Metadata-conscious anonymous messaging

Abstract
Anonymous messaging platforms allow users to spread messages over a network (e.g., a social network) without revealing message authorship to other users. Popular demand for anonymous messaging is evidenced by the success of mobile apps like Whisper and Yik Yak. In such platforms, the spread of messages is typically modeled as a diffusion process. Recent advances in network analysis have revealed that such diffusion processes are vulnerable to author deanonymization by adversaries with access to metadata, such as timing information. In this work, we ask the fundamental question of how to intervene in the propagation of anonymous messages in order to make it difficult to find the source. In particular, we study the performance of a message propagation protocol called adaptive diffusion introduced in (Fanti et al., 2015). We prove that it achieves asymptotically optimal source-hiding and significantly outperforms standard diffusion. We further demonstrate empirically that adaptive diffusion hides the source effectively on real social graphs.

1. Introduction
People have the right to express themselves without fear of repercussion. Popular means of expression today (Facebook, Twitter, and various messaging apps—Whatsapp, Kakao) seamlessly allow users to share potentially sensitive content with their friends. However, messaging platforms are not designed with user privacy in mind. Indeed, the contrary is often true (Johnson, 2010; CNBC, 2010), and the wealth of information in these social networks can lead to invasive monitoring by advertisers, employers, service providers, or government agencies. This monitoring typically exploits metadata: non-content data that characterizes content, like timestamps. Metadata can often be as sensitive as data itself (Narayanan & Shmatikov, 2009; Greschbach et al., 2012).

The privacy implications of social media are gaining attention; in response, a number of anonymous social media accounts have cropped up recently, including Whisper (whi), Yik Yak (yik), Blind (bli) and the now-defunct Secret (sec). These anonymous messaging apps are microblogging services that hide message authorship from other users. When a user posts a message, the message spreads (without authorship information) to the users’ contacts, or friends, in an underlying social network. If a message recipient approves a message by pressing the ‘like’ button, the message is further propagated to the recipient’s friends, and so on. The message thus spreads anonymously through the network—no single user can learn who authored a message. One drawback of existing anonymous messaging applications is that they are centralized, so company-owned servers store all messages and metadata. These servers are a central point of failure; an adversary wanting to deanonymize an individual can access the centralized servers via legal or technological means. And of course, the service provider itself has immediate access to authorship information.

A solution is to use a distributed architecture, in which there is no centralized repository of data or metadata (Cutillo et al., 2009). Users rely only on local information to transmit messages, and they pass only minimal metadata. Distributed architectures organically avoid many anonymity challenges, like the central point of failure. Unfortunately, recent advances in network analysis such as (Shah & Zaman, 2011; Pinto et al., 2012) suggest that a moderately powerful adversary can still infer which node has started the message, using limited metadata. Our goal in this work is to present a message propagation protocol and prove that it provides strong anonymity guarantees, even against an authoritarian adversary (described below).

Adversarial Models. We consider an adversary that has access to the underlying contact network $G(V,E)$. The adversary lacks the resources to monitor all network traffic, but it can collect partial metadata in a number of ways:

One way is to explicitly corrupt some fraction of nodes by bribery or coercion; these corrupted spy nodes continuously monitor metadata like message timestamps and relay IDs; we call this a spy-based adversary. This adversary represents government agencies using fake or corrupted social media accounts to monitor users (Sterbenz, 2013).

Alternatively, an adversary could use side channels to collect information on whether a node is infected, i.e., whether it received the message, at a fixed time; we call this a snapshot adversarial model. If an adversary uses spies...
and a snapshot, we call it a spy-snapshot adversary. The snapshot adversary has been well-studied in the literature; for both source identification (Shah & Zaman, 2011) and source obfuscation (Fanti et al., 2015). However, spy-based adversaries have not been studied from the source obfuscation perspective. In this paper, we focus on spy-based adversaries, and briefly discuss the implications of the spy-snapshot adversary in Sec. 4.

Under the spy-based adversarial model, we suppose each node other than the source is a spy independently with probability \( p \). At some point in time, the source node \( v^* \) starts propagating its message over the graph according to a spreading protocol chosen by the platform (to be determined). Each spy node \( s_i \in V \) observes: (1) the time \( T_{s_i} \) (relative to an absolute reference) at which it receives the message, (2) the parent node \( p_{s_i} \) that relayed the message, and (3) any other metadata used by the spreading mechanism (such as control signaling in the message header). At some time, spies aggregate their observations; using the collected metadata and the structure of the underlying graph, the adversary estimates the author of the message, \( \hat{v} \). A problem of central interest is to find a spreading mechanism that minimizes the probability of detection, \( \mathbb{P}(\hat{v} = v^*) \). This is the focus of this paper.

**Spreading mechanisms.** A common construction for modeling epidemic propagation over networks is diffusion: a symmetric random process in which each node spreads the message to its neighbors according to independent, random delays. Diffusion is a commonly-studied and useful model due to its simplicity and first-order approximation of actual propagation dynamics. Critically, it captures the symmetric spreading of most social media platforms.

Finding a computationally-efficient algorithm for (near-) optimal maximum likelihood (ML) message source inference is an open problem under the spy-based adversarial model, as is the corresponding detection probability analysis. Recent work (Pinto et al., 2012; Zhu et al., 2014) has focused on identifying the message source through heuristic, low-cost algorithms. These findings suggest that a spy-based adversary with metadata can locate the source with high probability under diffusion spreading. Indeed, when the underlying graph is a \( d \)-regular tree, we empirically observe that the probability of detection under diffusion increases with time and the degree of the underlying graph (Figure 5, Supplementary Materials). This has poor implications for anonymity; contact networks may have high degree nodes, and the adversary is not time-constrained.

We therefore seek a different spreading model with strong anonymity guarantees when the underlying graph has high degree, and estimation occurs at \( T = \infty \). In this paper, we analyze the anonymity properties of adaptive diffusion, the spreading model from (Fanti et al., 2015). Adaptive diffusion was originally designed to provide anonymity against a snapshot adversary. There is no reason to believe a priori that adaptive diffusion should perform well against a spy-based adversary with its access to timing information; surprisingly, it does.

**Contributions.** Our contributions are as follows:

1. We identify adaptive diffusion as an algorithm that provides strong anonymity guarantees against a spy-based adversary. Since (Fanti et al., 2015) contains multiple variants of adaptive diffusion, we identify the specific parameter setting under which it is both analytically tractable and provides strong anonymity guarantees.

2. Under the spy-based adversarial model and adaptive diffusion spreading, we identify a computationally-efficient algorithm for maximum likelihood source detection when the underlying contact network is infinite and tree-structured (Algorithm 1, Supplemental Material).

3. We give a precise analysis of the anonymity properties of adaptive diffusion. Such analysis is currently open for regular diffusion; we provide exact expressions for adaptive diffusion over regular trees (Theorem 1) and a lower bound for regular diffusion (Proposition 3.2), and show that our results are numerically stable for social network graphs, i.e., finite, irregular, and cyclic.

4. We show that over regular trees, adaptive diffusion has asymptotically optimal hiding guarantees (Proposition 3.1) as the degree of the underlying tree increases. This differs from regular diffusion, whose anonymity properties degrade as degree increases. Intuitively, spies near the source provide more information than distant ones; by spreading symmetrically, diffusion ensures that all nearby spies receive the message. Adaptive diffusion instead spreads asymmetrically, thereby preventing most nearby spies from seeing the message early enough to deanonymize.

**Related Work.** A snapshot-based adversary observes which nodes are infected at a certain time \( T \). When the infection spreads as per standard diffusion on a \( d \)-regular tree, efficient ML estimators exist for finding the source from the snapshot (Shah & Zaman, 2011). Further, the adversary can identify the source with probability converging to a constant lower-bounded by \( 1/3 \), as the time-to-attack grows. Subsequent work suggests that even under various diffusion models and estimators, source detection with a snapshot is reliable (Wang et al., 2014; Prakash et al., 2012; Fioriti & Chinnici, 2012; Luo et al., 2013; Zhu & Ying, 2013; Milling et al., 2012a; 2013; 2012b).

If we know when the adversary will attack, one solution for hiding the source on a \( d \)-regular tree is the following. For the first half, the infection propagates on a line in a randomly chosen direction; for the remaining half, the in-
fection spreads as per diffusion from the end of the line. At time $T$ all nodes in the boundary of the snapshot are equally likely to be the source, by symmetry. However, this line-and-diffusion protocol fails to protect the source if the adversary attacks before or after time $T$. As a remedy, adaptive diffusion was proposed to provide strong protection against a snapshot-based adversary (Fanti et al., 2015).

At any time $T$, adaptive diffusion ensures that all nodes are equally likely to have been the source. This provides perfect obfuscation; no adversary can find the source with probability larger than $1/N_T$ where $N_T$ is the number of infected nodes.

When the adversary collects timestamps (and other metadata) from spy nodes, standard diffusion reveals the location of the source (Pinto et al., 2012; Zhu & Ying, 2013). However, ML estimation is known to be NP-hard (Zhu et al., 2014), and analyzing the probability of detection is also challenging. Figure 5 (Supplemental Materials) shows that even with sub-optimal estimators, the source can be effectively identified. Since both snapshot and spy-based adversaries are plausible, we want to go beyond diffusion and line-and-diffusion. A natural question of interest is how to spread a message in order to provide strong protection against both types of adversaries: snapshot and spy-based.

Related challenges include (a) identifying the best algorithm that the adversary might use to infer the location of the source; (b) providing analytical guarantees for the proposed spreading model; and (c) identifying the fundamental limit on what any spreading model can achieve. We address all of these challenges.

Our work is part of a larger ecosystem that enables practical and truly anonymous messaging platforms. For instance, we assume that nodes communicate in a distributed fashion, but anonymity-preserving, peer-to-peer (P2P) presence lookup is in an active research area (Borisov et al., 2014), as is privacy-preserving distributed data storage in P2P systems (Jawad et al., 2009). Plausible attacks that are not addressed in this paper may operate below the application layer (e.g., by monitoring the network or even physical layer) (Winter & Lindskog, 2012; Singel, 2007).

Lower-level protections may be more appropriate against such an opponent, harnessing factors like physical proximity of users (Shadbolt, 2014). Even at the application layer, other cryptographic approaches exist, like Riposte, which anonymously writes content to electronic message boards (Corrigan-Gibbs, 2014), and numerous systems built around dining-cryptographer nets (Chaum, 1988; Corrigan-Gibbs & Ford, 2010). We focus on attacks based on statistical inference and learning by adversaries operating at the application layer.

2. Warm-up Example: Line Graph

We begin by considering the special contact network of a line graph. This example highlights how severely metadata can hurt anonymity; nonetheless, Section 3 illustrates that our seemingly-negative result on lines does not extend to higher-degree trees.

Consider a line graph $G(V, E)$ in which $V = \{0, 1, \ldots, n, n + 1\}$, nodes $s_1 = 0$ and $s_2 = n + 1$ are spies, and $E = \{(i, i + 1) \mid i \in \{0, \ldots, n\}\}$. One of the $n$ nodes between the spies is chosen uniformly at random as a source, denoted by $v^* \in \{1, \ldots, n\}$. When the message reaches a spy $s_i$, the spy collects at least two pieces of metadata: the timestamp $T_{s_i}$ and the parent node $p_{s_i}$ that relayed the message. We let $t_0$ denote the time the source starts propagating the message according to some global reference clock. Let $T_{s_1} = T_1 + t_0$ and $T_{s_2} = T_2 + t_0$ denote the timestamps when the two spy nodes receive the message, respectively. Knowing the spreading protocol and the metadata, the adversary uses the maximum likelihood estimator to optimally estimate the source.

In this section, we first show that under standard diffusion, the probability of source detection scales as $1/\sqrt{n}$. We also show that if spy nodes observed only timestamps and parent nodes, adaptive diffusion would achieve the optimal detection probability of $1/n$. However, adaptive diffusion passes extra metadata, which we call a control packet, to coordinate the message spread (details below). Control packets allow a spy to identify the source with probability 1. To overcome this challenge, we propose a new implementation of adaptive diffusion that provably achieves $1/\sqrt{n}$ (Proposition 2.1). It is an open question if a smaller probability of detection can be achieved on a line.

Standard diffusion. Consider a standard discrete-time random diffusion with a parameter $q \in (0, 1)$ where each uninfected neighbor is infected with probability $q$. The adversary observes $T_{s_1}$ and $T_{s_2}$. Knowing the value of $q$, it computes the ML estimate $\hat{v}_{\text{ML}} = \arg \max_{v \in \{0, \ldots, n\}} P(T_{s_1} - T_{s_2} \mid v)$, which is optimal assuming uniform prior on $v^*$. Since $t_0$ is not known, the adversary can only use the difference $T_{s_1} - T_{s_2} = T_1 - T_2$ to estimate the source. We can exactly compute the corresponding probability of detection; Figure 1 (bottom) illustrates that the posterior (and the likelihood) is concentrated around the ML estimate, and the source can only hide among $O(\sqrt{n})$ nodes. The detection probability correspondingly scales as $1/\sqrt{n}$ (top).

Adaptive diffusion on a line. Adaptive diffusion introduced in (Fanti et al., 2015) on a line is a random message spreading model governed by the location of a virtual source $v_t$ at any (even) time $t$. At time $0$, the source determines either the left or the right neighbor to be the next
then numerically compute the ML estimate \( \hat{v}_{\text{ML}} = \arg \max_{v \in \mathbb{V}} \mathbb{P}_{T_1 - T_2 | V^*}(T_s - T_{s_2} | v) \). Figure 1 shows the posterior is close to uniform (bottom) and the probability detection would scale as \( 1/n \) (top), which is the best one can hope for. Of course, spies do observe control packets, including the information to generate the randomness. This reveals the distance to the true source \( \delta_H(v_T, v^*) \), and the true source is exactly identified with probability 1. We therefore introduce a new implementation (tailored for the line graph) that is robust to control packet information.

**Adaptive diffusion via Pólya’s urn.** The random process governing the virtual source’s propagation is identical to a Pólya’s urn process (Johnson & Kotz, 1977). We propose the following alternative implementation of adaptive diffusion. At \( t = 0 \) the protocol decides whether to pass the virtual source left \( (D = 0) \) or right \( (D = r) \) with probability half. Let \( D \) denote this random choice. Then, a latent variable \( q \) is drawn from the uniform distribution over [0, 1]. Thereafter, at each even time \( t \), the virtual source is passed with probability \( q \) or kept with probability \( 1 - q \). The Bayesian interpretation of Pólya’s urn processes shows that this process is equivalent to the adaptive diffusion process.

Further, in practice, the source could simulate the whole process in advance. The control packet would simply reveal to each node how long it should wait before further propagating the message. Under this implementation, spy nodes only observe timestamps \( T_{s_1} \) and \( T_{s_2} \), parent nodes, and control packets containing the infection delay for the spy and all its descendants in the infection. Given this, the adversary can exactly determine the timing of infection with respect to the start of the infection \( T_1 \) and \( T_2 \), and also the latent variables \( D \) and \( q \). A proof of this statement and the following proposition is provided in Section B.1 of the Supplementary Material. Precisely, we provide an upper bound on the detection probability for such an adversary.

**Proposition 2.1.** When the source is uniformly chosen from \( n \) nodes between two spy nodes, the ML estimator achieves a detection probability upper bounded by

\[
\mathbb{P}(V^* = \hat{v}_{\text{ML}}) \leq \frac{\pi \sqrt{8}}{\sqrt{n}} + \frac{2}{n}.
\]

Equipped with the ML estimator, we can also simulate adaptive diffusion on a line. Figure 1 (top) illustrates that even with access to control packets, the adversary achieves probability of detection scaling as \( 1/\sqrt{n} \) – similar to standard diffusion. For a given value of \( T_1 \), the posterior and the likelihood are concentrated around the ML estimate, and the source can only hide among \( O(\sqrt{n}) \) nodes, as shown in the bottom panel for \( T_1 = 58 \). In the realistic adversarial setting where control packets are revealed at spy nodes, adaptive diffusion can only hide as well as standard diffusion over a line.
3. Main results on $d$-regular trees

In this section, we show that adaptive diffusion hides the source better than diffusion over $d$-regular trees, $d > 2$, and its probability of detection is asymptotically optimal in the degree of the underlying tree. In contrast to the line example, this holds even when the adversary has access to all meta-data. We first present a characterization of the fundamental limit for any spreading protocol. Namely, a lower bound on the probability of detection for any choice of spreading protocol. 

\textbf{Proposition 3.1.} No spreading protocol that infects at least one node can have a probability of detection less than $p$, i.e.

$$\min_{\hat{v}} \max_{\tilde{v}} \mathbb{P}(\hat{v} = \tilde{v}^*) \geq p,$$

where the minimization is over all spreading protocols that infect at least one node and the maximization is over all estimators that are measurable functions over the observed meta-data and the network.

Consider the first-spy estimator, which returns as the estimated source the parent of the first spy to observe the message. Regardless of spreading mechanism, this estimator returns the true source with probability at least $p$; with probability $p$, the first node (other than the true source) to receive the message is a spy. This is illustrated in the top panels in Figure 2 as a fundamental limit. Note that this lower bound is independent of the degree, and we expect this to be tighter for larger degree trees. The reason is that if $d$ is larger, then it is more likely that one of the neighbors of the source is a spy. However, for standard diffusion, the gap between this fundamental limit and the detection probability achieved becomes larger as degree increases. This is illustrated in the top center and top right panels above.

\textbf{Standard diffusion.} The ML estimator under standard diffusion is computationally intractable, and characterizing the probability of detection achieved by such an estimator is also an open problem. We consider a discrete-time diffusion process, in which each infected node passes the message to each neighbor with probability $q$ in each timestep. As $q$ increases, the variance of the associated geometric delay decreases, revealing the true source with higher probability. To lower bound the probability of detection achieved by the best estimator, we consider two heuristic estimators in the numerical experiments: (1) the Gaussian estimator from (Pinto et al., 2012), and (2) the first-spy estimator, which simply returns the parent of the first spy to observe the message. The estimator in (Pinto et al., 2012) is ML when delays are i.i.d. Gaussian, whereas our delays are geometric. We nonetheless expect it to perform well for small $p$; since the distance between spies will be large, the delay

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example.png}
\caption{Adaptive diffusion (AD) theoretical performance for varying $d$ (left). Adaptive diffusion improves over standard diffusion (D) and the gap increases as the degree of the underlying contact network increases (center, right).}
\end{figure}
distribution can be approximated by a Gaussian.

Figure 2 compares the probability of detection and expected hop distance for diffusion (q = 0.7) using heuristic estimators, against adaptive diffusion using the ML estimator. The lower bound for detection probability under standard diffusion (top) is the maximum of the simulated Pinto et al. estimator (Pinto et al., 2012) and the first-spy estimator; the opposite holds for expected hop distance (bottom).

For all p, adaptive diffusion performs better than diffusion, and the gap increases with degree. This effect is sensitive to q for small d, but we show in Section 4 that over real social graphs, the sensitivity to q becomes negligible. We make this observation precise in the following lower bound:

**Proposition 3.2.** Suppose the contact network is a regular tree with degree d. Consider a spy-based adversary and diffusion spreading— that is, in each timestep, each infected node infects each uninfected neighbor independently with probability q. The optimal source estimator achieves a detection probability at least

\[ \max_{\hat{v}} \Pr(\hat{v} = v^*) \geq 1 - (1 - qp)^d, \]

where the maximization is taken over all measurable functions over the observed meta-data and the network.

This bound implies that as degree increases, the probability of detecting the true source of diffusion approaches 1. The proposition also results from the first-spy estimator used in Proposition 3.1. We consider all neighbors of v* that (a) are spies and (b) receive the message at t = 1. If there is at least one such node, then the source is identified with probability 1. Each neighbor of v* meets these criteria with probability pq.

**Adaptive diffusion.** Unlike standard diffusion, the ML estimator is tractable under adaptive diffusion. Further, we can characterize the probability of detection achieved by this ML estimator precisely, and prove it significantly improves over the standard diffusion and achieves the asymptotically optimal performance.

In (Fanti et al., 2015), the authors present two protocols for spreading over trees with degree d > 2: the ‘tree protocol’ and the generalized ‘adaptive diffusion’ algorithm. Against a snapshot adversary, adaptive diffusion provides stronger anonymity guarantees, but against a spy-based adversary, its spreading pattern can lead to deanonymization. However, if the underlying contact network is a tree, the tree protocol is equivalent to adaptive diffusion for a specific parameter choice. This choice always places the source at a leaf of the infected subgraph, and has strong anonymity properties. We focus on this tree protocol, exploiting its simplicity and asymmetric spreading. Further, we show that this protocol achieves provably (asymptotically) optimal source obfuscation, significantly improving upon the standard diffusion. Moving forward, we use the terms ‘tree protocol’ and ‘adaptive diffusion’ interchangeably. We did not analyze the tree protocol over lines because the metadata deterministically reveals the source.

The spreading protocol follows Protocol 4 (tree protocol) from (Fanti et al., 2015); the goal is to build an infected subtree with the true source at one of the leaves. Whenever a node v passes a message to node w, it includes three pieces of metadata: (1) the parent node p_w = v, (2) a binary direction indicator u_w ∈ {↑, ↓}, and (3) the node’s level in the infected subtree m_w ∈ \( \mathbb{N} \). The parent p_w is the node that relayed the message to w. The direction bit u_w flags whether node w is a spine node, responsible for increasing the depth of the infected subtree. The level m_w describes the hop distance from w to the nearest leaf node in the final infected subtree, as \( t \to \infty \). The parent metadata did not appear in the original protocol (Fanti et al., 2015), and is included purely to facilitate the adversary’s source estimation. Even with this extra metadata, the tree protocol achieves asymptotically optimal hiding.

![Figure 3: Message spread using the tree protocol from (Fanti et al., 2015) (left), and the information observed by the spy nodes 3, 7, and 8 (right). Timestamps in this figure are absolute, but they need not be.](image)

At time t = 0, the source chooses a neighbor uniformly at random (e.g., node 1) and passes the message and metadata \( (p_1 = 0, u_1 = \uparrow, m_1 = 1) \). Figure 3 illustrates an example spread, in which node 0 passes the message to node 1. Yellow denotes spine nodes, which receive the message with \( u_w = \uparrow \), and gray denotes those that receive it with \( u_w = \downarrow \). Whenever a node w receives a message, there are two cases. If \( u_w = \uparrow \), node w forwards the message to another neighbor \( z \) chosen uniformly at random with ‘up’ metadata: \( (p_z = w, u_z = \uparrow, m_z = m_w + 1) \). All of w’s remaining neighbors \( z' \) receive the message with ‘down’ metadata: \( (p_{z'} = w, u_{z'} = \downarrow, m_{z'} = m_w - 1) \). In Figure 3, node 1 passes the ‘up’ message to node 2 and the ‘down’ message to node 3. On the other hand, if \( u_w = \downarrow \) and \( m_w > 0 \), node w forwards the message to all its remaining neighbors with ‘down’ metadata: \( (p_z = w, u_z = \downarrow, m_z = m_w - 1) \). If a node receives \( m_w = 0 \), it does not forward the message further. This protocol ensures that the infected subgraph is
a symmetric tree with the true source at one of the leaves; Algorithm 2 in the Supplemental Materials includes a more precise description.

In the spy-based adversarial model, each spy \( s_i \) in the network observes any received messages, the associated metadata, and a timestamp \( T_{s_i} \). Figure 3 (right) illustrates the information observed by each spy node, where spies are outlined in red.

## Algorithm 1 ML Source Estimator for Algorithm 2

1. **Input:** contact network \( G = (V,E) \), spy nodes \( S = \{s_0,s_1 \ldots \} \) and metadata \( s_i : (p_{s_i},m_{s_i},u_{s_i}) \).
2. **Output:** ML source estimate \( \hat{v}_{ML} \).
3. Let \( s_0 \) denote the lowest-level spine spy, with metadata \( (p_{s_0},m_{s_0},u_{s_0}) \).
4. \( V \leftarrow \{v \in V : \delta_H(v,s_0) \leq m_{s_0} \text{ and } p_{s_0} \in \mathcal{P}(v,s_0)\} \)
5. \( E \leftarrow \{(u,v) : (u,v) \in E \text{ and } u,v \in V\} \).
6. Define the feasible subgraph as \( F(V,E) \).
7. \( L \leftarrow \emptyset \) \{Set of eliminated pivot neighbors\}
8. \( K \leftarrow \emptyset \) \{Set of eliminated pivot neighbors\}
9. **for all** \( s \in S \) with \( s \in V \) **do**
10. \( \text{Let } \left[ \begin{array}{cc} h_{s,\ell_s} & 0 \\ h_{s,\ell_s} & 1 \end{array} \right], \left[ \begin{array}{c} |P(s,s_0)| \\ T_{s_0} - T_s \end{array} \right] \)
11. \( \ell_s \leftarrow v \in \mathcal{P}(s,s_0) : \delta_H(s,\ell_s) = h_{s,\ell_s} \)
12. \( k_s \leftarrow v \in \mathcal{P}(s,s_0) : \delta_H(s,k_s) = h_{s,\ell_s} - 1 \)
13. \( L \leftarrow L \cup \{\ell_s\} \) \{Add pivot\}
14. \( K \leftarrow K \cup \{k_s\} \) \{Add pivot neighbor\}
15. **end for**
16. Find the lowest-level pivot: \( \ell_{\text{min}} \leftarrow \arg\min_{\ell \in L} m_\ell \)
17. **for all** \( v \in V \) where \( v \) is a leaf in \( F(V,E) \) **do**
18. \( \text{if } \mathcal{P}(v,\ell_{\text{min}}) \cap K = \emptyset \) **then**
19. \( U \leftarrow \emptyset \cup \{v\} \)
20. **end if**
21. **end for**
22. return \( \hat{v}_{ML} \), drawn uniformly from \( U \).

### ML estimator under adaptive diffusion

The precise ML estimation algorithm is detailed in Algorithm 1. Because adaptive diffusion has deterministic timing, spies only help the estimator discard candidate nodes. We assume the message spreads for an infinite time. There is at least one spy on the spine; consider the first such spy to receive the message, \( s_0 \). This spine spy (along with its parent and level metadata) allows the estimator to specify a feasible subtree in which the true source must lie. In Figure 3, node 8 is on the spine with level \( m_8 = 4 \), so the feasible subtree is rooted at node 5 and contains all the pictured nodes except node 8 (9's children and grandchildren also belong, but are not pictured). Spies outside the feasible subtree do not influence the estimator, because their information is independent of the source conditioned on \( s_0 \)'s metadata. Only leaves of the feasible subtree could have been the source—e.g., nodes 0, 3, 6, and 7, as well as 9’s grandchildren.

The estimator then uses spies within the feasible subtree to prune out candidates. The goal is to identify nodes in the feasible subtree that are on the spine and close to the source. For each spy in the feasible subtree, there exists a unique path to the spine spy \( s_0 \), and at least one node on that path is on the spine; the spies’ metadata reveals the identity and level of the spine node on that path with the lowest level—we call this node a pivot (details in Algorithm 1). For instance, in Figure 3 (right), we can use spies 7 and 8 to learn that node 2 is a pivot with level \( m_2 = 2 \). Estimation hinges on the minimum-level pivot across all spy nodes, \( \ell_{\text{min}} \). In the example, \( \ell_{\text{min}} = 1 \), since spies 3 and 8 identify node 1 as a pivot with level \( m_1 = 1 \). The true source must lie in a subtree rooted at a neighbor of \( \ell_{\text{min}} \), with no spies. In our example, this leaves only node 0, the true source.

### Anonymity properties of adaptive diffusion

Using the described ML estimation procedure, we can exactly compute the probability of detection when adaptive diffusion is run over a \( d \)-regular tree.

**Theorem 1.** Suppose the contact network is a regular tree with degree \( d > 2 \). There is a source node \( v^* \), and each node other than the source is chosen to be a spy node i.i.d. with probability \( p \) as described in the spy model. Against colluding spies attempting to detect the location of the source, adaptive diffusion achieves the following:

(a) The probability of detection is

\[
\mathbb{P}(\hat{v}_{ML} = v^*) = p + \frac{1}{d-2} - \sum_{k=1}^{\infty} \frac{q_k}{(d-1)^k},
\]

where

\[
q_k = (1 - (1 - p)^{(d-1)^k-1/(d-2) - 1}) + (1 - p)^{(d-1)^k+1-1}/(d-2).
\]

(b) The expected distance between the source and the estimate is bounded by

\[
\mathbb{E}[\delta_H(\hat{v}_{ML}, v^*)] \geq 2 \sum_{k=1}^{\infty} k \cdot r_k,
\]

where \( |T_{d,k}| = \frac{(d-1)^k-1}{d-2} \), and

\[
r_k = \frac{1}{d-1} \left((1 - (1 - p)|T_{d,k}|)^{d-1} + (d - 1)(1 - p)|T_{d,k}| - (d - 2)(1 - p)|T_{d,k}|(d-1) - 1\right).
\]

The proof is included in the Supplemental Materials. Briefly, it computes the probability of detection by conditioning on the lowest-level pivot node, \( \ell_{\text{min}} \). Given a pivot,
the probability of detection depends on the number of subtrees rooted at the neighbors of \( \ell_{\min} \) containing no spies. Figure 2 illustrates the theoretical probability of detection and lower bound on expected distance from the true source as a function of the spy probability. We make two key observations:

Asymptotically optimal probability of detection: As tree degree \( d \) increases, the probability of detection converges to the degree-independent fundamental limit in Proposition 3.1, i.e., \( \mathbb{P}(V^\star = \hat{v}_{\text{ML}}) = p \). This is in contrast to diffusion, whose probability of detection tends to 1 asymptotically in \( d \). The median Facebook user has 200 friends (Smith, 2014), so these asymptotics have practical implications, as we will see in Section 4.

Expected hop distance asymptotically increasing: We observe empirically that for regular diffusion, \( \mathbb{E}[\delta_H(\hat{v}_{\text{ML}}, v^\star)] \) approaches 0 as \( d \) increases. On the other hand, for adaptive diffusion with a fixed \( p > 0 \), as \( d \to \infty \), \( \lim_{d \to \infty} \sup \mathbb{E}[\delta_H(\hat{v}_{\text{ML}}, v^\star)] = 2(1 - p) \). This holds because with probability \( (1 - p) \), the first node is not a spy, but with probability approaching 1 for \( d \) large enough, the first node on the spine will be a pivot node. Since the source is always a leaf, the distance from the estimate to the source will be at most 2 with probability approaching \( 2(1 - p) \). Figure 2 includes the line \( 2(1 - p) \) for reference, and we observe that as \( d \to \infty \), \( \mathbb{E}[\delta_H(\hat{v}_{\text{ML}}, v^\star)] \) appears to converge precisely to this line. However, for a fixed \( d \), Theorem 1 implies that as \( p \to 0 \), \( \mathbb{E}[\delta_H(\hat{v}_{\text{ML}}, v^\star)] \to \infty \).

4. Generalizations

Graphs. Here, we consider irregular, finite graphs that arise in real contact networks. Regardless of spreading protocol, the message always propagates over a tree superimposed on the underlying contact network. The probability of detection over irregular trees is therefore tied to performance over general graphs. ML estimation over irregular trees is more straightforward than in (Fanti et al., 2015), primarily because we use the specialized tree protocol that places the source at a leaf node.

Proposition 4.1. Suppose the underlying contact network \( G(V, E) \) is an irregular tree with the degree of each node larger than one. One node \( v^\star \in V \) starts spreading a message at time \( T = 0 \) according to Protocol 2. Each node \( v \in V \), \( v \neq v^\star \) is a spy with probability \( p \). Let \( U \) denote the set of feasible candidate sources obtained by estimation Algorithm 1. Then the maximum likelihood estimate of \( v^\star \) given \( U \) is \( \hat{v}_{\text{ML}} = \arg\max_{u \in U} \frac{1}{\deg(u)} \prod_{v \in \mathcal{P}(u, \ell_{\min}) \setminus \{u, \ell_{\min}\}} \frac{1}{\deg(v) - 1}, \)

where \( \ell_{\min} \) is the lowest-level pivot node, \( \mathcal{P}(u, \ell_{\min}) \) is the unique shortest path between \( u \) and \( \ell_{\min} \), and \( \deg(u) \) denotes the degree of node \( u \). (Proof in Section D, Supplemental Materials.)

This ML estimator allows us to evaluate adaptive diffusion over real dataset (social graph connections among 10,000 Facebook users (Viswanath et al., 2009)) against a spy-based adversary. We simulate adaptive diffusion and regular diffusion for \( q \in \{0.1, 0.5\} \). We evaluated diffusion with the first-spy estimator, and adaptive diffusion with a slightly modified version of the ML estimator in Proposition 4.1, that accounts for cycles in the underlying graph. Figure 4 lists the probability of detection averaged over 200 trials, for \( p \) up to 0.15. We chose this value because at its height, the Stasi employed 11 percent of the population as spies (Koehler, 1999). Not only does adaptive diffusion hide the source better than diffusion, its probability of detection in practice is close to the fundamental lower bound of \( p \). This is likely because the mean node degree in the dataset is 25, so high-degree asymptotics are significant. While adaptive diffusion can never reach all nodes in a tree, cycles in the Facebook graph allow it to reach 81% of nodes within 20 timesteps.

Adversaries. The spy-based and snapshot adversarial models capture very different behavior. The spy-snapshot model considers a natural combination of both: at a certain time \( T \), the adversary collects both types of metadata and infers the source. Notably, this stronger model does not significantly impact the probability of detection as time increases. The snapshot helps detection when there are few spies by revealing which nodes are true leaves. This effect is most pronounced for small \( T \) and/or small \( p \). The exact analysis of the probability of detection at \( T \) is given in Equation (15) in the Supplementary material, and Figure 6 illustrates the tradeoff between snapshots and spy nodes.

References

Corrigan-Gibbs, H. Riposte: An anonymous messaging system handling millions of users. 2014.


Luo, W., Tay, W., and Leng, M. How to identify an infection source with limited observations. 2013.


Shadbolt, P. FireChat in Hong Kong: How an app tapped its way into the protests. CNN, 2014.


Sterbenz, Christina. Cops are creating totally bogus facebook profiles just so they can arrest people. Business Insider, 2013.


Winter, P. and Lindskog, S. How the great firewall of china is blocking Tor. FOCI, 2012.


A. Introduction

Figure 5 shows that under a discrete-time diffusion model, probability of detection increases with time and degree of the underlying graph. In the diffusion model used here, each node propagates the message to each of its neighbors independently with probability $q = 0.7$ in each time step. We used the Gaussian estimator from (Pinto et al., 2012), which is suboptimal for this spreading model; as such, the plotted curves are lower bounds on the probability of detection using an ML estimator.

Figure 5. Probability of detection (suboptimal) when a message is spread using diffusion over a $d$-regular tree. Detection becomes more accurate as time and underlying graph degree increase.

B. Line Analysis

B.1. Proof of Proposition 2.1

The control packet at spy node $s_1$ includes the amount of delay at $s_1 = 0$ and all descendants of $s_1$, which is the set of nodes $\{-1, -2, \ldots\}$. The control packet at spy node $s_2$ includes the amount of delay at $s_2 = n + 1$ and all descendants of $s_2$, which is the set of nodes $\{n + 2, n + 3, \ldots\}$. Given this, it is easy to figure out the whole trajectory of the virtual source for time $t \geq 1$. Since the virtual source follows i.i.d. Bernoulli trials with probability $q$, one can exactly figure out $q$ from the infinite Bernoulli trials. Also the direction $D$ is trivially revealed.

To lighten the notation, suppose that $T_1 \leq T_2$ (or equivalently $T_{s_1} \leq T_{s_2}$). Now using the difference of the observed time stamps $T_{s_2} - T_{s_1}$ and the trajectory of the virtual source between $T_{s_1}$ and $T_{s_2}$, the adversary can also learn the time stamp $T_1$ with respect to the start of the infection. Further, once the adversary learns $T_1$ and the location of the virtual source $v_{T_1}$, the timestamp $T_2$ does not provide any more information. Hence, the adversary performs ML estimate using $T_1$, $D$ and $q$. Let $B(k, n, q) = \binom{n}{k}q^k(1 - q)^{n-k}$ denote the pmf of the binomial distribution. Then, the likelihood can be computed for $T_1$ as

\[
\begin{aligned}
\mathbb{P}_{T_1|V^*, Q, D}(t_1|v^*, q, \ell) &= \\
qB(v^* - \frac{1}{2} - \frac{1}{2}, 2)B(v^* - \frac{1}{2}, 2)B(v^* - \frac{1}{2} - 1, 2) &\text{if } t_1 \text{ even} \\
B(v^* - \frac{1}{2} - 1, 2) &\text{if } t_1 \text{ odd}
\end{aligned}
\]

(2)

\[
\begin{aligned}
\mathbb{P}_{T_1|V^*, Q, D}(t_1|v^*, q, r) &= \\
(1 - q)B(v^* - \frac{1}{2} - v^*, 2) &\text{if } t_1 \text{ even} \\
0 &\text{if } t_1 \text{ odd}
\end{aligned}
\]

(3)

This follows from the construction of the adaptive diffusion. The protocol follows a binomial distribution with parameter $q$ until $(T_1 - 1)$. At time $T_1$, one of the following can happen: the virtual source can only be passed (the first equation in (2)), it can only stay (the second equation in (3)), or both cases are possible (the second equation in (2)).

Given $T_1$, $Q$ and $D$, which are revealed under the adversarial model we consider, the above formula implies that the posterior distribution of the source also follows a binomial distribution. Hence, the ML estimate is the mode of a binomial distribution with a shift, for example when $t_1$ is even, ML estimate is the mode of $2 + (t_1/2) + Z$ where $Z \sim \text{Binom}((t_1/2) - 2, q)$. The adversary can compute the
ML estimate:
\[
\hat{\theta}_{\text{ML}} = \begin{cases}
\frac{T_1 + 2}{2} + \left[ q \left( \frac{T_1 - 2}{2} \right) \right] & \text{if } T_1 \text{ even and } D = \ell , \\
\frac{T_1 + 3}{2} + q \left( \frac{T_1 - 1}{2} \right) & \text{if } T_1 \text{ odd and } D = \ell , \\
1 + \left[ (1 - q) \left( \frac{T_1 - 1}{2} \right) \right] & \text{if } T_1 \text{ odd and } D = r .
\end{cases}
\tag{4}
\]

Together with the likelihoods in Eqs. (2) and (3), this gives
\[
P_{T_1, D | v^*, Q}(t_1, r, \hat{\theta}_{\text{ML}} | q) = \begin{cases}
\frac{1}{2} (1 - q) B \left( \frac{t_1 - 1}{2} - v^*, \frac{t_1 - 3}{2} - q \right) I(t_1 \text{ is odd}) & \text{if } T_1 \text{ even and } D = \ell , \\
\frac{1}{2n} (1 - q) B \left( \frac{t_1 - 1}{2} - \hat{\theta}_{\text{ML}}, \frac{t_1 - 3}{2} - q \right) I(t_1 \text{ is odd}) & \text{if } T_1 \text{ odd and } D = \ell , \\
\frac{1}{2n} (1 - q) B \left( \frac{t_1 - 1}{2} - \hat{\theta}_{\text{ML}}, \frac{t_1 - 3}{2} - q \right) I(t_1 \text{ is odd}) & \text{if } T_1 \text{ odd and } D = r \text{.}
\end{cases}
\tag{5}
\]

Similarly, we can show that
\[
P(D = r, V^* = \hat{\theta}_{\text{ML}}, T_1 \text{ odd} | Q = q) \leq \frac{(1 - q) \sqrt{2}}{2n \sqrt{q(1 - q)}} \sqrt{8 \left[ \frac{n - 1}{2} \right] + \frac{1 - q}{2n}} .
\tag{8}
\]

C. Regular Tree Analysis

C.1. Algorithms

Algorithm 2 Spreading on a tree
1: Input: contact network \( G = (V, E) \), source \( v^* \), time \( T \)
2: Output: infected subgraph \( G_T = (V_T, E_T) \)
3: \( V_0 \leftarrow \{v^*\} \)
4: \( m_{w, v} \leftarrow 0 \text{ and } u_{w, v} \leftarrow \uparrow \)
5: \( v^* \text{ selects one of its neighbors } w \text{ at random} \)
6: \( V_1 \leftarrow V_0 \cup \{w\} \)
7: \( m_w \leftarrow 1 \text{ and } u_w \leftarrow \uparrow \)
8: \( t \leftarrow 2 \)
9: for \( t \leq T \) do
10: for all \( v \in V_{t-1} \text{ with uninfected neighbors and } m_v > 0 \text{ do} \)
11: if \( u_v = \uparrow \text{ then} \)
12: \( v \text{ selects one of its uninfected neighbors } w \text{ at random} \)
13: \( V_t \leftarrow V_{t-1} \cup \{w\} \)
14: \( m_w \leftarrow m_w + 1 \text{ and } u_w \leftarrow \uparrow \)
15: end if
16: for all uninfected neighboring nodes \( z \) of \( v \text{ do} \)
17: \( V_t \leftarrow V_{t-1} \cup \{z\} \)
18: \( u_z \leftarrow \downarrow \text{ and } m_z \leftarrow m_v - 1 \)
19: end for
20: end for
21: \( t \leftarrow t + 1 \)
22: end for

C.2. Proof of Theorem 1

We begin by expanding some points regarding the ML estimator in Algorithm 1 that were omitted in Section 3. First, note that it is possible to derive an ML estimate without requiring the presence of a spine spy; the estimator described here uses a spine spy purely for ease of exposition. The omitted details are: (1) given a spy node \( s \), how does the estimator find that spy node’s pivot \( \ell_s \)? (2) Why does timing information enable the estimator to disregard any subtree neighboring \( \ell_{min} \) that contains at least one spy?

To answer the first question, consider the first spine spy \( s_0 \) and all spies in the feasible subtree. For each spy \( s \) in the feasible subtree (none of which lies on the spine), there exists a unique path between \( s \) and \( s_0 \). There exists a unique node on this path that is both part on the spine and closer to the true source than any other node in the path—this is precisely the pivot node. The estimator uses the observed metadata to infer the pivot, as well as its level in the infected subtree, for each spy in the feasible subtree. This
inference proceeds by solving a system of equations:

\[
\begin{align*}
    h_{s, \ell_s} + h_{\ell_s, s_0} &= |P(s, s_0)| \\
    h_{\ell_s, s_0} - h_{s, \ell_s} &= T_{s_0} - T_s
\end{align*}
\]

where \( P(s, s_0) \) denotes the path between \( s \) and \( s_0 \), \( h_{s, \ell_s} = \delta_H(s, \ell_s) \) denotes the distance from spy \( s \) to the pivot node \( \ell_s \), and \( h_{\ell_s, s_0} \) is equal to \( \delta_H(\ell_s, s_0) \) by construction. This system of equations always has a unique solution; hence the uniqueness of \( \ell_s \) given \( s \) and \( s_0 \). The first equation holds by construction. The second equation holds because conditioned on the time at which the pivot receives the message \( T_{\ell_s} \), \( s_0 \) receives the message at time \( T_{\ell_s} + h_{\ell_s, s_0} \), and \( s \) receives it at \( T_{\ell_s} + h_{s, \ell_s} \).

Let \( L \) denote the set of pivots corresponding to each spy in the feasible subtree; in the example in Figure 3, \( L = \{1, 2\} \). Define \( \ell_{\min} = \arg\min_{\ell \in L} \ell \). That is, \( \ell_{\min} \) denotes the pivot closest to the true source in hop distance, i.e., whose level is lowest. Now consider the subtree of depth \( m_{\ell_{\min}} - 1 \) rooted at the neighbors of \( \ell_{\min} \). The subtree including \( s_0 \) cannot contain the true source because we know the message traveled from \( \ell_{\min} \) to \( s_0 \). The source must therefore lie in one of the remaining \( d - 1 \) neighbor subtrees, which we refer to as candidate subtrees.

We now argue that the estimator can rule out any candidate subtree of \( \ell_{\min} \) that contains at least one spy node. Suppose otherwise: there is a candidate subtree containing a spy \( s \), and the source \( v^* \) is contained in that subtree. Then the path \( P(v^*, s) \) cannot pass through \( \ell_{\min} \) because \( \ell_{\min} \) does not belong to any of its own neighboring subtrees by construction. Then there must exist some node \( \ell' \) on the spine such that \( |P(\ell', s)| < |P(\ell_{\min}, s)| \). But this is a contradiction because \( \ell_{\min} \) is chosen as the minimum-level pivot across all spies, and each spy has a unique pivot on the spine.

Since we can now rule out candidate subtrees with at least one spy, let \( X + 1 \), \( X \in \mathbb{N} \) be the number of candidate subtrees containing no spies. We use this notation because there will always be at least one candidate subtree with no spies (the one containing the true source). In Figure 3, \( X = 0 \). Thus, the ML estimator chooses one of the leaves in the remaining \( X + 1 \) candidate subtrees uniformly at random. All remaining nodes in \( V \setminus U \) have likelihood 0.

**Probability of Detection:** We condition on the lowest-level pivot node, \( \ell_{\min} \), giving \( P(v_{\text{ML}} = v^*) = \sum_{\ell_{\min}} P(v_{\text{ML}} = v^* | \ell_{\min}) P(\ell_{\min}) \). Since \( \ell_{\min} \) lies on the spine, this is equivalent to conditioning on the distance of \( \ell_{\min} \) from the true source.

\[
P(v_{\text{ML}} = v^*) = \sum_{k=1}^{\infty} \left( 1 - p \right)^{\left[ T_{d,k} - 1 \right]} p \left( \frac{\ell_{\min} (k \text{th spine node}) \text{ is a spy}}{\ell_{\min} (k \text{th spine node}) \text{ not a spy}} \right) \left( \frac{1}{X + 1} \right) |T_{d,k}| - 2)
\]

where \( X \sim \text{Binom}(d - 2, (1 - p)|T_{d,k}|) \). \( |T_{d,k}| = (d - 1)k - 1 \) is the number of nodes in each candidate subtree for a pivot at level \( k \), and \( |\partial T_{d,k}| = (d - 1)k - 1 \) is the number of leaf nodes in each candidate subtree. Let \( w = (1 - p) \). We have that

\[
\mathbb{E}_X \left[ \left( \frac{1}{X + 1} \right) |\partial T_{d,k}| \right] = \frac{1}{w |\partial T_{d,k}|} \left( \frac{1}{d - 1} (1 - (1 - p |\partial T_{d,k}|)^{d-1}) - \left( \frac{1}{d - 1} w |\partial T_{d,k}| (d - 2) \right) \right)
\]

where the last line results from the expression for the expectation of \( 1/(1 + X) \) when \( X \) is binomially-distributed. Namely if \( X \sim \text{Binom}(\tilde{n}, \tilde{p}) \), then \( \mathbb{E}[1/(X + 1)] = 1/(\tilde{n} + 1)\tilde{p}(1 - (1 - \tilde{p})^{\tilde{n} + 1}) \). Simplifying gives

\[
P_D = \sum_{k=1}^{\infty} \left( \frac{1}{(d - 1)k} (d - 2) + \left( \frac{1}{(d - 1)k} \right)^{d - 1} - \left( \frac{1}{(d - 1)k} \right)^{d - 1} \right)
\]

**Expected hop distance:** In the main paper, we lower bounded the expected hop distance by assuming that the estimator guesses the source exactly whenever (a) the pivot \( \ell_{\min} \) is a spy node or (b) the estimator chooses the correct candidate subtree. Therefore, if the pivot \( \ell_{\min} \) is at level \( k \), we only consider estimates that are exactly \( 2k \) hops away.

\[
\mathbb{E}[|T_{d,k+1} - 1| = |T_{d,k}| \cdot (d - 1)]
\]
The estimator chooses an incorrect candidate subtree with probability $X/(X + 1)$.

$$
\mathbb{E} [\delta_H (\hat{v}_{\text{ML}}, v^*)] \geq \sum_{k=1}^{\infty} 2k(1 - p)^{T \cdot k} \cdot \mathbb{E} \left[ X_k \cdot \mathbb{I} (X_k \neq d - 2) \right] / (X_k + 1). \quad (14)
$$

If $X_k \sim \text{Binom}(\tilde{n}, \tilde{p})$, where $\tilde{n}$ and $\tilde{p}$ depend on $d$ and $k$, we have

$$
\mathbb{E} [X_k \cdot \mathbb{I} (X_k \neq \tilde{n})] = \frac{1}{(\tilde{n} + 1)\tilde{p}} \left[ (1 - \tilde{p})^\tilde{n} + \tilde{p}(1 - (1 - \tilde{p})^\tilde{n} + \tilde{n}) - 1 - \tilde{n}\tilde{p}^{\tilde{n}+1} \right]
$$

Simplifying and substituting $\tilde{p} = (1 - p)^{T \cdot k}$ and $\tilde{n} = d - 2$ gives the expression in the theorem.

Note that this bound is trivially 0 for $d = 3$, since we ignore all nodes in the correct candidate subtree; when $d = 3$, the source’s candidate subtree is the only valid option if $\ell_{\text{min}}$ is not a spy. However, for a fixed $p$ with $d \rightarrow \infty$, this lower bound approaches the upper bound of $2(1 - p)$.

Obtaining a tighter bound is straightforward, but increases the complexity of the expression. These tighter bounds were used for the plots in the main paper. A tighter bound results from considering the cases when (a) the pivot $\ell_{\text{min}}$ is a spy node or (b) the estimator chooses the correct candidate subtree. In both cases, we ignore all but the most distant estimates. For instance, if $\ell_{\text{min}}$ is on the spine at level $k$, then the estimate will be at most $2(k - 1)$ hops away. Using this rule for both cases (a), we compute the probability of selecting one of the most distant options:

$$
a_k \equiv \frac{d - 2}{d - 1} (1 - p)^{T \cdot k} \cdot (d - 1)
$$

and for case (b):

$$
b_k \equiv p \frac{d - 2}{d - 1} (1 - p)^{T \cdot k} \cdot (d - 1)
$$

Overall, we get a lower bound of

$$
\mathbb{E} [\delta_H (\hat{v}_{\text{ML}}, v^*)] \geq \sum_{k=1}^{\infty} 2(kr_k + (k - 1)(a_k + b_k))
$$

### D. D.2. Analysis of spy-snapshot adversarial model

We follow closely the proof of Theorem 1 in Appendix C.2. Given a snapshot at a certain even time $T$, if there are at least two spy nodes infected at time $T$, then the adversary can perform the exact same estimation as he did with only spy nodes with $T \rightarrow \infty$. We only need to carefully analyze what happens when there are only one spy infected or no spies are infected.

We condition on the lowest-level pivot node, $\ell_{\text{min}}$, giving $\mathbb{P} (\hat{v}_{\text{ML}} = v^*) = \sum_{\ell_{\text{min}}} \mathbb{P} (\hat{v}_{\text{ML}} = v^*) \cdot \mathbb{P} (\ell_{\text{min}})$. Since $\ell_{\text{min}}$ lies on the spine, this is equivalent to conditioning on the distance of $\ell_{\text{min}}$ from the true source. We first define $|S_{d,T}| = 1 + d(2d - 1)^{T/2} - 1/(d - 2)$ as the number of nodes infected at time $T$, and $|\partial S_{d,T}| = d(2d - 1)^{(T/2) - 1}$ as the number of leaves in the infected subtree.

### D. Irregular Tree Analysis

#### D.1. Proof of Proposition 4.1

All nodes in $V \setminus U$ have likelihood zero, as discussed in the proof of Theorem 1 (recall that $V$ denotes the set of all nodes, and $U$ denotes the set of candidate nodes). The only randomness in adaptive diffusion spreading occurs when a spine node with uninfected neighbors decides which of its neighbors will be added to the spine next. Thus, the (log) likelihood of a candidate source is the sum of the (log) likelihoods of all candidate spine nodes starting at the candidate source. Regardless of which node $u \in U$ is the true source, the spine must pass through $\ell_{\text{min}}$; since there is a unique path between $u$ and $\ell_{\text{min}}$ over trees, the only feasible sequence of candidate spine nodes starting at candidate $u$ must traverse $\mathcal{P} (u, \ell_{\text{min}})$. By the Markov property of the adaptive diffusion spreading mechanism, we only need to consider the likelihood of a candidate spine prior to reaching $\ell_{\text{min}}$. The propagation of the spine thereafter is conditionally independent of the true source, and therefore equally-likely for all candidates. The maximum likelihood estimator must therefore compute the likelihood of each such candidate sub-spine $\mathcal{P} (u, \ell_{\text{min}})$. Since each spine node $v$ chooses one of its uninfected neighbors uniformly at random to be the next spine node, the choice of next spine node is simply $1/\deg(v) - 1$. Similarly, the likelihood of candidate source $u$ sending the spine in a particular direction is $1/\deg(u)$. The overall likelihood of a candidate is therefore proportional to the product of these degree terms.
Figure 6. Probability of detection under the spies+snapshot adversarial model. As estimation time and tree degree increase, the effect of the snapshot on detection probability vanishes.

\[
\mathbb{P}(\hat{v}_{ML} = v^*) = \frac{(1 - p)^{|S_d|^{-1}}}{|\partial S_d|} + \\
\sum_{k=1}^{T/2} \left\{ \frac{(1 - p)^{|T_{d,k}|^{-1}} p}{|\partial T_{d,k}|} + \\
(1 - p)^{|T_{d,k}|} (1 - (1 - p)^{|S_d|^{-1}}) |T_{d,k}| |\partial T_{d,k}| \mathbb{E}_{X} \left[ \mathbb{I}(X \neq d - 2) \frac{1}{(X + 1)} \right] + \\
(1 - p)^{|S_d|^{-1}} (1 - |T_{d,k}|) |\partial T_{d,k}| \mathbb{E}_{X} \left[ \mathbb{I}(X \neq d - 2) \frac{1}{|\partial S_d| - (d - 2 - X)} \right] \right\},
\]

where \( X \sim \text{Binom}(d - 2, (1 - p)^{|T_{d,k}|}) \), \(|T_{d,k}| = \frac{(d - 1)^{k-1}}{d-2}\) is the number of nodes in each candidate subtree for a pivot at level \( k \), and \(|\partial T_{d,k}| = (d - 1)^{k-1}\) is the number of leaf nodes in each candidate subtree.