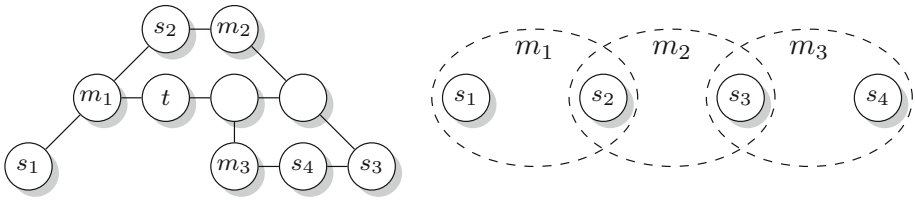


Fig. 4 An illustration of Step 2 of Algorithm 3. In the example, $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, and $\mathcal{M} = \{m_1, m_2, m_3\}$. The given graph is on the *left*, and the constructed set cover instance is on the *right*

First, we show that there is a one-to-one and onto mapping between set covers of $(\mathcal{S}, \mathcal{M})$ and a monitor sets with utility 1. For any monitor set M such that $\mathcal{V}(M) = 1$, consider the collection of sets that correspond to M; with abuse of notation we also call this M. Since, $\mathcal{V}(M) = 1$, for every choice of attacker seed nodes $s \in \mathcal{S}$, there exists a monitor $m \in M$, such that $d(s, m) < d(s, t)$, i.e., the monitor m is infected before target t. Therefore, for such m, we have $m \in \Gamma_{d(s,t)-1}(s)$. It follows that the collection of sets that correspond to the choice of monitors in M forms a set cover for $(\mathcal{S}, \mathcal{M})$. Conversely, for any set cover M for $(\mathcal{S}, \mathcal{M})$, consider the set of monitor nodes that correspond to M; with abuse of notation we also call this M. Since M is a set cover, for all $s \in \mathcal{S}$ there exists a set $m \in M$ such that $s \in M$. Consider the corresponding nodes s and m in the diffusion instance. This means that $m \in \Gamma_{d(s,t)-1}(s)$. So, if s is the seed node, m gets is infected before t. Therefore, for every choice of attacker seed node $s \in \mathcal{S}$, there is a monitor in M that is infected before the target, so $\mathcal{V}(M) = 1$.

It therefore suffices to show that the greedy set cover algorithm produces a set cover of size at most $k \ln(|\mathcal{S}|)$. This is a well-known fact. Here, we provide a simple proof of this fact for completeness. Since there is a one-to-one mapping between the set covers and monitor sets with utility 1, there is a set cover of size k for $(\mathcal{S}, \mathcal{M})$. Therefore, there must be a set that covers at least $\frac{|\mathcal{S}|}{k}$ of the points. The greedy procedure chooses this largest set, so there are at most $|\mathcal{S}|(1 - \frac{1}{k})$ uncovered elements left after the first greedy choice. Similarly, since the optimal algorithm uses at most k sets to cover the remaining uncovered nodes after step $i - 1$, there must be a set that covers $\frac{1}{k}$ of the remaining elements. So, there are at most $|\mathcal{S}|(1 - \frac{1}{k})^i$ elements left after the i^{th} greedy choice. After $i = k \ln(|\mathcal{S}|)$ greedy choices, there are $|\mathcal{S}|(1 - \frac{1}{k})^{k \ln |\mathcal{S}|} < 1$ uncovered elements in $\mathcal{S}$. We conclude that there is a set cover of size $k \ln(|\mathcal{S}|)$. This corresponds to a monitor set of size $k \ln(|\mathcal{S}|)$ with utility 1. □

The idea of "covering" the seeds nodes, used in this algorithm, leads to heuristic algorithms for diffusion processes that are not deterministic (general ρ). Even though the theoretical guarantees of the above algorithm do not extend to the case of general diffusion processes, the smaller number of monitor nodes required by this algorithm (Theorem 8), compared to the larger number of monitor nodes required by Algorithm 2, motivates experimental study of algorithms that attempt to greedily "cover" the set of seed nodes even when $\rho < 1$. We discuss these algorithms in Sect. 7.

After the publication of the conference version of our results, we were made aware of a related existing work in the space of robust submodular optimization by Krause et al. [20]. In this work, the authors consider maximizing the minimum of n monotone submodular function, $F_i : 2^X \rightarrow [0, 1]$, i.e., $\max_{A \subseteq X} \min_i F_i$ subject to $|A| \leq k$. They show that by using $O(\log(n))$ multiplicatively larger budget, they can recover OPT. We can view the problem of monitoring in the maximin setting through the lens of robust optimization by considering each function $U(\cdot, s)$ to be the monotone submodular function representing the

defender's utility under the condition that s is used as the seed node. In this case, finding the optimal monitoring corresponds to the robust maximization of functions $U(\cdot, s)$ for all s. Using the robust optimization framework of [20] together with our estimation guarantees of Sect. 3, we can improve the guarantees of Theorem 7 to work with additional budget $O(\ln |S|)$. This result asymptotically matches our guarantees of Theorem 8 (additional $\ln(|\mathcal{S}|)$ budget) for the case of deterministic diffusion.

5 Multiple-seed nodes

The model of Sect. 2 and our theoretical results are formulated in terms of a single seed node. It is natural, though, to ask about the case where, like the defender, the attacker has a budget b and selects a subset $S \subset \mathcal{S}$ of seed nodes such that $|S| \leq b$. In this section, we discuss which of our results extend to this more general setting.

5.1 Distributional setting

In our results for the distributional case (Sect. 3), the restriction to $b = 1$ is made purely for ease of exposition. That is, our hardness results (Theorem 5) which works for the case of $b = 1$, becomes only stronger when larger b is considered. As for our positive results, Theorem 1 can be extended to work with a general attacker budget b with the same approximation guarantee. To see why the proof of this theorem can be generalized, we show how each of the key ingredients of the proof can be generalized. As for Lemma 2, the argument for submodularity of the utility extends to a general budget b, by taking σ to be the partial ordering induced on the set of nodes which indicates the order in which the nodes became infected when the infection started from all b seeds nodes. Furthermore, we let $f_\sigma(M) = 1$ if and only if there is a monitor $m \in M$ that is infected in σ before the infection started at *all* selected seed nodes reaches the target.

Another key ingredient of the proof of Theorem 1 requires that $\hat{\mathcal{U}}^\tau$, which is obtained by taking the average utility of the defender under ℓ runs of the diffusion when the diffusion process is only considered up to τ time steps, is a good estimate of $\mathcal{U}$. This argument relies on two steps, first that the diffusion proceeds fast enough that after τ time steps it has infected a target or a monitor, with high probability (Lemma 3), and a concentration bound that shows that the average of defender's utility under a diffusion upto step τ is highly concentrated around its mean (Lemma 4). Note that for the first case, having a larger budget $b \geq 1$ only increases the speed of the diffusion and still ensures that a monitor or a target is infected, with high probability, before step τ. As for the second argument, the concentration bound does not depend on the nature of the diffusion, just that there are ℓ independent simulations of such a diffusion. Therefore, our Theorem 1 extends to the case of general attacker budget function immediately.

5.2 Maximin setting

In our results for the Maximin setting (Sect. 4), the $b = 1$ restriction does play a technical role and not all of our results can be extended for a general b. Here, we outline to what degree our results extend to this more general setting.

For our positive result, Theorem 7, our Algorithm 2 processes each possible seed node separately and achieves a $(1 - \epsilon)$-approximation of $\mathcal{V}(\text{OPT}_k)$ using $|\mathcal{S}|k \ln(\frac{1}{\epsilon})$ monitors. This approach provides guarantees when any single seed node can be selected. But when

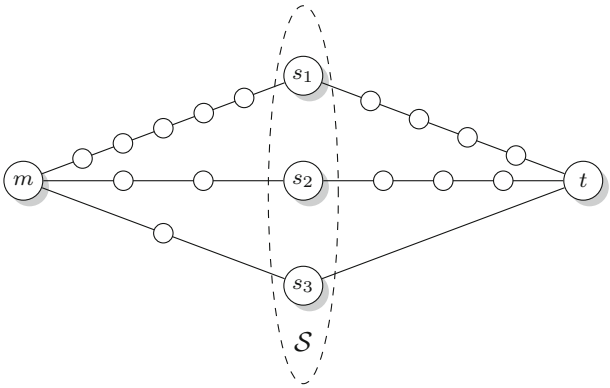


Fig. 5 A construction demonstrating the lack of submodularity and monotonicity of the attacker's utility. The diffusion is deterministic, i.e., $\rho = 1$

multiple-seed nodes are selected, this approach does not account for the diffusion process as a whole. We do not know whether there is a polynomial algorithm with similar approximation guarantee for any b. However, when b is a constant, a simple reduction to Theorem 7 solves this problem, albeit while requiring even larger budget for the defender. This reduction follows by creating a new seed node s_A, for any $A \subseteq \mathcal{S}$ such that $|A| \leq b$. We connect s_A to every seed node in A using edges with transmission probability 1. Now we consider the defender's problem when a *single* seed node is chosen from this new set of seed nodes $\{s_A \mid A \subseteq \mathcal{S} \text{ and } |A| \leq b\}$. Using Theorem 7, we can now obtain a $(1-\epsilon)$-approximation of $\mathcal{V}(\mathrm{OPT}_k)$ using $|\mathcal{S}|^b k \ln(\frac{1}{\epsilon})$ monitors.

As for the deterministic case of $\rho = 1$, Theorem 8 essentially goes through unchanged. Indeed, because the diffusion process is deterministic, for a choice of k monitors M, there are b seeds such that the process starting at all of them reaches the target before (or at the same time as) any monitor if and only if there is a single seed node with this property. So, in the deterministic case, it is sufficient to consider diffusion started at any single node, even if the attacker's budget $b > 1$.

5.3 Attacker's optimization problem

Another interesting aspect of the setting where the attacker can choose a set of b seeds nodes is from the attacker's point of view. From attacker's perspective, the objective of the attacker is not monotone in the size of the set S, unlike the traditional influence maximization problem: While seeding more nodes would increase the likelihood of reaching the target (or decrease the time to reach it), it may also increase the likelihood of being detected.

To formalize our results in this sections, consider the attacker's utility from seeds nodes $S \subseteq \mathcal{S}$ and a collection of monitor M. The attacker's utility is defined as the probability that the target node t is infected before any monitoring node detects an infection. We denote this utility by $\mathcal{U}_A(S, M)$. When considering a fixed set of monitors M, we simply refer to this quantity as $\mathcal{U}_A(S)$. Next, we show with an example that the attacker's utility is not submodular or monotone for a fix set of monitors M.

Example 9 Consider a network with 3 seed nodes $s_1, \ldots, s_3$, a fixed monitor node m, a target t, and additional nodes. Let s_1 be connected to t by a path of length 5, and by a path of length

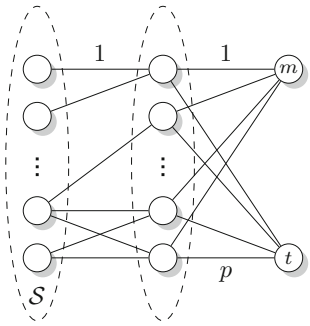


Fig. 6 Illustration of the construction used in the proof of Theorem 10. Edges are labeled by their probability

6 to m. Let s_2 be connected to t by a path of length 4, and by a path of length 3 to m, and s_3 to be connected to t directly, and by a path of length 2 to m. See Fig. 5.

Now consider two sets $X = \{s_1\}$ and $Y = \{s_1, s_2\}$, and consider s_3. $\mathcal{U}_A(X) = 1$, since the s_1-t path is shorter than the s_1-m path. On the other hand, $\mathcal{U}_A(Y) = 0$, since s_2 is closer to m than s_1 and s_2 are to t. Therefore, $\mathcal{U}_A(\cdot)$ is not monotone.

As for lack of submodularity, notice that $\mathcal{U}_A(X \cup \{s_3\}) = 1$ and $\mathcal{U}_A(Y \cup \{s_3\}) = 1$. This is because s_3 is directly connected to t, so it will infect t before the monitor goes off. But, this shows that

$$1 = \mathcal{U}_A(Y \cup \{s_3\}) - \mathcal{U}_A(Y) > \mathcal{U}_A(X \cup \{s_3\}) - \mathcal{U}_A(X) = 0,$$

despite $X \subseteq Y$, so $\mathcal{U}_A(\cdot)$ is not submodular.

Next, we show that the attacker's utility cannot be approximated to $(1 - \frac{1}{e} + o(1))$ for the general budget b. This hardness result is obtained by a reduction from the MAX- COVER problem. At a high level, we embed a MAX- COVER instance in a diffusion problem, such that every element in the instance corresponds to a node that infects the target and monitor with probability p and 1, respectively (See Fig. 6). Every subset of the elements is represented by a candidate seed node that infects all of its members with probability 1. The higher the number of elements that are covered by seed nodes, the higher the probability that the target is infected no later than the monitor. Probability p is chosen such that the fraction of covered elements is translated directly into the probability that target is infected before the monitor. Therefore, the best possible approximation for the MAX- COVER problem translates into an approximation of the attacker's utility. The details of this approach are described below.

Theorem 10 *Finding a $(1 - \frac{1}{e} + o(1))$-approximately optimal seeding set is NP-hard under the IC and RIC models for the attacker. That is, it is NP-hard to find a set $S \subseteq \mathcal{S}$ such that $|S| \leq b$ and*

$$\frac{\mathcal{U}_A(S)}{\mathcal{U}_A(\mathrm{OPT}_b)} > 1 - \frac{1}{e}.$$

Proof We use a reduction from the promise version of MAX- COVER problem: Given a set of elements U, a collection of its subsets $A \subseteq 2^U$, and a budget b such that there exists a subset of A with size b that covers all the elements U, it is NP-hard to find a subset of A of size b that covers more than $1 - \frac{1}{e}$ fraction of U [15].

We create a graph $G = (V, E)$ as follows. V includes one vertex per $a \in A$, one vertex per $u \in U$, a monitor node m, and the target t (see Fig. 6). The set of edges and their corresponding probabilities are as follow.

$$E = \left\{ \begin{array}{lll} e : au & \forall a \in A, u \in U, \text{ s.t. } u \in a & p_e = 1 \\ e : ut & \forall u \in U & p_e = \frac{1}{|U|^2} \\ e : um & \forall u \in U & p_e = 1 \end{array} \right\}$$

This graph is an instance of the targeted diffusion problem with potential seed set $\mathcal{S}$ corresponding to nodes in A, m being the fixed monitor node, and t being the target node. Let S' be the choice of seed nodes that correspond to a b-cover of (U, A) and OPT_b be the optimal set of b seed nodes. To receive a nonzero utility, the attacker has to choose at least 1 seed from $\mathcal{S}$. Since there is a path of length 2 between any $s \in S$ to t that consists of edges with probability 1, the monitor goes off at time step 2. So for the attacker to succeed, t has to be infected at step 1 or 2.

Let $S' \subseteq \mathcal{S}$ be any selection of k seed nodes and consider the set of coverage of S' (set of its neighbors) $\Gamma(S')$. So, the utility of S' to the attacker is the probability that at least one of the nodes in $\Gamma(S')$, which is definitely infected in the first time step, infects t at the next time step. Then, the utility of S' is the probability of the complement of the event where none of the members of $\Gamma(S')$ transmit the infection to t. That is,

$$\mathcal{U}_A(S') = 1 - \left(1 - \frac{1}{|U|^2}\right)^{|\Gamma(S')|}.$$

Similarly, for $S \subseteq \mathcal{S}$ that represent the optimal choice for max-coverage, we have that $\Gamma(S) = U$, and

$$\mathcal{U}_A(\text{OPT}_b) = 1 - \left(1 - \frac{1}{|U|^2}\right)^{|U|}.$$

Choose α such that $|\Gamma(S')| = \alpha|U|$. Let $|U| = m$ for each of notation. We have

$$\begin{aligned} \lim_{m\to\infty} \frac{\mathcal{U}_A(S')}{\mathcal{U}_A(S)} &= \frac{1 - \left(1 - \frac{1}{m^2}\right)^{\alpha m}}{1 - \left(1 - \frac{1}{m^2}\right)^{m}} \\ &= \lim_{m\to\infty} \frac{-\left(1 - \frac{1}{m^2}\right)^{\alpha m} \left(\frac{2\alpha}{\left(1-\frac{1}{m^2}\right)m^2} + \alpha \log\left(1 - \frac{1}{m^2}\right)\right)}{-\left(1 - \frac{1}{m^2}\right)^{m} \left(\frac{2}{\left(1-\frac{1}{m^2}\right)m^2} + \log\left(1 - \frac{1}{m^2}\right)\right)} \\ &= \lim_{m\to\infty} \frac{\alpha \log\left(1 - \frac{1}{m^2}\right)}{\log\left(1 - \frac{1}{m^2}\right)} = \alpha, \end{aligned}$$

where the calculation is similar to the calculation in the proof of Theorem 5. So, if $\frac{\mathcal{U}_A(S)}{\mathcal{U}_A(S')} > 1 - \frac{1}{e}$, then $|\Gamma(S')| > (1 - \frac{1}{e})m$. This implies that a polynomial-time algorithm produces a $(1 - \frac{1}{e})$-approximation for any MAX- COVER instance, which contradicts the hardness of $(1 - \frac{1}{e})$-approximation for MAX- COVER. □

To solve the attacker's problem in practice, we propose a greedy heuristic, presented below as Algorithm 4. It is easy to see that this algorithms runs in polynomial time. In Sect. 7, we will also demonstrate using numerical results that it performs exceptionally well in practice.

Algorithm 4 GREEDY ATTACK

Input: $G, M, \mathcal{S}, b, t$.

1. Start with $S \leftarrow \emptyset$ and $DONE \leftarrow false$.
2. While $|S| \leq b$ and not $DONE$ do
   (a) Let $s \in \mathcal{S} \backslash S$ be a node that maximizes the marginal gain $\mathcal{U}_A(S \cup \{s\}) - \mathcal{U}_A(S)$.
   (b) If the marginal gain is positive, set $S \leftarrow S \cup \{s\}$; otherwise, set $DONE \leftarrow true$.

Output: Set of seed nodes S.

6 Other generalizations

In addition, our model and results can be generalized in another direction: detection delay. Specifically, we can allow monitoring to take arbitrarily long to detect an infection, by associating with each node $v \in V$ a discrete distribution over the number of iterations of the diffusion process between the point of time v is infected and the point in which it detects the infection.

Happily, essentially all our results go through when detection delays are allowed. In particular, submodularity of the utility function can be shown to hold by taking the detection delays, too, into account when considering each infection order σ. For example, if m was infected two rounds before t, but its detection delay is, say, five rounds, then it will appear after t in the order.

Above we say "*essentially* all our results" because Theorem 8 is stated for a deterministic diffusion process; it does generalize to the detection delay setting when delays are deterministic (in that case each vertex can simply be replaced by a path).

7 Numerical results

In this section, we present numerical results on the algorithms proposed in Sects. 3, 4, and 5. Furthermore, we also introduce two simple heuristics for the maximin setting, which perform very well in practice.[3]

We conducted our experiments on three types of networks:

- Erdős–Rényi (E–R) random graphs [10]: We generated random networks having 100 nodes and each possible edge being present with probability 0.5. This model is one of the most widely used random-graph models and, hence, constitutes a good baseline.
- Barabási–Albert (B–A) random graphs [2]: We generated random networks of 100 nodes, starting with cliques of 3 nodes and connecting every additional node to 3 existing ones. B–A graphs are widely used to construct synthetic graphs as their heavy-tailed degree distribution resembles real social and technological networks.
- autonomous system (AS) relationship graph: In the Internet, an AS is a collection of connected routing prefixes under the control of a single administrative entity. Even though the network formed by AS does not correspond directly to the propagation network, it arises from similar technological and business processes. The graph used in our experiments was obtained from the Cooperative Association for Internet Data Analysis (CAIDA),[4] and consists of 68,526 nodes and 177,000 edges.

To instantiate our problem, we selected uniformly at random:

[3] The software and dataset used for these experiments are available at http://aronlaszka.com/data/haghtalab2015monitoring.zip.

[4] http://as-rank.caida.org/.

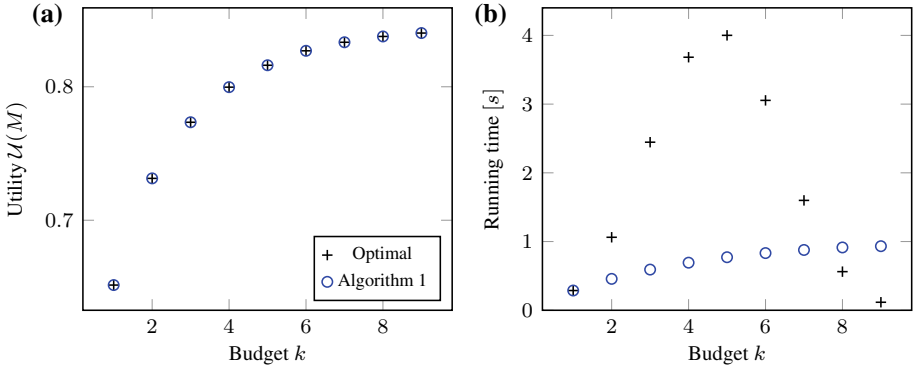


Fig. 7 Comparison of algorithms for the distributional setting on *B–A graphs* with *independent cascades* **a** utility, **b** running time

– 1 node to be the target node,
– 10 nodes to be the potential seed nodes,
– and 10 nodes to be the potential monitored nodes,

ensuring no overlap among these. Finally, we set the infection probability of each edge to 0.5.

For each setting, propagation model, network type, and budget value, we generated 15 instances (i.e., 15 random graphs and/or random node subsets as above) and plotted the average values over these instances. Finally, to estimate $\mathcal{U}(M)$ or $\mathcal{V}(M)$ for a given set of nodes M in an instance, we simulated the diffusion process 10,000 times, each time running until either the target or a monitored node was infected.

We omit results for the repeated independent cascade model for the maximin setting and the attacker's problem, as they are qualitatively the same as the results presented below.

7.1 Distributional setting

In this setting, we showed that Algorithm 1 has provable approximation guarantees. In our experiments, we consider empirically how close its solutions are to optimal (computed by exhaustive search).

Figures 7, 8, and 9 show that our algorithm performs exceptionally well in the independent cascades model for B–A graphs, E–R graphs, and the AS relationship graph, respectively. Furthermore, as expected, its running time is much lower than that of the exhaustive search in the computationally more challenging cases. From the measured running times, we can see that our algorithm scales well (appears sublinear in the budget). Another interesting observation is that in the large AS network, increasing the budget beyond 4 appears to make little difference in the objective value, suggesting that it is most important to place the first few monitors well.

Figures 10 and 11 compare our algorithm to the exhaustive search in the repeated independent cascades model for B–A graphs and E–R graphs, respectively. Similarly, to the independent cascades model, we see that our algorithm performs exceptionally well. Since the results for the two models are qualitatively the same, we will omit numerical results for the repeated independent cascades model in the remainder of this paper.

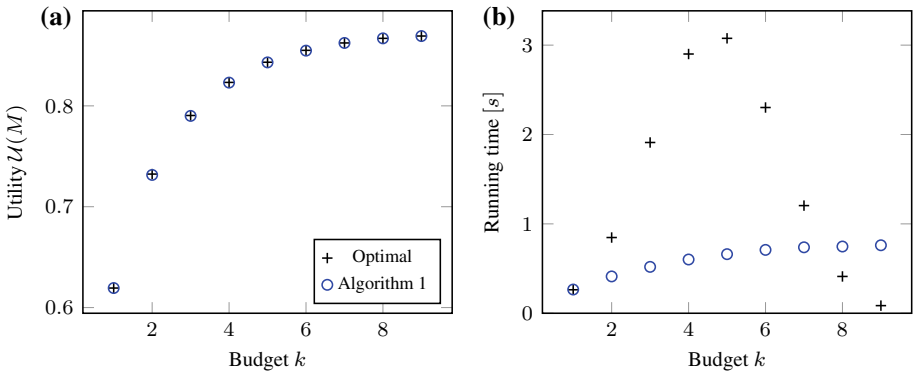


Fig. 8 Comparison of algorithms for the distributional setting on *E–R graphs* with *independent cascades* **a** utility, **b** running time

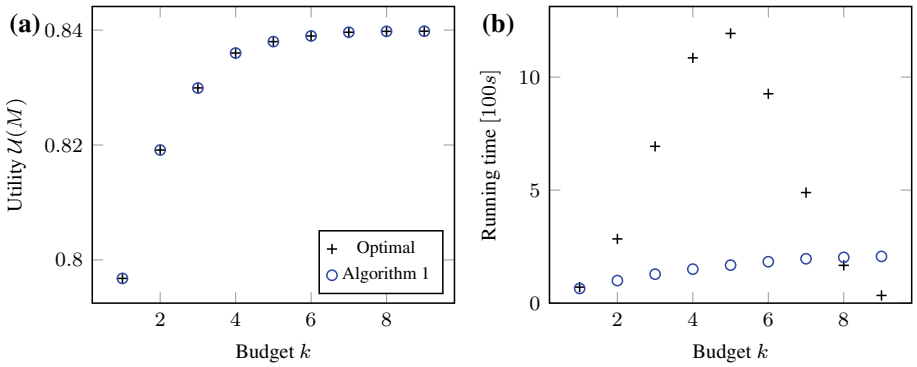


Fig. 9 Comparison of algorithms for the distributional setting on the *AS relationship graph* with *independent cascades* **a** utility, **b** running time

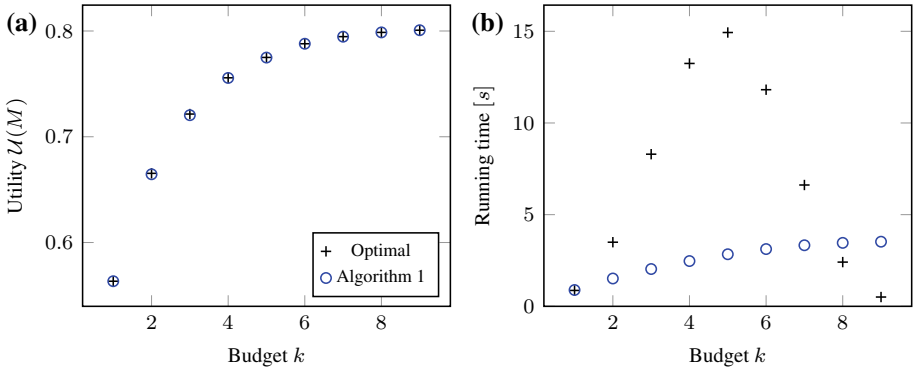


Fig. 10 Comparison of algorithms for the distributional setting on *B–A graphs* with *repeated independent cascades* **a** utility, **b** running time

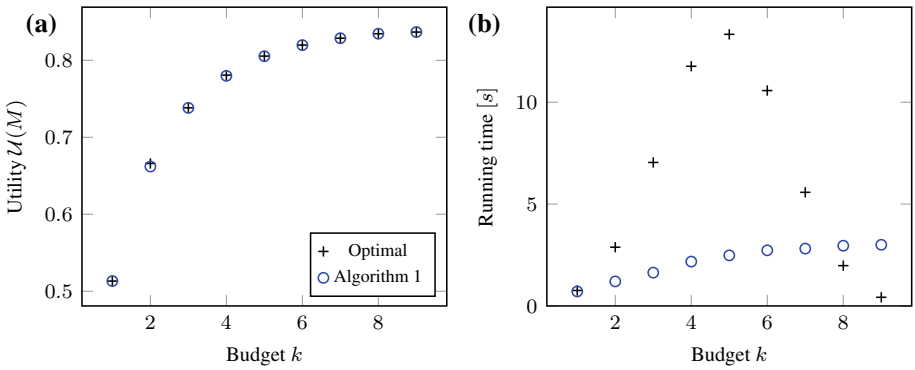


Fig. 11 Comparison of algorithms for the distributional setting on *E–R graphs* with *repeated independent cascades* **a** utility, **b** running time

7.2 Maximin setting

Next, we compare Algorithm 2 to an exhaustive search in the maximin setting. Recall from Sect. 4 that Algorithm 2 may output a set of monitored nodes whose size exceeds the budget. Consequently, to make a fair comparison, we use a variation of Algorithm 2, which is based on the same principle, but always produces a set of size k. More specifically, we increment the sets $M(s)$ at the same time (i.e., we iterate over all the seed nodes and increment each set, then iterate over all the seed nodes again) and stop the algorithm as soon as the size of their union $M = \cup_s M(s)$ reaches k.

As we will see, Algorithm 2 does not perform as well in the maximin setting as Algorithm 1 does in the distributional setting. Consequently, we introduce two new algorithms, called *greedy* and *heuristic*, which are closer to optimal in practice.

- **Greedy** is a straightforward greedy algorithm for maximizing the set function $\mathcal{V}(M)$ (i.e., the same as Algorithm 1, but maximizes $\mathcal{V}$ instead of $\mathcal{U}$).
- **Heuristic** is a greedy heuristic algorithm which works as follows: start with an empty set $M = \emptyset$ and add nodes to M iteratively; in each iteration, take a seed node s with minimum $\mathcal{U}_s$, and add a monitoring node m that maximizes $\mathcal{U}_s(M \cup \{m\})$ to M. The rationale behind this heuristic is that in order to secure the target against the worst-case attacker of the maximin setting, we have to "cover" the seed node that is least "covered."

Figures 12, 13, and 14 compare Algorithm 2, greedy, heuristic, and exhaustive search in the independent cascades model for B–A graphs, E–R graphs, and the AS relationship graph, respectively (in the AS graph, we omit optimal exhaustive search, which is intractable). Firstly, we can see that Algorithm 2 does not perform well, even compared to the greedy and heuristic algorithms. On the other hand, the greedy algorithm is near optimal, but its running time is the highest among the suboptimal algorithms. Finally, the heuristic algorithm performs reasonably well, especially in more complex cases, and its running time is the lowest among all. That said, an advantage of Algorithm 2 is that it provides worst-case guarantees, whereas there are examples showing that the greedy and heuristic algorithms fail miserably in the worst case.

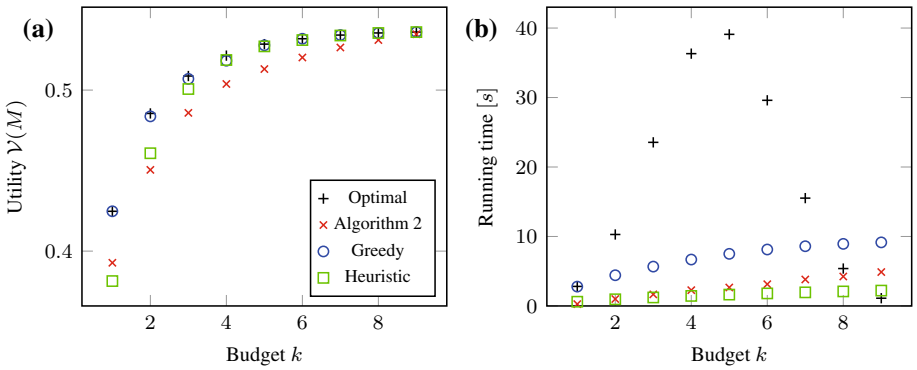


Fig. 12 Comparison of algorithms for the maximin setting on *B–A graphs* with *independent cascades* **a** utility, **b** running time

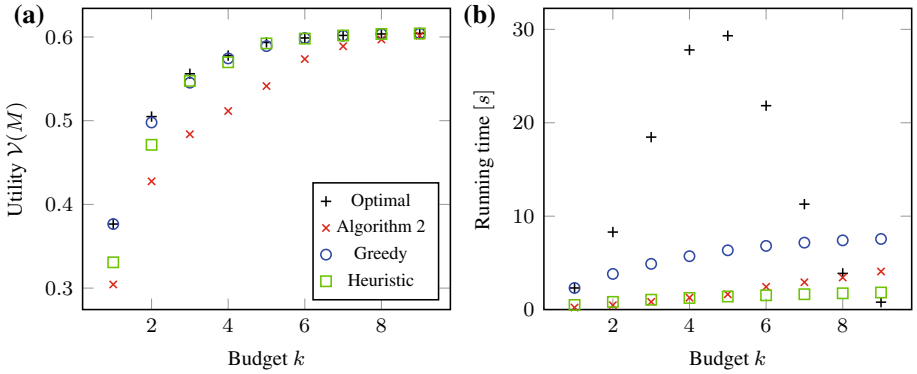


Fig. 13 Comparison of algorithms for the maximin setting on *E–R graphs* with *independent cascades* **a** utility, **b** running time

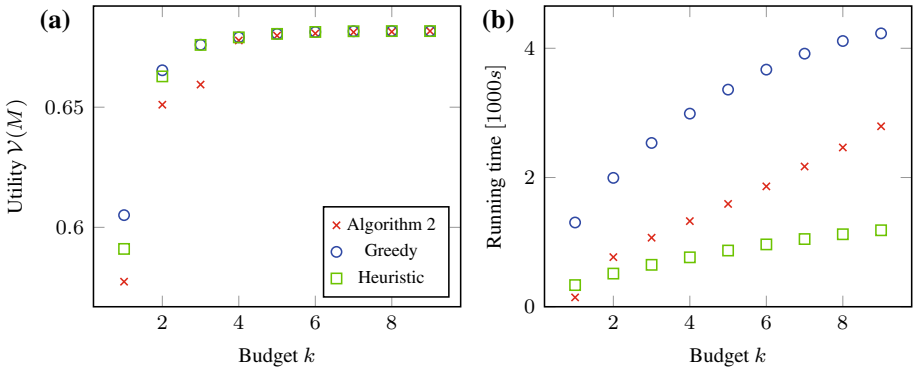


Fig. 14 Comparison of algorithms for the maximin setting on the *AS relationship graph* with *independent cascades* **a** utility, **b** running time

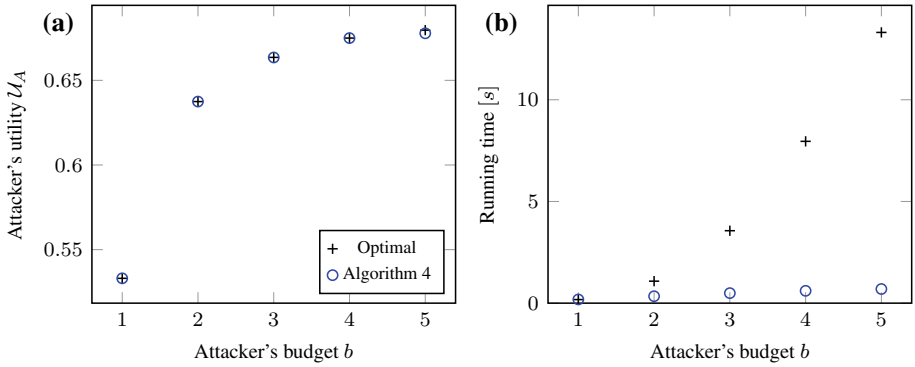


Fig. 15 Comparison of attack algorithms on *B–A graphs* with *independent cascades* **a** utility, **b** running time

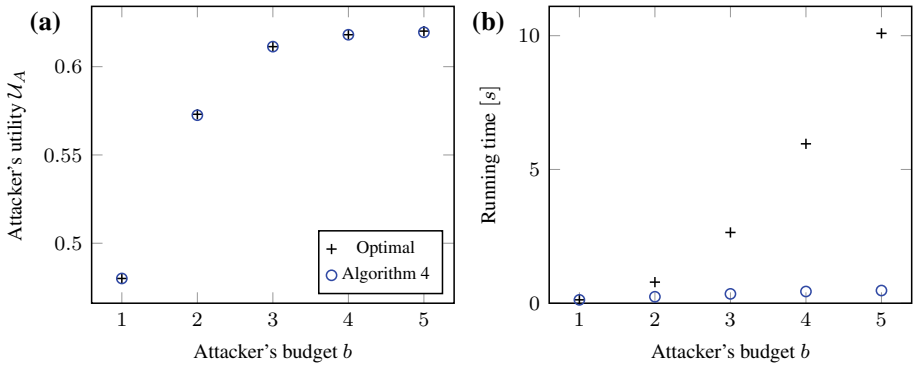


Fig. 16 Comparison of attack algorithms on *E–R graphs* with *independent cascades* **a** utility, **b** running time

7.3 Attack algorithms for multiple-seed nodes

Finally, we evaluate Algorithm 4, which we introduced in Sect. 5 for finding multiple-seed attacks in polynomial time. Similarly to the evaluation of the defense algorithms, we consider how close the solutions of Algorithm 4 are to optimal (computed by exhaustive search) in terms of utility. However, in this case, the comparison is based on the attacker's utility (i.e., probability of winning) instead of the defender's. Lastly, since the problem requires finding an attack against a given set of monitoring nodes, we selected 5 monitoring nodes at random to be M for each instance.

Figures 15 and 16 show that our algorithm performs exceptionally well for both B–A and E–R graphs. For most instances, the output of Algorithm 4 is in fact optimal, and the average difference to the optimum remains below 0.3%. Moreover, as expected, its running time is orders of magnitude lower than that of the exhaustive search in cases that are computationally more challenging.

Figure 17 shows that our algorithm also performs well for the AS relationship graph. The difference to the optimum is higher than for B–A and E–R graphs, but on average, it remains below 3.1%. Finally, the running time of our algorithm is again orders of magnitude lower than that of the exhaustive search in computationally more challenging cases.

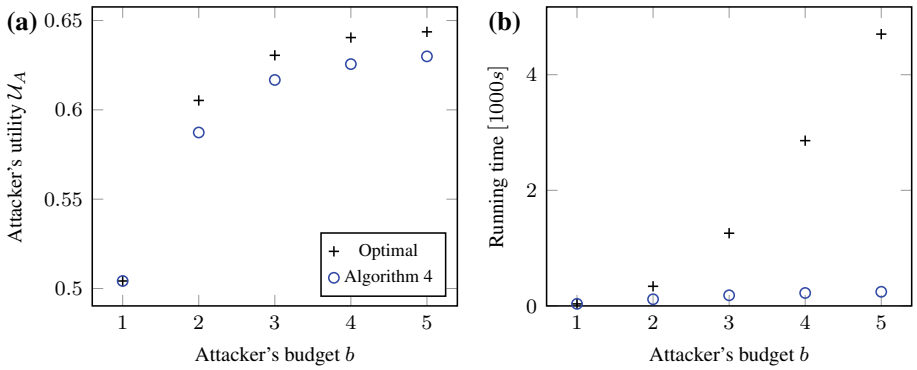


Fig. 17 Comparison of attack algorithms on the *AS relationship graph* with *independent cascades* **a** utility, **b** running time

8 Conclusion

We introduced a novel model of stealthy diffusion, relevant in many cyber (and cyber-physical system) security settings, whereby an adversary aims to attack a specific target but simultaneously to avoid detection. Focusing on the defender's problem of choosing monitor locations so as to maximize the probability of detecting such stealthy diffusion (e.g., of malware) prior to its reaching the target, we present both negative (inapproximability) results, and polynomial-time algorithms for several natural variants of this problem. In one of these variants, where the attacker randomly chooses an initial site of infection, we exhibited a greedy algorithm which achieves a constant factor approximation. In another, where the attacker optimally responds to monitor placement in the choice of initial infection, we exhibited several polynomial-time algorithms which can return solutions arbitrarily close to optimal, but at the cost of using more monitoring nodes. In our experiments, we introduced two additional heuristics for the latter variant of the problem, and while all algorithms proved effective at solving the problem, the two heuristics were particularly good, even though they can be arbitrarily suboptimal on some classes of networks.

We also considered generalizations of the above settings, in which the attacker chooses more than a single initial site of infection. While all of our results about the distributional setting readily extended to this case, generalizing results for the maximin setting proved to be non-trivial. Moreover, with more than a single node to choose, the attacker's problem itself becomes quite challenging, which we have confirmed by proving that the problem is in fact NP-hard. Finally, we provided a polynomial-time heuristic algorithm for solving the attacker's problem and demonstrated using numerical results that it performs exceptionally well in practice.

Acknowledgements We thank the anonymous reviewers for their helpful comments on the conference version of this paper. This work was supported in part by the National Science Foundation (CNS-1238959, CCF-1215883, IIS-1350598, IIS-1526860, and CCF-1525932), National Institute of Standards and Technology (70NANB13H169), Air Force Research Laboratory (FA8750-14-2-0180), Office of Naval Research (N00014-15-1-2621), Army Research Office (W911NF-16-1-0069), a Sloan Research Fellowship, an IBM Ph.D. Fellowship, and a Microsoft Research Ph.D. Fellowship.

References

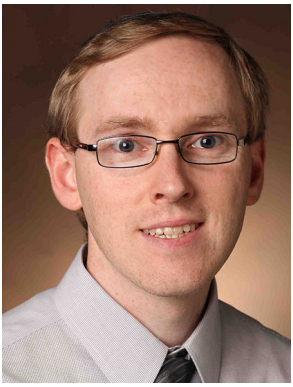
1. Adler M, Räcke H, Sivadasan N, Sohler C, Vöcking B (2003) Randomized pursuit-evasion in graphs. Comb Probab Comput 12(03):225–244
2. Barabási A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512
3. Bass FM (1969) A new product growth for model consumer durables. Manag Sci 15(5):215–227
4. Bharathi S, Kempe D, Salek M (2007) Competitive influence maximization in social networks. In: Proceedings of the 3rd conference on web and internet economics (WINE), pp 306–311
5. Borodin A, Filmus Y, Oren J (2010) Threshold models for competitive influence in social networks. In: Proceedings of the 6th conference on web and internet economics (WINE), pp 539–550
6. Chen W, Wang C, Wang Y (2010) Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th international conference on knowledge discovery and data mining (KDD). ACM, pp 1029–1038
7. Clark A, Poovendran R (2011) Maximizing influence in competitive environments: A game-theoretic approach. In: Proceedings of the 2nd conference on decision and game theory for security (GameSec), pp 151–162
8. Dinur I, Steurer D (2014) Analytical approach to parallel repetition. In: Proceedings of the 46th annual ACM symposium on theory of computing (STOC). ACM, pp 624–633
9. Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the 7th international conference on knowledge discovery and data mining (KDD). ACM, pp 57–66
10. Erdős P, Rényi A (1959) On random graphs I. Publ Math 6:290–297
11. Ganesh A, Massoulié L, Towsley D (2005) The effect of network topology on the spread of epidemics. In: Proceedings of the 24th annual IEEE joint conference of the IEEE computer and communications societies, vol 2. IEEE, pp 1455–1466
12. Haghtalab N, Laszka A, Procaccia AD, Vorobeychik Y, Koutsoukos X (2015) Monitoring stealthy diffusion. In: Proceedings of the 15th IEEE international conference on data mining (ICDM), pp 151–160
13. He X, Song G, Chen W, Jiang Q (2012) Influence blocking maximization in social networks under the competitive linear threshold model. In: Proceedings of the 12th IEEE international conference on data mining (ICDM), pp 463–474
14. Isler V, Kannan S, Khanna S (2006) Randomized pursuit-evasion with local visibility. SIAM J Discrete Math 20(1):26–41
15. Johnson DS (1973) Approximation algorithms for combinatorial problems. In: Proceedings of the 5th annual ACM symposium on theory of computing (STOC). ACM, pp 38–49
16. Kaspersky Labs' Global Research & Analysis Team (2012) Gauss: abnormal distribution. https://securelist.com/analysis/36620/gauss-abnormal-distribution/. Accessed 30 May 2015
17. Kelley MB (2013) The Stuxnet attack on Iran's nuclear plant was 'far more dangerous' than previously thought', Business Insider. http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11. Accessed 30 May 2015
18. Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: Proceedings of the 9th international conference on knowledge discovery and data mining (KDD). ACM, pp 137–146
19. Kempe D, Kleinberg J, Tardos E (2005) Influential nodes in a diffusion model for social networks, In: Proceedings of the international colloquium on automata, languages and programming (ICALP). Springer, pp 1127–1138
20. Krause A, McMahan B, Guestrin C, Gupta A (2007) Selecting observations against adversarial objectives. In: Proceedings of the 21st annual conference on neural information processing systems (NIPS), pp 777–784
21. Lelarge M (2009) Economics of malware: epidemic risks model, network externalities and incentives. In: Proceedings of the 47th annual Allerton conference on communication, control, and computing (Allerton), pp 1353–1360
22. Mossel E, Roch S (2007) On the submodularity of influence in social networks. In: Proceedings of the 39th annual ACM symposium on theory of computing (STOC). ACM, pp 128–134
23. Mukhopadhyay A, Zhang C, Vorobeychik Y, Tambe M, Pence K, Speer P (2016) Optimal allocation of police patrol resources using a continuous-time crime model, In: Proceedings of the 7th conference on decision and game theory for security (GameSec). Springer, pp 139–158
24. Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions. Math Program 14(1):265–294
25. Omic J, Orda A, Van Mieghem P (2009) Protecting against network infections: A game theoretic perspective. In: Proceedings of the 28th IEEE conference on computer communications (INFOCOM), pp 1485–1493

26. Parsons TD (1978) Pursuit-evasion in a graph. In: Theory and applications of graphs. Springer, pp 426–441
27. Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th international conference on knowledge discovery and data mining (KDD). ACM, pp 61–70
28. Tsai J, Nguyen TH, Tambe M (2012) Security games for controlling contagion. In: Proceedings of the 26th AAAI conference on artificial intelligence (AAAI), pp 1464–1470
29. Tsai J, Qian Y, Vorobeychik Y, Kiekintveld C, Tambe M (2013) Bayesian security games for controlling contagion. In: Proceedings of the 2013 ASE/IEEE international conference on social computing (SocialCom), pp 33–38
30. Van Mieghem P, Omic J, Kooij R (2009) Virus spread in networks. IEEE/ACM Trans Netw 17(1):1–14
31. Vorobeychik Y, Letchford J (2015) Securing interdependent assets. J Auton Agents Multiagent Syst 29(2):305–333
32. Yang J, Leskovec J (2010) Modeling information diffusion in implicit networks. In: Proceedings of the 10th IEEE international conference on data mining (ICDM). IEEE, pp 599–608
33. Zou CC, Gong W, Towsley D (2002) Code Red worm propagation modeling and analysis. In: Proceedings of the 9th ACM conference on computer and communications security (CCS), pp 138–147

Nika Haghtalab is a Ph.D. student at the Computer Science department of Carnegie Mellon University, co-advised by Avrim Blum and Ariel Procaccia. Nika received B.Math and M.Math degrees in Computer Science from the University of Waterloo, Canada. Her research interests include machine learning theory, computational aspects of economics, and algorithms. Nika is a recipient of the IBM and Microsoft Research Ph.D. fellowships.

Aron Laszka is a Research Assistant Professor in the Department of Electrical Engineering and Computer Science at Vanderbilt University. Previously, he was a Postdoctoral Scholar at the University of California, Berkeley. Between 2014 and 2015, he was a Postdoctoral Research Scholar at Vanderbilt University. He received Ph.D. (2014) and M.Sc. (2011) degrees in Computer Science and Engineering from Budapest University of Technology and Economics. In 2013, he was a Visiting Research Scholar at the Pennsylvania State University. His research work focuses on the security and resilience of cyber-physical systems, the economics of security, and game-theoretic modeling of security problems.

Ariel D. Procaccia is an Assistant Professor in the Computer Science Department at Carnegie Mellon University, and an Affiliated Faculty in the Machine Learning Department. He usually works on problems at the interface of computer science and economics. His distinctions include the IJCAI Computers and Thought Award (2015), the Sloan Research Fellowship (2015), the NSF Faculty Early Career Development Award (2014), and the IFAAMAS Victor Lesser Distinguished Dissertation Award (2009); as well as multiple paper awards including Best Paper (2016) and Best Student Paper (2014) at the ACM Conference on Economics and Computation (EC). Procaccia's work has helped many thousands of people make smarter group decisions, via the not-for-profit websites Spliddit.org and RoboVote.org. He is co-editor of the Handbook of Computational Social Choice (Cambridge University Press, 2016).



Yevgeniy Vorobeychik is an Assistant Professor of Computer Science and Biomedical Informatics at Vanderbilt University. Previously, he was a Principal Member of Technical Staff at Sandia National Laboratories. Between 2008 and 2010 he was a postdoctoral research associate at the University of Pennsylvania Computer and Information Science department. His work focuses on game-theoretic modeling of security and privacy, algorithmic and behavioral game theory and incentive design, optimization, complex systems, epidemic control, network economics, and machine learning. His research has been supported by the National Science Foundation, the National Institutes of Health, the Department of Energy, and the Department of Defense. He was nominated for the 2008 ACM Doctoral Dissertation Award and received honorable mention for the 2008 IFAAMAS Distinguished Dissertation Award. He is a member of the IEEE.

Xenofon Koutsoukos received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 2000. He is a Professor with the Department of Electrical Engineering and Computer Science and a Senior Research Scientist with the Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, TN, USA. He was a Member of Research Staff at the Xerox Palo Alto Research Center (PARC) (2000–2002), working in the embedded collaborative computing area. His research work is in the area of cyber-physical systems with emphasis on formal methods, data-driven methods, distributed algorithms, security and resilience, diagnosis and fault tolerance, and adaptive resource management. Prof. Koutsoukos was the recipient of the NSF Career Award in 2004, the Excellence in Teaching Award in 2009 from the Vanderbilt University School of Engineering, and the 2011 NASA Aeronautics Research Mission Directorate (ARMD) Associate Administrator (AA) Award in Technology and Innovation.