Metadata-Conscious Anonymous Messaging

Giulia Fanti, *Member, IEEE*, Peter Kairouz, *Student Member, IEEE*, Sewoong Oh, *Member, IEEE*, Kannan Ramchandran, *Fellow, IEEE*, and Pramod Viswanath, *Member, IEEE*

***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 [1]. 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.**

***Index Terms*—Communication networks, privacy.**

I. 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 [2], [3], 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 [4], [5].

The privacy implications of social media are gaining attention; in response, a number of *anonymous social networks* have cropped up recently, including Whisper [6], Yik Yak [7], Blind [8] and the now-defunct Secret [9]. 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 [10]. 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 [11], [12] 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 [13].

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 [11] and source obfuscation [1]. 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 Section IV.

Under the spy-based adversarial model, we suppose each node other than the source is a spy independently with probability p.

Manuscript received February 15, 2016; revised July 4, 2016; accepted July 22, 2016. Date of publication September 2, 2016; date of current version November 4, 2016. This work was supported by NSF CISE awards CCF-1422278, CCF-1553452, and CCF-1409135, SaTC award CNS-1527754, ARO W911NF-14-1-0220, and AFOSR 556016. This paper was presented in part at the 33rd International Conference on Machine Learning 2016; the present work includes full proofs and more detailed explanations of key concepts. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Venkatesh Saligrama.

G. Fanti, P. Kairouz, and P. Viswanath are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 94709 USA (e-mail: fanti@illinois.edu; kairouz2@illinois.edu; pramodv@illinois.edu).

S. Oh is with the Industrial and Enterprise Systems Engineering Department, University of Illinois at Urbana-Champaign, Champaign, IL 94709 USA (e-mail: swoh@illinois.edu).

K. Ramchandran is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: kannanr@eecs.berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSIPN.2016.2605761

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 [12], [14] 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 (Fig. 1). In the diffusion spreading used to generate Fig. 1, 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 [12], which is suboptimal for this spreading model; as such, the plotted curves are lower bounds on the probability of detection using an ML estimator. 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 [1]. 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 [1] 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 2).
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 III.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 III.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.

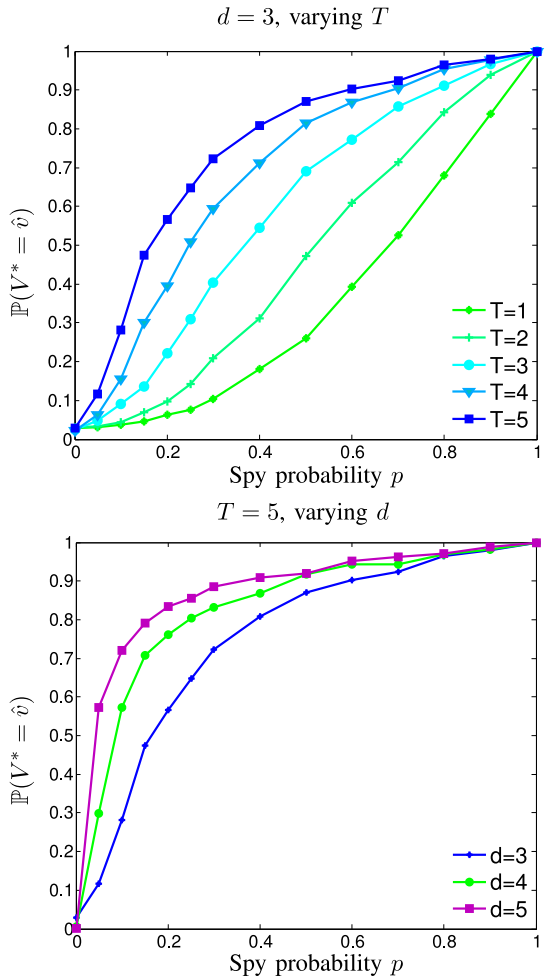

Fig. 1. Probability of detection (computed with a non-ML estimator) when a message is spread using diffusion over a d-regular tree. Detection becomes more accurate as time and underlying graph degree increase.

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 [11]. 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 [15]–[22].

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 infection 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 [1]. 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 [12], [19]. However, ML estimation is known to be NP-hard [14], and analyzing the probability of detection is also challenging. Fig. 1 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 an active research area [23], as is privacy-preserving distributed data storage in P2P systems [24]. 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) [25], [26]. Lower-level protections may be more appropriate against such an opponent, harnessing factors like physical proximity of users [27]. Even at the application layer, other cryptographic approaches exist, like Riposte, which anonymously writes content to electronic message boards [28], and numerous systems built around dining-cryptographer nets [29], [30]. We focus on attacks based on *statistical inference and learning* by adversaries operating at the application layer.

II. 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 III illustrates that our seemingly-negative result on lines does not extend to higher-degree trees.

Consider a line graph $G(V, E)$ in which $V = \mathbb{Z}$, nodes $s_1 = 0$ and $s_2 = n + 1$ are spies, and $E = \{(i, i+1) \mid i \in \mathbb{Z}\}$. 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 II.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 $\hat{v}_{\mathrm{ML}} = \arg\max_{v \in [n]} \mathbb{P}_{T_1 - T_2 | V^*}(T_{s_1} - T_{s_2} | v)$, 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; Fig. 2 (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 [1] 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 virtual source with equal probability. The message is propagated to the chosen node at time $t = 1$. At $t = 2$, the new virtual source v_2 propagates the message to its uninfected neighbor. At this point, three nodes are infected, with the virtual source v_2 at the center. At any given even time t, the infected subgraph is a subset of $t + 1$ nodes, centered around the virtual source v_t. At each even time t, the protocol has two options: keep the virtual source where it is, or pass it to the only neighbor who has not yet been a virtual source. The protocol keeps the current virtual source with probability $\frac{2\delta_H(v_t, v^*)}{t+2}$, where $\delta_H(v_t, v^*)$ denotes the hop distance between the source and the virtual source, and passes it otherwise. The control packet therefore contains two pieces of information: $\delta_H(v_t, v^*)$ and t. In the next two time steps, the message spreads

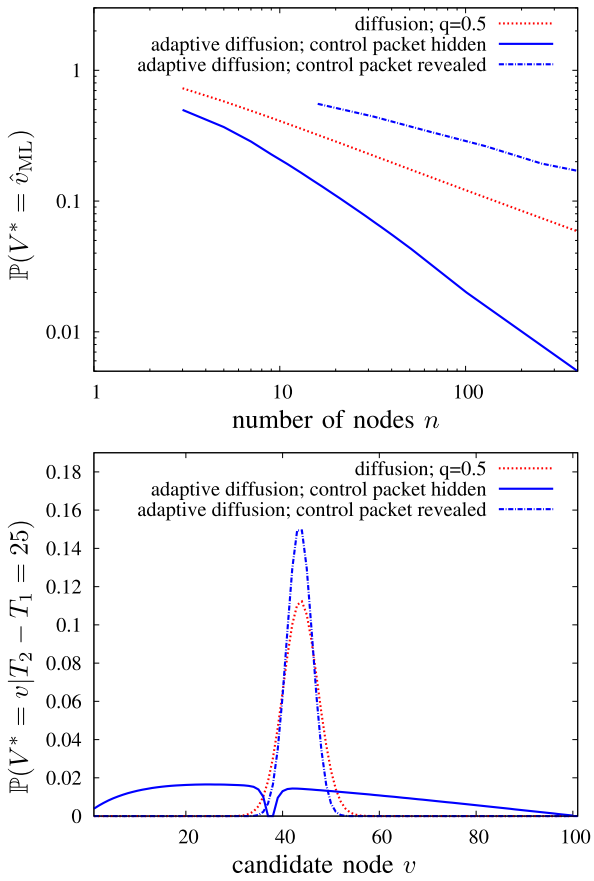

Fig. 2. Comparisons of probability of detection as a function of n (top) and the posterior distribution of the source for an example with $n = 101$ and $T_2 - T_1 = 25$ (bottom). The line with 'control packet revealed' uses the Pólya's urn implementation.

in such a way that two more nodes are infected, and the virtual source is again at the center of the infected subgraph. This choice of virtual-source-spreading probability is optimal against a snapshot adversary, guaranteeing perfect obfuscation of the source.

Suppose spy nodes only observed timestamps and parent nodes but *not* control packets. The adversary could then numerically compute the ML estimate $\hat{v}_{\mathrm{ML}} = \arg\max_{v\in[n]} \mathbb{P}_{T_1-T_2|V^*}(T_{s_1} - T_{s_2}|v)$. Fig. 2 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 [31]. We propose the following alternative implementation of adaptive diffusion. At $t = 0$ the protocol decides whether to pass the virtual source left ($D = \ell$) 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 VI-A1. Precisely, we provide an upper bound on the detection probability for such an adversary.

Proposition II.1: When the source is uniformly chosen from n nodes between two spy nodes, the ML estimator achieves a detection probability upper bounded by

$$\mathbb{P}\left(V^* = \hat{v}_{\mathrm{ML}}\right) \leq \frac{\pi\sqrt{8}}{\sqrt{n}} + \frac{2}{n}\,.$$

Equipped with the ML estimator, we can also simulate adaptive diffusion on a line. Fig. 2 (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.

III. 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 metadata. 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.

Proposition III.1: No spreading protocol that infects at least one node can have a probability of detection less than p, i.e.

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

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 Fig. 3 as a

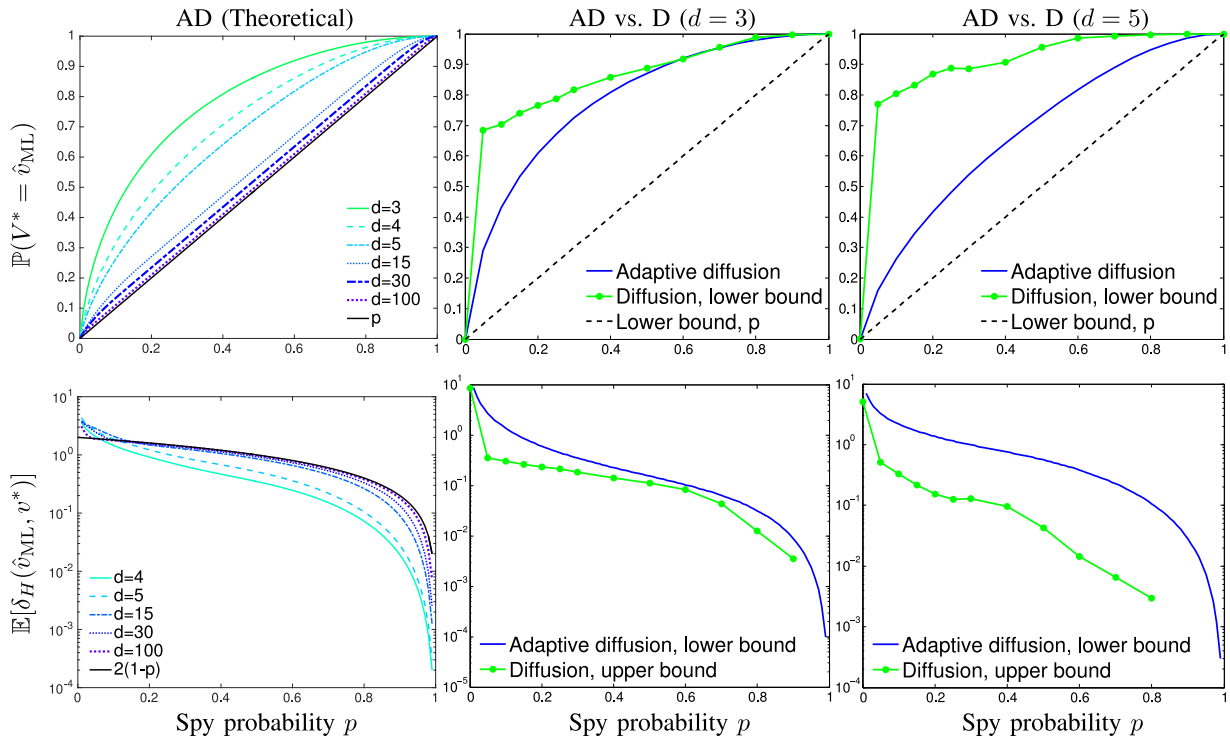

Fig. 3. Adaptive diffusion (AD) theoretical performance for varying d (left). Adaptive diffusion improves over standard diffusion (D) and the gap increases as the degree of the underlying contact network increases (center, right).

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.

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 [12], and (2) the first-spy estimator, which simply returns the parent of the first spy to observe the message. The estimator in [12] 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 distribution can be approximated by a Gaussian.

Fig. 3 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 [12] 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 IV that over real social graphs, the sensitivity to q becomes negligible. We make this observation precise in the following lower bound:

Proposition III.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

$$\max_{\hat{v}} \mathbb{P}(\hat{v} = v^*) \geq 1 - (1 - qp)^d \ ,$$

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 III.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 [1], 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

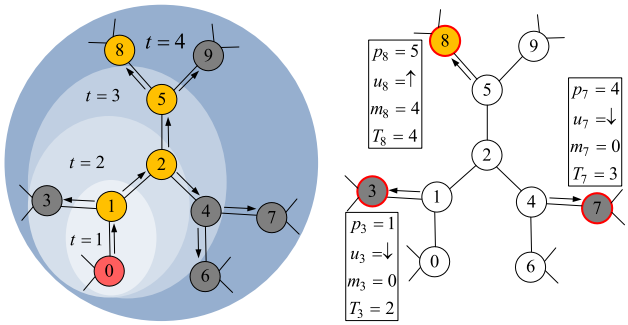

Fig. 4. Message spread using the tree protocol from [1] (left), and the information observed by the spy nodes 3, 7, and 8 (right). Timestamps in this figure are absolute, but they need not be.

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 the tree protocol, exploiting its simplicity and asymmetric spreading. Further, we show that this protocol achieves provably (asymptotically) optimal source obfuscation, significantly improving upon 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 [1]; 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 $u_w \in \{\uparrow, \downarrow\}$, and (3) the node's *level* in the infected subtree $m_w \in \mathbb{N}$. 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 [1], and is included purely to facilitate the adversary's source estimation. Even with this extra metadata, the tree protocol achieves asymptotically optimal hiding.

At time $t = 0$, the source chooses a neighbor uniformly at random (e.g., node 1) and passes the message and metadata $(p_1 = 0, u_1 = \uparrow, m_1 = 1)$. Fig. 4 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 chooses another neighbor z uniformly at random and forwards the message 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: $(p_{z'} = w, u_{z'} = \downarrow, m_{z'} = m_w - 1)$. For instance, in Fig. 4, 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. Algorithm 1 describes this process more precisely.

Algorithm 1: Spreading on a tree.

Input: contact network $G = (V, E)$, source v^*, time T
Output: infected subgraph $G_T = (V_T, E_T)$
$V_0 \leftarrow \{v^*\}$
$m_{v^*} \leftarrow 0$ and $u_{v^*} \leftarrow \uparrow$
v^* selects one of its neighbors w at random
$V_1 \leftarrow V_0 \cup \{w\}$
$m_w \leftarrow 1$ and $u_w \leftarrow \uparrow$
$t \leftarrow 2$
for $t \leq T$ **do**
  for all $v \in V_{t-1}$ with uninfected neighbors and $m_v > 0$ **do**
    if $u_v = \uparrow$ **then**
      v selects one of its uninfected neighbors w at random
      $V_t \leftarrow V_{t-1} \cup \{w\}$
      $m_w \leftarrow m_w + 1$ and $u_w \leftarrow \uparrow$
    end if
    for all uninfected neighboring nodes z of v **do**
      $V_t \leftarrow V_{t-1} \cup \{z\}$
      $u_z \leftarrow \downarrow$ and $m_z \leftarrow m_v - 1$
    end for
  end for
  $t \leftarrow t + 1$
end for

Observe that adaptive diffusion ensures that the infected subgraph is a balanced tree with the true source at one of the leaves. Moreover, unlike regular diffusion, the message does not reach all the nodes in the network under adaptive diffusion (even when $T = \infty$). Even though this may seem like a fundamental drawback for adaptive diffusion, it can be shown that the infected subgraph has a size proportional to $(d-1)^{T/2}$ on regular trees (compared to $(d-1)^T$ under regular diffusion). More critically, real social networks have cycles, so neighbors of nodes with $m_w = 0$ can still get the message from other nodes in the network. For instance, in simulation on a subset of the Facebook social graph, messages spread with adaptive diffusion reached 81% of network nodes within 20 time steps. Real social networks (and the associated simulation details) are discussed in greater detail in Section IV.

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}. Fig. 4 (right) illustrates the information observed by each spy node, where spies are outlined in red.

ML estimator under adaptive diffusion: A precise ML estimation algorithm is detailed in Algorithm 2. Because adaptive diffusion has deterministic timing, spies only help the estimator discard candidate nodes. We assume the message spreads for an infinite time. Then there is at least one spy on the spine with probability one; consider the first such spy to receive the message, s_0. Notice 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. This spine spy (along with its parent and level metadata)

Algorithm 2: ML Source Estimator for Algorithm 1.

Input: contact network $G = (V, E)$, spy nodes $S = \{s_0, s_1 \ldots\}$ and metadata $s_i : (p_{s_i}, m_{s_i}, u_{s_i})$
Output: ML source estimate $\hat{v}_{\mathrm{ML}}$
Let s_0 denote the lowest-level spine spy, with metadata $(p_{s_0}, m_{s_0}, u_{s_0})$.
$\tilde{V} \leftarrow \{v \in V : \delta_H(v, s_0) \leq m_{s_0}$ and $p_{s_0} \in \mathcal{P}(v, s_0)\}$
$\tilde{E} \leftarrow \{(u, v) : (u, v) \in E$ and $u, v \in \tilde{V}\}$
Define the feasible subgraph as $F(\tilde{V}, \tilde{E})$
$L \leftarrow \emptyset$ {Set of feasible pivots}
$K \leftarrow \emptyset$ {Set of eliminated pivot neighbors}
for all $s \in S$ with $s \in \tilde{V}$ **do**
  Let $\begin{bmatrix} h_{s,\ell_s} \\ h_{\ell_s,s_0} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} |P(s, s_0)| \\ T_{s_0} - T_s \end{bmatrix}$
  $\ell_s \leftarrow v \in \mathcal{P}(s, s_0) : \delta_H(s, \ell_s) = h_{s,\ell_s}$
  $k_s \leftarrow v \in \mathcal{P}(s, s_0) : \delta_H(s, k_s) = h_{s,\ell_s} - 1$
  $L \leftarrow L \cup \{\ell_s\}$ {Add pivot}
  $K \leftarrow K \cup \{k_s\}$ {Add pivot neighbor}
end for
Find the lowest-level pivot: $\ell_{\min} \leftarrow \mathrm{argmin}_{\ell \in L} m_\ell$
$U \leftarrow \emptyset$ {Candidate sources}
for all $v \in \tilde{V}$ where v is a leaf in $F(\tilde{V}, \tilde{E})$ **do**
  if $\mathcal{P}(v, \ell_{\min}) \cap K = \emptyset$ **then**
    $U \leftarrow U \cup \{v\}$
  end if
end for
return $\hat{v}_{\mathrm{ML}}$, drawn uniformly from U

allows the estimator to specify a *feasible subtree* in which the true source must lie. In Fig. 4, 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*.

To identify pivot nodes, 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:

$$h_{s,\ell_s} + h_{\ell_s,s_0} = |\mathcal{P}(s, s_o)|$$
$$h_{\ell_s,s_0} - h_{s,\ell_s} = T_{s_0} - T_s$$

where $\mathcal{P}(s, s_o)$ denotes the path between s and s_o, $h_{s,\ell_s} = \delta_H(s, \ell_s)$ denotes the distance from spy s_i to the pivot node ℓ_s, and h_{ℓ_s,s_0} is equal to $\delta_H(\ell_s, s_o)$ by construction. This system of equations always has a unique solution; hence the uniqueness of ℓ_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_{ℓ_s}, s_0 receives the message at time $T_{\ell_s} + h_{\ell_s,s_0}$, and s receives it at $T_{\ell_s} + h_{s,\ell_s}$.

For instance, in Fig. 4 (right), we can use spies 7 and 8 to learn that node 2 is a pivot with level $m_2 = 2$. After identifying all the pivot nodes, the estimator chooses the minimum-level pivot across all spy nodes, $\ell_{\min}$. In the example, $\ell_{\min} = 1$, since spies 3 and 8 identify node 1 as a pivot with level $m_1 = 1$. The true source must lie in a subtree that is rooted at a neighbor of $\ell_{\min}$, and contains no spies (in our example, this leaves only node 0, the true source).

We now explain why timing information enables the estimator to disregard any subtree neighboring $\ell_{\min}$ that contains at least one spy. Let L denote the set of pivots corresponding to each spy in the feasible subtree; in the example in Fig. 4, $L = \{1, 2\}$. Define $\ell_{\min} = \mathrm{argmin}_{\ell \in L} m_\ell$. That is, $\ell_{\min}$ denotes the pivot closest to the true source in hop distance, i.e., whose level is lowest. Now consider the subtrees 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 $\mathcal{P}(v^*, s)$ 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 ℓ' on the spine such that $|\mathcal{P}(\ell', s)| < |\mathcal{P}(\ell_{\min}, s)|$. 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 Fig. 4, $X = 0$. Thus, the ML estimator chooses one of the leaves in the remaining $X + 1$ candidate subtrees uniformly at random. All remaining nodes in $V \backslash U$ have likelihood 0, where U is the set of all candidate source nodes.

Anonymity properties of adaptive diffusion: Using the previously-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

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

where

$$q_k \equiv \; (1-(1-p)^{((d-1)^k-1)/(d-2)})^{d-1} + (1-p)^{((d-1)^{k+1}-1)/(d-2)}.$$

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

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

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

$$r_k \equiv \; \frac{1}{d-1}\Big((1-(1-p)^{|T_{d,k}|})^{d-1} + (d-1)(1-p)^{|T_{d,k}|} - (d-2)(1-p)^{|T_{d,k}|(d-1)} - 1\Big).$$

The proof is included in Section VI-B1. Briefly, it computes the probability of detection by conditioning on the lowest-level pivot node, $\ell_{\min}$. Given a pivot, the probability of detection depends on the number of subtrees rooted at the neighbors of $\ell_{\min}$ containing no spies. Fig. 3 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 III.1, i.e., $\mathbb{P}(V^* = \hat{v}_{\mathrm{ML}}) = p$. This is in contrast to diffusion, whose probability of detection tends to 1 asymptotically in d. The median Facebook user has 200 friends [32], so these asymptotics have practical implications, as we will see in Section IV.

Expected hop distance asymptotically increasing: We observe empirically that for regular diffusion, $\mathbb{E}[\delta_H(\hat{v}_{\mathrm{ML}}, v^*)]$ approaches 0 as d increases. On the other hand, for adaptive diffusion with a fixed $p > 0$, as $d \to \infty$, $\limsup \mathbb{E}[\delta_H(\hat{v}_{\mathrm{ML}}, v^*)] = 2(1-p)$. 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 $(1-p)$. Fig. 3 includes the line $2(1-p)$ for reference, and we observe that as $d \to \infty$, $\mathbb{E}[\delta_H(\hat{v}_{\mathrm{ML}}, v^*)]$ appears to converge precisely to this line. However, for a fixed d, Theorem 1 implies that as $p \to 0$, $\mathbb{E}[\delta_H(\hat{v}_{\mathrm{ML}}, v^*)] \to \infty$.

IV. Generalizations

Graphs: Here, we consider irregular, cyclic, and finite graphs that arise in real contact networks. Regardless of whether the graph has cycles, the message always propagates over a tree superimposed on the underlying contact network. This is because we do not allow nodes to be 'infected' more than once.[1]

Given that messages always propagate over a tree, the probability of detection over *irregular* trees is tied to performance over general graphs. ML estimation over irregular trees is more straightforward than in [1], primarily because we use the specialized tree protocol that places the source at a leaf node.

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

This ML estimator allows us to evaluate adaptive diffusion over real dataset (social graph connections among 10,000 Facebook users [33]) 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 IV.1, that accounts for cycles in the underlying graph. Fig. 5 lists the probability of detection averaged over 200 trials, for p up to 0.15. 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 Section VI-C, and Fig. 6 illustrates the tradeoff between snapshots and spy nodes.

[1] In practice, we can satisfy this condition without leaking information by asking nodes to send hashed IDs of their received messages over each active neighboring connection. A node that wants to transmit a message (and its associated metadata) only transmits if the message's hash is not included in the recipient's list of previously-received packets. This achieves two goals: recipients will not be infected more than once, and the recipient does not learn which message the sender wanted to transmit. This prevents the recipient from learning metadata for any given message after the first time it receives that message. Notice that this mechanism assumes an honest-but-curious adversarial model (as is the case throughout this work).

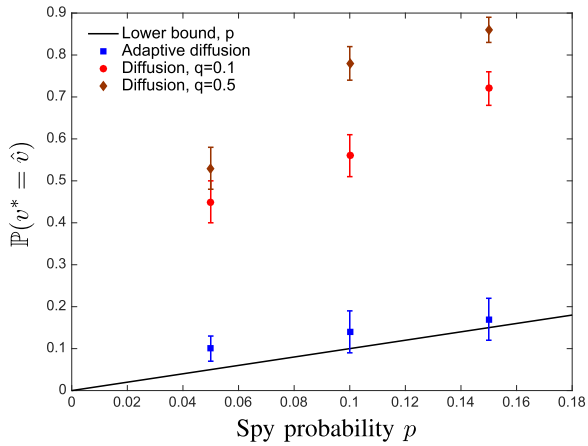

Fig. 5. Probability of detection over the Facebook dataset [33], with standard error.

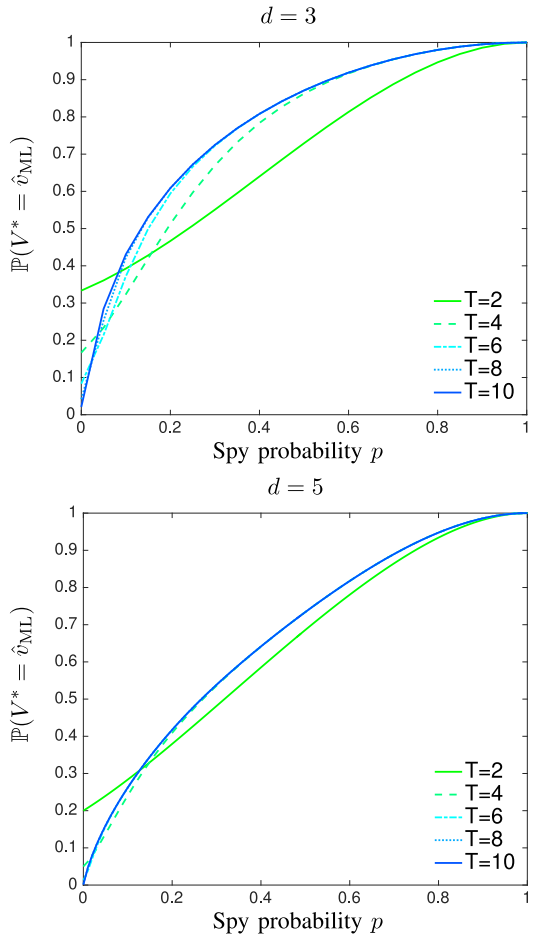

Fig. 6. Probability of detection under the spies+snapshot adversarial model. As estimation time and tree degree increase, the effect of the snapshot on detection probability vanishes.

V. Conclusion

In this paper, we demonstrate that adaptive diffusion has asymptotically optimal anonymity properties over regular trees; we also observe in simulation that in real social networks, adaptive diffusion hides the source of a message against computationally-unbounded adversaries that can observe message metadata at a fraction of corrupted nodes. This is in contrast with regular diffusion, under which message sources are reliably caught by a number of different adversarial models.

We emphasize that these guarantees are statistical rather than cryptographic; this introduces interesting questions for future work. For instance, how does the probability of detection change if an adversary chooses the placement of "spy nodes" adversarially (e.g. by trying to corrupt more popular nodes)? What happens if spy nodes choose not to follow protocol in order to try to boost their probability of detection?

VI. Proofs

A. Line Analysis

1) Proof of Proposition II.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 $t \geq T_1$. 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

$$\mathbb{P}^{(\text{adaptive})}_{T_1|V^*,Q,D}\left(t_1 \middle| v^*, q, \ell\right) = \begin{cases} qB(v^* - \frac{t_1}{2} - 2, \frac{t_1}{2} - 2, q)\mathbb{I}_{(v^* \in [2+\frac{t_1}{2}, t_1])} & \text{if } t_1 \text{ even} \\ B(v^* - \frac{t_1+3}{2}, \frac{t_1-3}{2}, q)\,\mathbb{I}_{(v^* \in [\frac{t_1+3}{2}, t_1])} & \text{if } t_1 \text{ odd} \end{cases} \tag{2}$$

$$\mathbb{P}^{(\text{adaptive})}_{T_1|V^*,Q,D}\left(t_1 \middle| v^*, q, r\right) = \begin{cases} 0 & \text{if } t_1 \text{ even} \\ (1-q)B(\frac{t_1-1}{2} - v^*, \frac{t_1-3}{2}, q)\mathbb{I}_{(v^* \in [1, \frac{t_1-1}{2}])} & \text{if } t_1 \text{ odd.} \end{cases} \tag{3}$$

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:

$$\hat{v}_{\text{ML}}=\begin{cases}\frac{T_1+2}{2}+\left\lfloor q\left(\frac{T_1-2}{2}\right)\right\rfloor & \text{if } T_1 \text{ even } \&\ D=ell,\\ \frac{T_1+3}{2}+\left\lfloor q\left(\frac{T_1-1}{2}\right)\right\rfloor & \text{if } T_1 \text{ odd } \&\ D=ell,\\ 1+\left\lfloor (1-q)\left(\frac{T_1-1}{2}\right)\right\rfloor & \text{if } T_1 \text{ odd } \&\ D=r.\end{cases} \quad (4)$$

Together with the likelihoods in Eqs. (2) and (3), this gives

$$\mathbb{P}^{(\text{adaptive})}_{T_1,D|V^*,Q}\big(t_1,r,\hat{v}_{\text{ML}}=v^*\big|v^*,q\big) = \frac{1}{2}(1-q)\,B\Big(\frac{t_1-1}{2}-v^*,\frac{t_1-3}{2},q\Big)\mathbb{I}_{(\hat{v}_{\text{ML}}=v^*)}\mathbb{I}_{(t_1 \text{ is odd})} \quad (5)$$

$$\mathbb{P}^{(\text{adaptive})}_{T_1,D|Q}\big(t_1,r,V^*=\hat{v}_{\text{ML}}\big|q\big) \frac{1}{2\,n}(1-q)\,B\Big(\frac{t_1-1}{2}-\hat{v}_{\text{ML}},\frac{t_1-3}{2},q\Big)\mathbb{I}_{(t_1 \text{ is odd})} \quad (6)$$

$$\le \frac{(1-q)}{2\,n}\Big(\frac{\sqrt{2}\,\mathbb{I}_{(t_1 \text{ is odd and } t_1>3)}}{\sqrt{\frac{t_1-3}{2}q(1-q)}}+\mathbb{I}_{(t_1=3)}\Big) \quad (7)$$

where $\hat{v}_{\text{ML}}=\hat{v}_{\text{ML}}(t_1,q,r)$ is provided in (4), and the bound on $B(\cdot)$ follows from Gaussian approximation (which gives an upper bound $1/\sqrt{2\pi kq(1-q)}$) and Berry-Esseen theorem (which gives an approximation guarantee of $2\times 0.4748/\sqrt{kq(1-q)}$), for $k=(t_1-3)/2$. Marginalizing out $T_1\in\{3,5,\ldots,2\lfloor(n-1)/2\rfloor+1\}$ and applying an upper bound $\sum_{i=1}^{k}1/\sqrt{i}\le 2\sqrt{k+1}-2\le 2\sqrt{k-1}+\sqrt{1/(2(k-1))}-2\le\sqrt{4(k-1)}$, we get

$$\mathbb{P}\big(D=r,V^*=\hat{v}_{\text{ML}},T_1 \text{ is odd}\big|Q=q\big) \le \frac{(1-q)\sqrt{2}}{2\,n\,\sqrt{q(1-q)}}\sqrt{8\left\lfloor\frac{n-1}{2}\right\rfloor}+\frac{1-q}{2\,n}. \quad (8)$$

Similarly, we can show that

$$\mathbb{P}\big(D=\ell,V^*=\hat{v}_{\text{ML}},T_1 \text{ is odd}\big|Q=q\big) \le \frac{\sqrt{2}}{2\,n\,\sqrt{q(1-q)}}\sqrt{8\left\lfloor\frac{n-1}{2}\right\rfloor}+\frac{1}{n}, \quad (9)$$

$$\mathbb{P}\big(V^*=\hat{v}_{\text{ML}},T_1 \text{ is even}\big|Q=q\big) \le \frac{q\sqrt{2}}{2\,n\,\sqrt{q(1-q)}}\sqrt{8\left\lfloor\frac{n}{2}\right\rfloor}+\frac{1+q}{2\,n}, \quad (10)$$

Summing up,

$$\mathbb{P}(V^*=\hat{v}_{\text{ML}}|Q=q)\le\sqrt{\frac{8}{n\,q\,(1-q)}}+\frac{2}{n}. \quad (11)$$

Recall Q is uniformly drawn from $[0,1]$. Taking expectation over Q gives

$$\mathbb{P}(V^*=\hat{v}_{\text{ML}})\le\pi\sqrt{\frac{8}{n}}+\frac{2}{n}, \quad (12)$$

where we used $\int_0^1 1/\sqrt{x(1-x)}dx=\arcsin(1)-\arcsin(-1)=\pi$.

B. Regular Tree Analysis

1) Proof of Theorem 1: Probability of Detection: We condition on the lowest-level pivot node, $\ell_{\min}$, giving $\mathbb{P}(\hat{v}_{\text{ML}}=v^*)=\sum_{\ell_{\min}}\mathbb{P}(\hat{v}_{\text{ML}}=v^*|\ell_{\min})\mathbb{P}(\ell_{\min})$. Since $\ell_{\min}$ lies on the spine, this is equivalent to conditioning on the distance of $\ell_{\min}$ from the true source.

$$\mathbb{P}(\hat{v}_{\text{ML}}=v^*)=\sum_{k=1}^{\infty}\underbrace{\frac{(1-p)^{(|T_{d,k}|-1)}\,p}{|\partial T_{d,k}|}}_{\ell_{\min}\ (k^{th}\text{ spine node) is a spy}} + \underbrace{(1-p)^{|T_{d,k}|}\mathbb{E}_X\left[\frac{\mathbb{I}(X\neq d-2)}{(X+1)\,|\partial T_{d,k}|}\right]}_{\ell_{\min}\ (k^{th}\text{ spine node) not a spy}} \quad (13)$$

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. Let $w=(1-p)$. We have that

$$\begin{aligned}\mathbb{E}_X\left[\frac{\mathbb{I}(X\neq d-2)}{(X+1)\,|\partial T_{d,k}|}\right]&=\frac{1}{|\partial T_{d,k}|}\left(\mathbb{E}_X\left[\frac{1}{X+1}\right]-\frac{1}{d-1}w^{|T_{d,k}|\cdot(d-2)}\right)\\&=\frac{1}{|\partial T_{d,k}|}\left(\frac{1}{(d-1)w^{|T_{d,k}|}}\left(1-(1-w^{|T_{d,k}|})^{d-1}\right)-\frac{1}{d-1}w^{|T_{d,k}|\cdot(d-2)}\right)\end{aligned}$$

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 $\mathbb{E}[1/(X+1)]=\frac{1}{(\tilde{n}+1)\tilde{p}}(1-(1-\tilde{p})^{\tilde{n}+1})$. Simplifying gives

$$\begin{aligned}P_D&=\sum_{k=1}^{\infty}\frac{1}{(d-1)^k}\Big[(d-1)pw^{|T_{d,k}|-1}+1-w^{|T_{d,k}|\cdot(d-1)}-(1-w^{|T_{d,k}|})^{d-1}\Big]\\&=p+\frac{1}{d-2}+\sum_{k=1}^{\infty}\frac{1}{(d-1)^k}\Big[pw^{|T_{d,k+1}|-1}-w^{|T_{d,k}|\cdot(d-1)}-(1-w^{|T_{d,k}|})^{d-1}\Big]\\&=p+\frac{1}{d-2}-\sum_{k=1}^{\infty}\frac{1}{(d-1)^k}\Big[w^{|T_{d,k+1}|}+(1-w^{|T_{d,k}|})^{d-1}\Big].\end{aligned}$$

where the last line holds because $|T_{d,k+1}|-1=|T_{d,k}|\cdot(d-1)$.

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. The estimator chooses an incorrect candidate subtree with probability $X/(X+1)$.

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

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

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

Simplifying and substituting $\tilde{p} = (1-p)^{|T_{d,k}|}$ 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 $\ell_{\min}$ is not a spy. However, for a fixed p with $d \to \infty$, 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 $\ell_{\min}$ 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 $\ell_{\min}$ 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:

$$a_k \equiv \frac{d-2}{d-1}(1-p)^{|T_{d,k}|(d-1)}$$

and for case (b):

$$b_k \equiv p\frac{d-2}{d-1}(1-p)^{|T_{d,k}|-1}$$

Overall, we get a lower bound of

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

C. Irregular Tree Analysis

1) Proof of Proposition IV.1: All nodes in $V \backslash U$ 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 $\ell_{\min}$; since there is a unique path between u and $\ell_{\min}$ over trees, the only feasible sequence of candidate spine nodes starting at candidate u must traverse $\mathcal{P}(u, \ell_{\min})$. By the Markov property of the adaptive diffusion spreading mechanism, we only need to consider the likelihood of a candidate spine prior to reaching $\ell_{\min}$. 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 $\mathcal{P}(u, \ell_{\min})$. 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.

2) Analysis of spy + snapshot adversarial model: We follow closely the proof of Theorem 1 in Appendix VI-B1. 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 $T \to \infty$. 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, $\ell_{\min}$, giving $\mathbb{P}(\hat{v}_{\mathrm{ML}} = v^*) = \sum_{\ell_{\min}} \mathbb{P}(\hat{v}_{\mathrm{ML}} = v^* | \ell_{\min})\mathbb{P}(\ell_{\min})$. Since $\ell_{\min}$ lies on the spine, this is equivalent to conditioning on the distance of $\ell_{\min}$ from the true source. We first define $|S_{d,T}| = 1 + d((d-$

$$\mathbb{P}(\hat{v}_{\mathrm{ML}} = v^*) = \underbrace{\frac{(1-p)^{|S_{d,T}|-1}}{|\partial S_{d,T}|}}_{\text{no spy}} + \sum_{k=1}^{T/2}\Bigg\{ \underbrace{\frac{(1-p)^{(|T_{d,k}|-1)}\,p}{|\partial T_{d,k}|}}_{\ell_{\min}\ (k^{th}\ \text{spine node}) \text{ is a spy}}$$
$$+ \underbrace{(1-p)^{|T_{d,k}|}(1-(1-p)^{|S_{d,T}|-|T_{d,k+1}|})\mathbb{E}_X\left[\frac{\mathbb{I}(X \neq d-2)}{(X+1)\,|\partial T_{d,k}|}\right]}_{\ell_{\min}\ (k^{th}\ \text{spine node}) \text{ not a spy}}$$
$$+ \underbrace{(1-p)^{|S_{d,T}|-(|T_{d,k+1}|-|T_{d,k}|)}\mathbb{E}_X\left[\frac{\mathbb{I}(X \neq d-2)}{|\partial S_{d,T}| - (d-2-X)|\partial T_{d,k}|}\right]}_{\text{all spy descendants of } k\text{-th spine node}}\Bigg\}, \quad (15)$$

$1)^{T/2}-1)/(d-2)$ as the number of nodes infected at time T, and $|\partial S_{d,T}| = d(d-1)^{(T/2)-1}$ as the number of leaves in the infected subtree. Then, Eq. (15) as shown at the bottom of the previous page. 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.

ACKNOWLEDGMENT

The authors thank Paul Cuff for helpful discussions and for pointing out the Bayesian interpretation of the Pólya's urn process.

REFERENCES

[1] G. Fanti, P. Kairouz, S. Oh, and P. Viswanath, "Spy versus spy: Rumor source obfuscation," in *Proc. ACM SIGMETRICS Perform. Eval. Rev.*, 2015, pp. 271–284.
[2] B. Johnson, "Privacy no longer a social norm, says facebook founder," *The Guardian*, 2010.
[3] "Inside the mind of Google," *CNBC*, 2010.
[4] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proc. IEEE Symp. Security Privacy*, 2009, pp. 173–187.
[5] B. Greschbach, G. Kreitz, and S. Buchegger, "The devil is in the metadata? New privacy challenges in decentralised online social networks," in *Proc. 4th Int. Conf. Pervasive Comput. Commun. Workshops*, 2012, pp. 333–339.
[6] Whisper (2014). [Online]. Available: http://whisper.sh
[7] Yik yak (2014). [Online]. Available: http://www.yikyakapp.com/
[8] Team blind (2014). [Online]. Available; http://us.teamblind.com/
[9] Secret (2014). [Online]. Available: https://www.secret.ly
[10] L. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 94–101, Dec. 2009.
[11] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5163–5181, Aug. 2011.
[12] P. C. Pinto, P. Thiran, and M. Vetterli, "Locating the source of diffusion in large-scale networks," *Phys. Rev. Lett.*, vol. 109, no. 6, 2012, Art. no. 068702.
[13] C. Sterbenz, "Cops are creating totally bogus Facebook profiles just so they can arrest people," *Business Insider*, 2013.
[14] K. Zhu, Z. Chen, and L. Ying, "Locating contagion sources in networks with partial timestamps," *Data Mining and Knowledge Discovery*, pp. 1–32, 2015.
[15] Z. Wang, W. Dong, W. Zhang, and C. Tan, "Rumor source detection with multiple observations: Fundamental limits and algorithms," in *Proc. ACM Int. Conf. Meas. Model. Comput. Syst.*, 2014, pp. 1–13.
[16] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 11–20.
[17] A. Arbore, V. Fioriti, and M. Chinnici, "The topological defense in SIS epidemic models," *Chaos, Solitons & Fractals*, vol. 86, pp. 16–22, 2016.
[18] W. Luo, W. Tay, and M. Leng, "How to identify an infection source with limited observations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 586–597, Apr./Jun. 2013.
[19] K. Zhu and L. Ying, "A robust information source estimator with sparse observations," *Computational Social Networks*, vol. 1, no. 1, pp. 1–21, 2014.
[20] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Network forensics: Random infection versus spreading epidemic," in *Proc. 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Meas. Model. Comput. Syst.*, 2012, pp. 223–234.
[21] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Detecting epidemics using highly noisy data," in *Proc. 14th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2013, pp. 177–186.
[22] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "On identifying the causative network of an epidemic," in *Proc. Proc. 50th Annu. Allerton Conf. Commun. Control Comput.*, 2012, pp. 909–914.
[23] N. Borisov, G. Danezis, and I. Goldberg, "DP5: A private presence service," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 4–24, 2015.
[24] M. Jawad, P. Serrano-Alvarado, and P. Valduriez, "Protecting data privacy in structured P2P networks," in *Proc. 2nd Int. Conf. Data Manag. Grid Peer-to-Peer Syst.*, 2009, pp. 85–98.
[25] P. Winter and S. Lindskog, "How the great firewall of China is blocking Tor," *FOCI*, 2012.
[26] R. Singel, "Point, click ... eavesdrop: How the FBI wiretap net operates," *Wired*, 2007.
[27] P. Shadbolt, "FireChat in Hong Kong: How an app tapped its way into the protests," *CNN*, 2014.
[28] H. Corrigan-Gibbs, "Riposte: An anonymous messaging system handling millions of users," *Proc. IEEE Symp. Security Privacy*, 2014, pp. 321–338.
[29] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *J. Cryptol.*, vol. 1, no. 1, pp. 65–75, Jan. 1988.
[30] H. C-Gibbs and B. Ford, "Dissent: Accountable anonymous group messaging," in *Proc. 17th ACM Conf. Comput. Commun. Security*, 2010, pp. 340–350.
[31] N. Johnson and S. Kotz, *Urn Models and Their Application*. Hoboken, NJ, USA: Wiley, 1977.
[32] A. Smith, "6 new facts about Facebook," *Pew Research*, 2014.
[33] B. Viswanath, A. Mislove, M. Cha, and K. Gummadi, "On the evolution of user interaction in Facebook," in *Proc. 2nd ACM Workshop Online Soc. Netw.*, Aug. 2009, pp. 37–42.

Giulia Fanti received the B.S. degree in ECE from Olin College of Engineering, Needham, MA, USA, in 2010, and the M.S. and Ph.D. degrees in EECS from the University of California, Berkeley, CA, USA. She is currently a Postdoctoral Scholar at the University of Illinois at Urbana-Champaign, Champaign, IL, USA, studying privacy-preserving technologies. She received the National Science Foundation Graduate Research Fellowship, as well as a Best Paper Award at ACM Sigmetrics 2015 for her work on anonymous rumor spreading.

Peter Kairouz received the B.E. degree in ECE from the American University of Beirut, Beirut, Lebanon, and the M.S. degree in ECE from the University of Illinois at Urbana Champaign (UIUC), Champaign, IL, USA, in 2010 and 2012, respectively. He is currently working toward the Ph.D. degree at UIUC. He was an R&D Intern at Qualcomm Research in 2012 and 2013, and a Research Intern at Google in 2015. He received numerous scholarships and awards including the 2016 Harold L. Olesen Award for Excellence in Undergraduate Teaching, the SIGMETRICS 2015 Best Paper Award, the 2012 Roberto Padovani Scholarship from Qualcomm's Research Center in 2012, and the 2012 Benjamin Franklin Scholarship. His research interests include statistical data privacy and security, machine learning, and big data.

Sewoong Oh received the Ph.D. degree from the Department of Electrical Engineering, Stanford University, Stanford, CA, USA, in 2011. He is currently an Assistant Professor of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana Champaign, Champaign, IL, USA. He was a Postdoctoral Researcher at the Laboratory for Information and Decision Systems, MIT, Cambridge, MA, USA. His research interests include statistical inference and privacy.

Kannan Ramchandran (F'05) received the Ph.D. degree in EE from Columbia University, New York, NY, USA, in 1993. He is currently a Professor of electrical engineering and computer science at University of California, Berkeley, CA, USA, where he has been since 1999. He was a Faculty at University of Illinois at Urbana Champaign (UIUC), Champaign, IL, USA, from 1993 to 1999, and with AT&T Bell Labs from 1984 to 1990. He has published extensively in his field, holds more than a dozen patents, and has received several awards for his research and teaching including the IEEE Information Theory Society and Communication Society Joint Best Paper Award for 2012, the IEEE Communication Society Data Storage Best Paper Award in 2010, two Best Paper Awards from the IEEE Signal Processing Society in 1993 and 1999, the Okawa Foundation Prize for Outstanding Research at Berkeley in 2001, and the Outstanding Teaching Award at Berkeley in 2009, and the Hank Magnuski Scholar Award at Illinois, in 1998.

Pramod Viswanath received the Ph.D. degree in EECS from the University of California, Berkeley, CA, USA, in 2000. He was a Member of Technical Staff at Flarion Technologies until August 2001 before joining the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA. Dr. Viswanath received the Xerox Award for Faculty Research from the College of Engineering, UIUC, in 2010, the Eliahu Jury Award from the Electrical Engineering and Computer Science Department, University of California at Berkeley, in 2000, the Bernard Friedman Award from the Mathematics Department, University of California at Berkeley, in 2000, and the National Science Foundation CAREER Award, in 2003. He was an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY from 2006 to 2008.