2015 IEEE International Conference on Data Mining

Monitoring Stealthy Diffusion

Nika Haghtalab*, Aron Laszka†, Ariel D. Procaccia*, Yevgeniy Vorobeychik† and Xenofon Koutsoukos†
*Carnegie Mellon University
{nhaghtal,arielpro}@cs.cmu.edu
†Vanderbilt University
{aron.laszka,yevgeniy.vorobeychik,xenofon.koutsoukos}@vanderbilt.edu

***Abstract*—Starting with the seminal work by Kempe et al., a broad variety of problems, such as targeted marketing and the spread of viruses and malware, have been modeled as selecting a subset of nodes to maximize diffusion through a network. In cyber-security applications, however, a key consideration largely ignored in this literature is stealth. In particular, an attacker often has a specific target in mind, but succeeds only if the target is reached (e.g., by malware) 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, so as to limit the likelihood of such targeted and stealthy diffusion processes reaching their intended targets. We investigate the problem of optimal monitoring of targeted stealthy diffusion processes, and show that a number of natural variants of this problem are NP-hard to approximate. On the positive side, we show that if stealthy diffusion starts from randomly selected nodes, the defender's objective is submodular, and a fast greedy algorithm has provable approximation guarantees. In addition, we present approximation algorithms for the setting in which an attacker optimally responds to the placement of monitoring nodes by adaptively selecting the starting nodes for the diffusion process. Our experimental results show that the proposed algorithms are highly effective and scalable.**

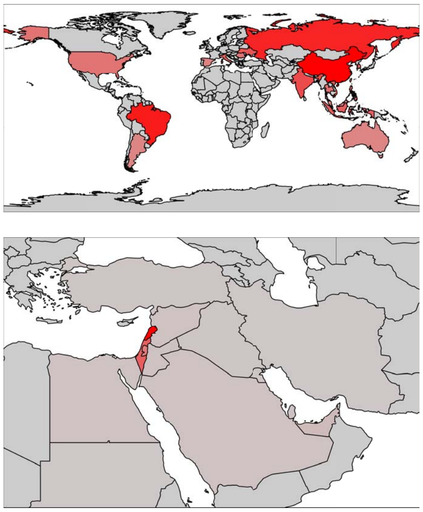


Fig. 1. Many worms, such as Conficker (top), spread so as to maximize the number of infections. Others, like Gauss (bottom), aim at specific targets, and deliberately try to avoid being detected, so that their spread is highly localized.

I. Introduction

In recent years, diffusion processes in social networks have been the focus of intense study [1], [2], [3], [4], [5]. 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 [6], [7]. 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 [8], 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 non-targeted 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 [9], 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 [10]. 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 (Figure 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.

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.118

A. 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). 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 [11], [12], [13], [14], [15], [16], [17]. 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. [3], 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.

B. 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 Section III, 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 [3].

In Section IV, 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 — $|\mathcal{S}| k \ln(1/\epsilon)$ — 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 Section VI, 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.

II. 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 Sections III and IV, we assume that S is a singleton; we discuss the generalization to any number of seed nodes in Section V.

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 [3]. A set of *seed* nodes $S \subseteq V$ is 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

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

the likelihood of reaching the target, but also increases the likelihood of being detected *before the target is reached*. To formalize this tradeoff, 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 is 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 Section V 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.

III. 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, 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 Chen et al. [18]. 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 $\ell$ samples.
     b) Set $M \leftarrow M \cup \{m\}$.
Output: Set of monitors $M$.
```

Like Kempe et al. [3], to establish the approximation guarantee of this algorithm, we rely on the celebrated result of Nemhauser et al. [19] on optimizing *monotonically non-decreasing submodular* functions. A function F defined over a set S is said to be *submodular* if $F : 2^S \rightarrow \mathbb{R}^+$ 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. [19] 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 $\mathrm{poly}(n, \frac{1}{\epsilon}, \frac{1}{\rho}, \log(\frac{1}{\delta}))$ 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 1. *$\mathcal{U}(\cdot)$ 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 non-negative 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 non-negative, therefore, $\mathcal{U}(\cdot)$ is also monotonically non-decreasing. ■

Lemma 2. *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 $|\mathcal{U}(M) - \mathcal{U}^\tau(M)| \leq \epsilon$. ■

Lemma 3. *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 3, for each of these estimates, with probability $1 - \frac{\delta}{k|\mathcal{M}|}$, $\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}^\tau(M)\right| \leq \epsilon/(4k)$. 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 2, 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 Nemhauser et al. [19] (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.

Theorem 2. *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.

[2] Proof of Theorem 4 formalizes this argument for a more general optimization problem discussed in the future section.

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 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 [20].

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 Figure 2). The set of edges and their corresponding probabilities are as follow.

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

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

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

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. ∎

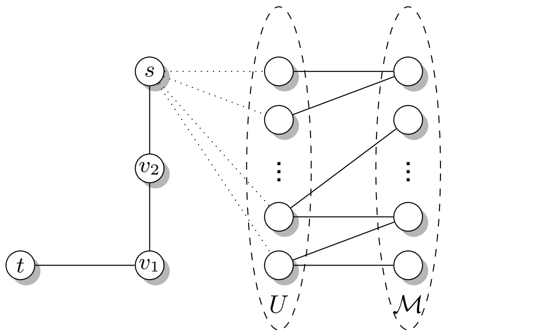


Fig. 2. Illustration of the construction used in the proof of Theorem 2. All solid edges have probability 1 and all dotted edges have probability $1/|U|^2$.

IV. 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 non-zero 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.

Theorem 3. *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*

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

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 [21].

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 Figure 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 (U, A). Then $\mathcal{V}(\mathrm{OPT}_k) = 1$, 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

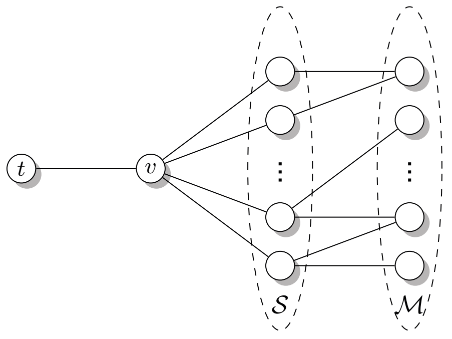


Fig. 3. Illustration of the construction used in the proof of Theorem 3. All edges have probability 1.

$\mathcal{V}(M) > 0$. 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 $u \in \mathcal{S} \setminus \Gamma(M)$ 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 $\mathcal{V}(\mathrm{OPT}_k)$ 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 $\ell$ tries up to $\tau$ time steps. Set $M(s) \leftarrow M(s)\cup\{m_i\}$.
     c) $M \leftarrow M \cup M(s)$.
Output: Set of monitors $M$.
```

Theorem 4. *For any maximin targeted diffusion instance, any* k*,* $\epsilon > 0$*,* $\gamma > 0$ *and* $\delta > 0$*, Algorithm 2 runs in time* $\mathrm{poly}(n, \frac{1}{\epsilon}, \frac{1}{\gamma}, \frac{1}{\rho}, \log(\frac{1}{\delta}))$ *and finds a set* $|M| \le |\mathcal{S}|k\ln(1/\epsilon)$ *such that with probability* $1-\delta$, $\mathcal{V}(M) \ge (1-\epsilon)\,\mathcal{V}(\mathrm{OPT}_k) - \gamma$*. 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 $\mathrm{OPT}_k(s)$ represent the optimal set of k monitors *when the attacker deterministically selects* s. Then for all $s \in \mathcal{S}$, $\mathcal{V}(OPT_k) \le \mathcal{U}_s(OPT_k(s))$.

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,

$$\mathcal{U}_s(M(s)) \ge (1-\epsilon)\,\mathcal{U}_s(OPT_k(s)) - \gamma, \tag{1}$$

and as a result,

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

Hereinafter, we focus on establishing Equation (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

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

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 $\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}^\tau(M)\right| \le \xi/4$. Additionally, by Lemma 2, $|\mathcal{U}^\tau(M) - \mathcal{U}(M)| \le \xi/4$. Therefore, with probability $1-\delta$, for any M considered by the algorithm, we have $\left|\hat{\mathcal{U}}^\tau(M) - \mathcal{U}(M)\right| \le \xi/2$.

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 $\hat{g}_M^\tau(m) = \hat{\mathcal{U}}^\tau(M\cup m) - \hat{\mathcal{U}}^\tau(M)$. Then, with probability $1-\delta$, for any M and m considered by the algorithm, we have $|\hat{g}_M^\tau(m) - g_M(m)| \le \xi$.

Next, for any $i \le k\ln(1/\epsilon)$, let $M_i = \bigcup_{j\le i} m_j$ 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.

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

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 $m_i = \arg\max_m \hat{g}_{M_{i-1}}^\tau(m)$. So in particular, m_i has higher estimated marginal utility than any monitor in the set $\mathrm{OPT}_k(s)\setminus M_{i-1}$. If $\mathrm{OPT}_k(s)\setminus M_{i-1} = \emptyset$, then we have already achieved utility of at least $\mathrm{OPT}_k(s)$ and the claim holds trivially. If not, then $0 < |\mathrm{OPT}_k(s)\setminus M_{i-1}| \le k$. So,

$$\hat{g}_{M_{i-1}}^\tau(m_i) \ge \frac{\sum_{m\in\mathrm{OPT}_k(s)\setminus M_{i-1}} \hat{g}_{M_{i-1}}^\tau(m)}{|\mathrm{OPT}_k(s)\setminus M_{i-1}|}.$$

Therefore,

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

On the other hand, using submodularlity, 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 Equation (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

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

Therefore, after $i = k \ln(1/\epsilon)$ rounds and replacing $\xi = \frac{\gamma}{2k \ln(1/\epsilon)}$, we get $\mathcal{U}_s(M(s)) \geq (1-\epsilon)\ \mathcal{U}_s(OPT_k(s)) - \gamma$. 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 3, which holds even for the $\rho = 1$ case, this is the smallest number of monitors needed to guarantee a non-zero 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 Figure 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 5. *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 $\mathcal{V}(\mathrm{OPT}_k) = 1$.

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 5), 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 Section VI.

V. GENERALIZATIONS

The model of Section II 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 $|S| \leq b$.

An interesting aspect of this more general problem from the attacker's perspective is that 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

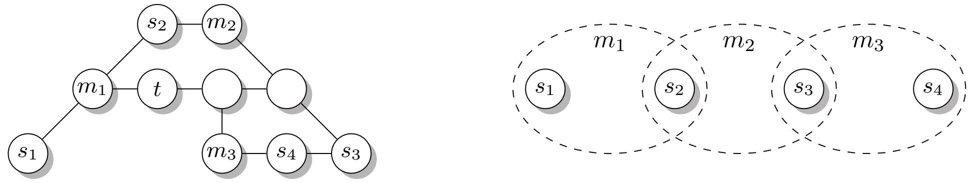


Fig. 4. An illustration of Step 2 of Algorithm 3. In the example, $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, and $\mathcal{M} = \{m_1, m_2, m_3\}$. The given graph is on the left, and the constructed set cover instance is on the right.

of being detected.

In Section III, the restriction to $b = 1$ is made purely for ease of exposition. Of course, it only makes the hardness result (Theorem 2) stronger. As for the positive result (Theorem 1), to see why the proof can be generalized with the same approximation guarantee, the key is to extend the submodularity argument (Lemma 1) by encoding the choice of all seed nodes in σ, and letting $f_\sigma(M) = 1$ if and only if there is a monitor $m \in M$ that is infected in σ before *all* selected seed nodes.

In Section IV, the $b = 1$ restriction does play a technical role — in the proof of Theorem 4. Specifically, Algorithm 2 processes each possible seed node separately. This approach provides guarantees when any single seed node can be selected. But when multiple seed nodes are selected, the approach does not account for the diffusion process as a whole.

In contrast, the positive result for $\rho = 1$ (Theorem 5) 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.

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 5 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).

VI. Numerical Results

In this section, we present numerical results on the approximation algorithms proposed in Sections III and IV. 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 [22]: 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 [23]: 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:

- 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.

Due to space constraints, we omit the results for repeated independent cascade model, as they are qualitatively the same as the results presented below.

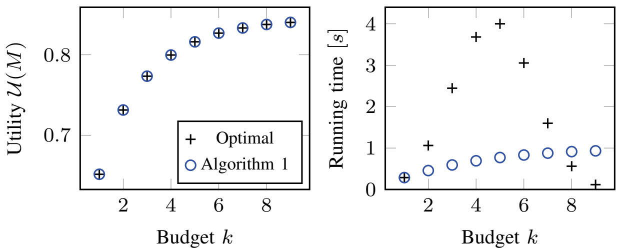
A. Distributional Setting

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 5 and 6 show that our algorithm performs exceptionally well for B-A and E-R graphs, respectively. Furthermore, as expected, its running time is much lower than that of the exhaustive search in the computationally more challenging cases. Moreover, Figure 7 shows the outputs

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

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

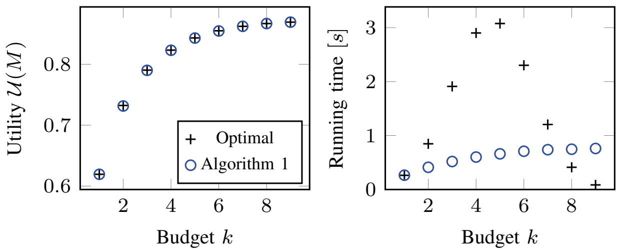
of Algorithm 1 for the AS relationship graph, and we can see from the measured running times that our algorithm scales well (appears sublinear in the budget). Another interesting observation is that in the large AS network increasing the budget beyond 4 appears to make little difference in the objective value, suggesting that it is most important to place the first few monitors well.



(a) Utility

(b) Running time

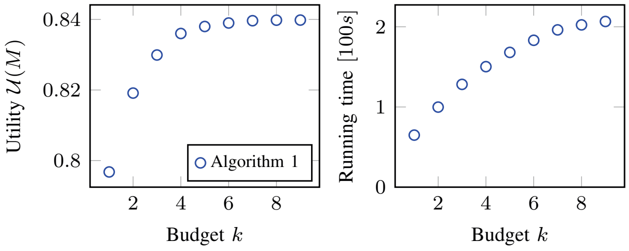
Fig. 5. Comparison of algorithms for the distributional setting on *B-A graphs* with *independent cascades*.



(a) Utility

(b) Running time

Fig. 6. Comparison of algorithms for the distributional setting on *E-R graphs* with *independent cascades*.



(a) Utility

(b) Running time

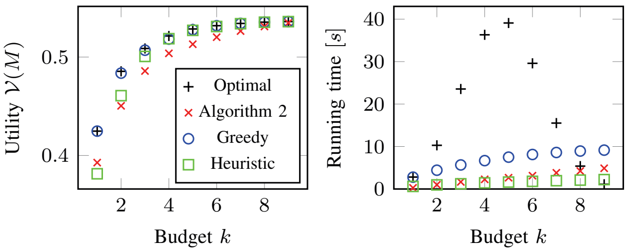
Fig. 7. Evaluation of Algorithm 1 for the distributional setting on the *AS relationship graph* with *independent cascades*.

B. Maximin Setting

Next, we compare Algorithm 2 to an exhaustive search in the maximin setting. Recall from Section IV 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, etc.) 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.

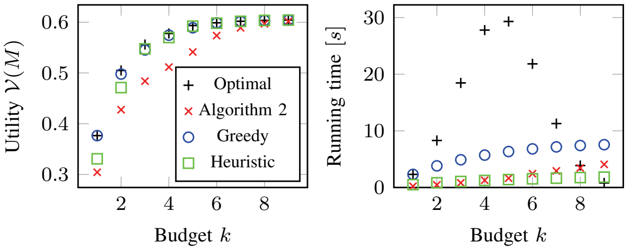
- **Greedy** is a straightforward greedy algorithm for maximizing the set function $\mathcal{V}(M)$ (i.e., the same as Algorithm 1, but maximizes $\mathcal{V}$ instead of $\mathcal{U}$).
- **Heuristic** is a greedy heuristic algorithm which works as follows: start with an empty set $M = \emptyset$ and 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."



(a) Utility

(b) Running time

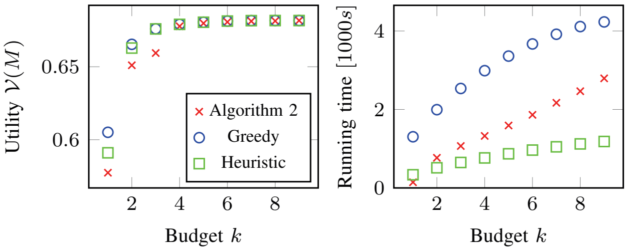
Fig. 8. Comparison of algorithms for the maximin setting on *B-A graphs* with *independent cascades*.



(a) Utility

(b) Running time

Fig. 9. Comparison of algorithms for the maximin setting on *E-R graphs* with *independent cascades*.



(a) Utility

(b) Running time

Fig. 10. Comparison of algorithms for the maximin setting on the *AS relationship graph* with *independent cascades*.

Figures 8, 9, and 10 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.

VII. 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.

There are a number of natural future research directions. First, while some of our results can be generalized to consider attackers choosing more than a single initial site of infection (see Section V), generalizing others (e.g., Algorithm 2) appears non-trivial. Moreover, with more than a single node to choose, the attacker's problem itself becomes quite challenging, and the development of both good algorithms and heuristics for this subproblem, as well as generalizing the defender's resulting task, are important open problems.

Acknowledgments

This work was supported in part by the National Science Foundation (CNS-1238959, CCF-1215883, IIS-1350598, and CCF-1525932), Air Force Research Laboratory (FA8750-14-2-0180), National Institute of Standards and Technology (70NANB13H169), Office of Naval Research (N00014-15-1-2621), and a Sloan Research Fellowship.

References

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. of the 7th International Conf. on Knowledge Discovery and Data Mining (KDD)*. ACM, 2001, pp. 57–66.

[2] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. of the 8th International Conf. on Knowledge Discovery and Data Mining (KDD)*. ACM, 2002, pp. 61–70.

[3] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. of the 9th International Conf. on Knowledge Discovery and Data Mining (KDD)*. ACM, 2003, pp. 137–146.

[4] ——, "Influential nodes in a diffusion model for social networks," in *Proc. of the International Colloquium on Automata, Languages and Programming (ICALP)*. Springer, 2005, pp. 1127–1138.

[5] E. Mossel and S. Roch, "On the submodularity of influence in social networks," in *Proc. of the 39th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2007, pp. 128–134.

[6] J. Omic, A. Orda, and P. Van Mieghem, "Protecting against network infections: A game theoretic perspective," in *Proc. of the 28th IEEE Conf. on Computer Communications (INFOCOM)*, 2009, pp. 1485–1493.

[7] M. Lelarge, "Economics of malware: Epidemic risks model, network externalities and incentives," in *Proc. of the 47th Annual Allerton Conf. on Communication, Control, and Computing (Allerton)*, 2009, pp. 1353–1360.

[8] C. C. Zou, W. Gong, and D. Towsley, "Code Red worm propagation modeling and analysis," in *Proc. of the 9th ACM Conf. on Computer and Communications Security (CCS)*, 2002, pp. 138–147.

[9] M. B. Kelley, "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, Nov 2013, accessed: May 30th, 2015.

[10] Kaspersky Labs' Global Research & Analysis Team, "Gauss: Abnormal distribution," https://securelist.com/analysis/36620/gauss-abnormal-distribution/, August 2012, accessed: May 30th, 2015.

[11] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *Proc. of the 3rd International Conf. on Internet and Network Economics (WINE)*, 2007, pp. 306–311.

[12] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in *Proc. of the 6th International Conf. on Internet and Network Economics (WINE)*, 2010, pp. 539–550.

[13] A. Clark and R. Poovendran, "Maximizing influence in competitive environments: A game-theoretic approach," in *Proc. of the 2nd Conf. on Decision and Game Theory for Security (GameSec)*, 2011, pp. 151–162.

[14] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in *Proc. of the 2012 SIAM International Conf. on Data Mining (SDM)*, 2012, pp. 463–474.

[15] J. Tsai, T. H. Nguyen, and M. Tambe, "Security games for controlling contagion," in *Proceedings of the 26th AAAI Conf. on Artificial Intelligence (AAAI)*, 2012, pp. 1464–1470.

[16] J. Tsai, Y. Qian, Y. Vorobeychik, C. Kiekintveld, and M. Tambe, "Bayesian security games for controlling contagion," in *Proc. of the 2013 ASE/IEEE International Conf. on Social Computing (SocialCom)*, 2013, pp. 33–38.

[17] Y. Vorobeychik and J. Letchford, "Securing interdependent assets," *Journal of Autonomous Agents and Multiagent Systems*, vol. 29, no. 2, pp. 305–333, 2015.

[18] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. of the 16th International Conf. on Knowledge Discovery and Data Mining (KDD)*. ACM, 2010, pp. 1029–1038.

[19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[20] D. S. Johnson, "Approximation algorithms for combinatorial problems," in *Proc. of the 5th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 1973, pp. 38–49.

[21] I. Dinur and D. Steurer, "Analytical approach to parallel repetition," in *Proc. of the 46th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2014, pp. 624–633.

[22] P. Erdős and A. Rényi, "On random graphs I." *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[23] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, October 1999.