Quantifying Unlinkability in Multi-hop Wireless Networks

Victoria Manfredi
vumanfredi@wesleyan.edu
Wesleyan University
Middletown, Connecticut, USA

Cameron Donnay Hill
cdhill@wesleyan.edu
Wesleyan University
Middletown, Connecticut, USA

ABSTRACT

Consider a multi-hop wireless network in which devices act as anonymizing routers. Even if devices anonymize their link transmissions, an adversary may still be able to infer key information by observing the traffic patterns in the network. In this work, we quantify what impacts how well a Kalman-filter based adversary can infer unlinkability, that is, the probability that different pairs of devices are communicating, from anonymized link transmissions. We assume that devices do not reorder packets to mix traffic and thereby increase unlinkability. Instead, we show that traffic mixing is still possible due to the use of multi-hop routing and broadcast transmissions, with the amount of mixing dependent on the network characteristics. In simulation, we find that i) for unicast links, as network connectivity increases, unlinkability decreases, while for broadcast links as connectivity increases unlinkability increases, ii) link dynamics increase unlinkability in poorly connected topologies, iii) well-connected topologies achieve the same level of unlinkability with fewer transmissions per packet delivered, and (iv) a lattice topology has consistently good unlinkability in different scenarios.

CCS CONCEPTS

• **Networks** → **Mobile ad hoc networks**; **Network privacy and anonymity**; **Network protocol design**; **Network simulations**.

KEYWORDS

wireless networks, multi-hop routing, anonymous communication, unlinkability

ACM Reference Format:
Victoria Manfredi and Cameron Donnay Hill. 2020. Quantifying Unlinkability in Multi-hop Wireless Networks. In *23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '20), November 16–20, 2020, Alicante, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3416010.3423216

1 INTRODUCTION

Rather than relying on fixed infrastructure like Internet routers or cell towers to relay traffic, in a multi-hop wireless network devices relay traffic for each other in a peer-to-peer fashion. Lack of infrastructure not only makes multi-hop wireless networks easier to deploy, it also increases privacy. For instance, devices can avoid communication over infrastructure that may be monitored [28, 38], and users can better control the distribution of their data by ensuring that any collected data is stored locally.

Consider then a multi-hop wireless network in which devices act as anonymizing routers. Even if devices anonymize their link transmissions an adversary may still be able to infer important information by observing the traffic patterns in the network, such as which pairs of devices are communicating. This is problematic since in many multi-hop wireless networks, different devices have different roles (e.g., sources vs. sinks in a sensor network) and some devices are more critical to network functionality (e.g., a military commander) than others. If an adversary can identify such devices it can prevent important information from reaching its destination.

Given this network scenario, our goal is to quantify what impacts how well an adversary can infer *unlinkability* [29], that is, the probability that different pairs of devices are communicating (see Sec. 2.1), given the anonymized link transmissions. We assume that the devices in the multi-hop wireless networks we consider do not mix (i.e., reorder) traffic, unlike a mix network [4]. Instead, we hypothesize that traffic mixing is still possible due to the use of multi-hop routing and broadcast transmissions (see Fig. 1 and Sec. 2.2). The amount of traffic mixing that is possible should depend on the flows present, the network connectivity, the link dynamics, and the routing strategy. It is these network characteristics whose influence on traffic mixing and thus unlinkability that we investigate in this work.

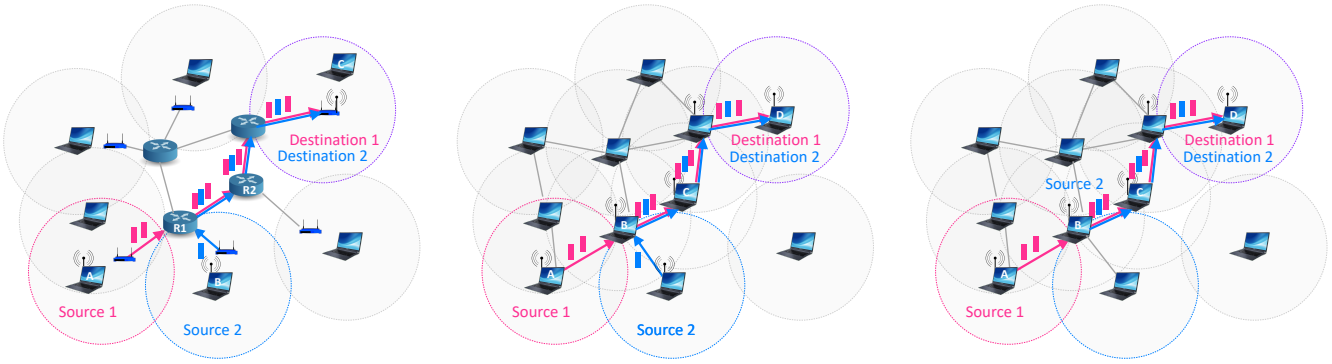
To quantify unlikability, we assume a global adversary that passively eavesdrops on the anonymized packet transmissions on each link. The adversary uses these transmissions to compute a probability distribution over the possible communicating pairs of devices. We formulate the adversary as a Kalman filter to compute this distribution and derive an unlinkability metric. We then introduce the idea of *anonymization efficiency* to quantify the efficiency of unlinkable communication in different network scenarios.

In simulation, we confirm that traffic mixing does occur even when devices themselves do not mix traffic. We show that i) for unicast links, as network connectivity increases, unlinkability decreases, while for broadcast links as connectivity increases unlinkability increases, ii) link dynamics increase unlinkability in poorly connected topologies, iii) well-connected topologies achieve the same level of unlinkability with fewer transmissions per packet delivered, and (iv) a lattice topology has consistently good unlinkability in different scenarios.

The rest of this paper is structured as follows. In Sec. 2, we explain how traffic mixing can happen in multi-hop wireless networks. In Sec. 3 we review related work. In Sec. 4, we describe our Kalman filter adversary. In Sec. 5, we show how we use our Kalman filter adversary to derive an unlinkability metric and propose the idea

MSWiM '20, November 16–20, 2020, Alicante, Spain

ACM ISBN 978-1-4503-8117-8/20/11…$15.00
https://doi.org/10.1145/3416010.3423216



(a) **Internet routing and flow interleaving.** On the Internet, devices are only either end-hosts or routers, and only end-hosts are sources or destinations of traffic. Flows from different end-hosts may cross at a router, but incoming and outgoing traffic at the router match. E.g., traffic from Source 1 and Source 2 cross at Router R1, but the incoming and outgoing traffic at R1 is the same.

(b) **Multi-hop routing and flow interleaving.** In a multi-hop wireless network, devices are both end-hosts and routers, and so any device may be the source or destination of traffic as well as a router. Like on the Internet, flows from different devices may cross at a device operating as a router. E.g., traffic from Source 1 and Source 2 cross at Device B.

(c) **Multi-hop routing and packet interleaving.** Only with multi-hop routing, will packet interleaving happen at a device. For instance, Device B is both a source of packets (for Source 2) and forwarder of packets (for Source 1). Consequently, the incoming packets at Device B are different than the outgoing packets, due to interleaving of Source 1 and Source 2 packets.

Figure 1: Illustration of how multi-hop routing supports packet mixing. A flow is a set of packets sent from a source to a destination over a sequence of links (i.e., path). Two-way communication requires two flows, one in each direction.

of anonymization efficiency. In Sec. 6, we evaluate our unlinkability metric in simulation. Finally, in Sec. 7, we summarize our contributions.

2 BACKGROUND

2.1 Computing Unlinkability

In this work, we focus on multi-hop wireless networks in which devices act as anonymizing routers. To anonymize transmissions, devices re-encrypt [13] packets at the network layer, and set link layer addresses in such a way as to hide the intended next hop of a packet yet still allow this hop to process the packet. We assume devices do not mix traffic, but, as we shall see in Sec. 2.2 and quantify in this paper, traffic mixing can still happen.

In the anonymity literature, the adversary's goal is to compute the *unlinkability* of a packet's source with its destination [29]. Unlinkability is also known as relationship or source-destination anonymity. To enable unlinkable communication, Chaum [4] proposed mix nodes that re-order and re-encrypt the messages passing through them to hide the message paths, and the idea of onion routing used in Tor [11], where messages are encrypted multiple times, each layer of encryption corresponding to the next hop to which the message is to be forwarded. In mix networks, mixing of messages at nodes is done to decorrelate input traffic from output traffic. When mixing is not done (e.g., as in Tor to reduce user latency), timing attacks can potentially be used [19, 42] to accurately correlate a message's source with its destination.

In the network tomography literature, the problem of traffic matrix inference [22, 35, 40] is similar to that of unlinkability but does not consider explicit obfuscation of traffic patterns. Additionally, such inference usually considers aggregated traffic, and assumes it is possible to periodically obtain the true traffic matrix at some cost, which is useful for training an inference algorithm.

In this work, we assume a global adversary uses the packet transmissions it passively observes over links to compute a probability distribution, i.e., the *flow distribution*, over the possible flows, see Fig. 1. Because this adversary cannot parse any packet header or payload data it does not know which flows are present. Assuming a passive adversary actually makes our problem harder, not easier, since our goal is to be the adversary and compute unlinkability, rather than to design mechanisms to increase unlinkability.

2.2 What impacts traffic mixing?

We assume that the devices in the multi-hop wireless networks we consider do not mix traffic. For instance, if traffic is rare or high delays are problematic, it may be infeasible for devices to wait for sufficient packets in their queues so that the packets can be reordered. Instead, we conjecture that traffic mixing is still possible due to the network features below. Our focus in this work is specifically on the impact of multi-hop routing, broadcast transmissions, network connectivity, and link dynamics on traffic mixing.

2.2.1 Multi-hop routing. Due to multi-hop routing, every device may be the source or destination of a flow, and hence packet, even though that device may also forward packets on flows to or from other devices. Thus, not every packet entering a device will leave it, and every packet leaving a device may or may not have been sourced by the device. We call this packet interleaving, see Fig. 1(c). Because of packet interleaving, the adversary must consider all possible devices along a path as possible sources and destinations of traffic. Flow interleaving, when two flows cross at a device, see Fig. 1(b), increases packet interleaving.

2.2.2 Wireless links. The MAC protocol used to access a wireless link typically has a component introducing random delays. Because wireless transmissions interfere, a device may attempt (typically up to 7) re-transmissions of the same packet at random backoff times,

interleaved with transmissions from other devices, to cope with collisions. Thus, the dwelling time of packets at devices is variable.

2.2.3 Broadcast transmissions. Wireless transmissions may be unicast (unidirectional) or broadcast (omnidirectional). When a wireless device transmits a packet using a broadcast transmission, all devices within range receive the transmission, not just the intended recipient. A receiving device then determines whether it is the intended recipient by checking the packet destination address. Thus, to an outside observer, which neighbor device is the intended recipient may be unclear, assuming no control traffic such as acknowledgements are sent upon receipt.

2.2.4 Network connectivity. Well-connected topologies should support higher traffic mixing and thus unlinkability. In our simulations in Sec. 6, we measure network connectivity using *algebraic connectivity*, λ_2, which is defined as the second-smallest eigenvalue of the normalized Laplacian matrix of a graph [6]. The larger the value of λ_2, the more well-connected is the graph.

2.2.5 Link dynamics. Due to wireless interference, fading, or mobility the network connectivity may change, and consequently, the pattern of wireless transmissions observed by an adversary may change, even if the underlying set of flows stays the same.

2.2.6 Multiple packet copies. Flow correlation attacks typically assume a single copy of a packet. To cope with link dynamics, a multi-hop routing strategy may transmit multiple copies of a packet.

3 RELATED WORK

Existing unlinkability metrics [12, 15, 18, 25, 26, 34] are not suitable for our work, as they do not give a straightforward way to compute unlinkability for arbitrary network scenarios or consider multi-hop routing or link dynamics. Other works have designed protocols for unlinkable [1, 2, 14, 31] and anonymous [3, 8, 17, 26, 33] communication for multi-hop wireless networks, but do not give us a way to compute unlinkability. This motivates our derivation of a new metric in Sections 4 and 5 based on a Kalman filter adversary.

Works [20, 30] on inferring unlinkability for anonymous wireless and mobile ad hoc networks correlate and aggregate link layer frames into traffic matrices over time. In comparison, we focus on performing inference at the network layer and build a statistical model that explicitly incorporates adversary knowledge and lets us quantify unlinkability in many different network scenarios.

Work [37] similar in spirit to ours but in mix networks builds a probabilistic model to infer unlinkability using user selected mix path lengths and mixing strategies to compute the model probabilities. Due to computational constraints they focus on smaller static networks and consider up to 10 mixes. In comparison, our Kalman filter model allows us to more directly incorporate different multi-hop routing strategies, as well as consider the impact of different network characteristics, including link dynamics. While we are also affected by compuational constraints, we look at networks with up to 25 devices.

In our work, we use a biased random walk routing strategy to limit control overhead and handle topology changes, see Sec. 4.1.4. This strategy lets us further quantify the benefits of anonymous broadcast [36] on unlinkability, and the additional impact of multi-hop routing. Other works [8, 23, 24] on anonymous communication have also considered a random walk routing strategy, but here our focus is not to design a new anonymous routing strategy, but to understand how routing randomness impacts unlinkability.

In comparison to works on traffic matrix inference [22, 35, 40], not only do we consider traffic obfuscation and multi-hop routing, we also focus on inferring individual flows in multi-hop wireless networks with potentially dynamic topologies. While our model has similarities with the Kalman filter based approach of [35], those authors operate under the assumption that their model can be initialized using the true traffic matrix and instead their goal is to track how traffic in this traffic matrix changes over time.

Works examining the impact of network topology in the context of anonymity primarily focus on identifying which mix topologies are more vulnerable to attack or enable faster mixing [7, 9, 11, 23, 27] or the interplay of mix connectivity with dummy packets for padding [9]. Of particular interest to us are works identifying well-connected topologies like expander graphs [7, 27], and scale-free and small-world topologies [27] as being mix topologies that support efficient mixing in terms of message path lengths. Here, however, our focus is to understand not just the impact of connectivity on unlinkability but also the impact of other network characteristics.

Recent mix network implementations [38] ensure unlinkability even when both the entry and exit nodes are controlled by an adversary, unlike Tor [11]. Other work on mix networks [39] looks at adding noise to protect against traffic analysis. Mix networks as well as the Tor onion routing overlay are typically constructed using end-hosts, which can be both sources and destinations of traffic as well as traffic relays, but these implementations still rely on the Internet to route traffic between relays. When Tor onion routing is implemented at the network layer [5] and mixes are instead high-speed routers, the mix network comprises only routers. Our model could thus be viewed as a multi-hop wireless network in which every device is both a wireless Tor node operating at the network layer and a possible source or destination of traffic. In our scenarios, though, not only can the number of relays be much larger than the three used in Tor, the next relay to use can change depending on network dynamics and multi-hop routing, while broadcast wireless transmissions further protect against traffic analysis.

4 KALMAN FILTERS FOR FLOW INFERENCE

We now overview how we use a Kalman filter [16, 41] to obtain the flow distribution. Computing the flow distribution is generally a computationally intensive task. The primary reason why we use a Kalman filter to model our states and observations with continuous rather than discrete random variables (like in a hidden Markov model) is to make our computations more efficient. Our goal, however, is not to propose Kalman filters as a real-time adversary for flow inference, but instead make meaningful comparisons of the unlinkability of different network scenarios.

4.1 Kalman Filter

Kalman filters originated in the target tracking literature and assume the true location (state) of a tracked object is unobservable

(hidden) and modeleded as a Gaussian random variable. Noisy observations of the true state are assumed to be available and are also modeled as a Gaussian random variable. The Kalman filter update equations are thus given as follows.

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \quad (1)$$

$$\mathbf{y}_t = \mathbf{B}\mathbf{x}_t + \mathbf{v}_t \quad (2)$$

Let the initial state be $\mathbf{x}_0 \sim \mathcal{N}[\mu, \Sigma]$. Then $\mathbf{x}_{t+1}$ is a linear function of the state $\mathbf{x}_t$ plus some Gaussian noise $\mathbf{w}_t \sim \mathcal{N}[\mathbf{0}, \mathbf{Q}]$. The observations $\mathbf{y}_t$ are similarly a linear function of the state $\mathbf{x}_t$ plus some Gaussian noise $\mathbf{v}_t \sim \mathcal{N}[\mathbf{0}, \mathbf{R}]$. The transition matrix $\mathbf{A}$ transforms the current state $\mathbf{x}_t$ to the next state $\mathbf{x}_{t+1}$. The observation matrix $\mathbf{B}$ transforms the current state $\mathbf{x}_t$ to the current observation, $\mathbf{y}_t$. When the assumptions of linearity and Gaussian noise are true, the Kalman filter is an optimal estimator of the state.

In the rest of this section we describe how we set-up a Kalman filter to solve the flow inference problem.

4.1.1 States $\mathbf{x}_0$, $\mathbf{x}_t$ and Covariances Σ, $\mathbf{Q}$. We model a multi-hop wireless network as a graph, $G = (V, E)$, where $N = |V|$ is the number of devices and E is the set of links. In a network with N devices, there are at most N^2 possible flows including those whose source and destination are the same device. Since which flows are present is unknown, we model all possible flows. We include the possibility of self-flows as this gives more flexibility to the model estimation: for instance, self-flows could correspond to cover traffic or to devices holding onto packets for extended periods of time.

We define the state $\mathbf{x}_t$ to be a vector of length $2N^2$. The first N^2 states represent the total traffic on each of the N^2 possible flows up to time t. The next N^2 states represent the traffic arrivals on each flow. While it would be natural to have the state additionally model the total traffic at each device for each flow at each timestep, we do not do this since it makes inference intractable as the state space size increases to $2N^3$ from $2N^2$.

We set each entry of the initial state vector, μ, to $1/2N^2$. We set the $2N^2 \times 2N^2$ initial covariance matrix, Σ, to the identity matrix times 0.1. We set the $2N^2 \times 2N^2$ covariance matrix, $\mathbf{Q}$, to the identity matrix.

4.1.2 Observations $\mathbf{y}_t$ and Covariance $\mathbf{R}$. In a network with N devices, there are at most N^2 possible links including self-links. While we assume which links are present in the network is known, which links exist or have traffic on them may change over time, and so we must model all links. We include self-links as these could model cover transmissions or delaying of transmissions.

We define the observations $\mathbf{y}_t$ to be a vector of length N^2 representing for each link, the total traffic transmitted up to time t. We consider both unicast and broadcast wireless links. For broadcast links, we assume all unicast links incident to a device are activated during packet transmission, and so all dimensions of $\mathbf{y}_t$ corresponding to those links will have traffic on them. We set the $N^2 \times N^2$ observation covariance matrix, $\mathbf{R}$, to the identity matrix.

4.1.3 Transition Matrix, $\mathbf{A}$. The transition matrix $\mathbf{A}$ is of size $2N^2 \times 2N^2$ and maps states from one timestep to the next. We assume traffic from one flow never switches to another flow, and that traffic currently on a flow accumulates over time. We set the entries of $\mathbf{A}$ as follow, where $src(i)$ indicates the source device for flow i and $dst(i)$ indicates the destination device for flow i.

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } j = N^2 + i \text{ and } src(i) \neq dst(i) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The first N^2 elements in $\mathbf{x}_t$ keep track of the total traffic on each flow, while the next N^2 elements in $\mathbf{x}_t$ keep track of the new traffic arrivals on each flow. Intuitively, when $\mathbf{x}_t$ is multiplied with $\mathbf{A}$, the result is the following. The total traffic on each flow in $\mathbf{x}_t$ is multiplied with the diagonal elements of $\mathbf{A}$ while the new traffic on each flow is multiplied with the elements above the main diagonal of $\mathbf{A}$. The results are then summed together, giving the new total traffic on each flow.

4.1.4 Observation Matrix, $\mathbf{B}$. The matrix $\mathbf{B}$ is of size $N^2 \times 2N^2$, where rows are the total traffic sent over each link and columns are the total traffic and arrivals in each flow. We set the entries of $\mathbf{B}$ directly given assumptions about (1) the multi-hop routing strategy in use and (2) the network connectivity.

In our simulations in Sec. 6, we use a φ-randomized routing strategy that forwards packets to the next device on the shortest path (or stays at the current device) with probability φ and forwards packets to a random neighbor device with probability $1 - \varphi$. A packet's path terminates once it reaches its destination. Setting $\varphi = 1$ gives shortest path routing and lets us quantify how much unlinkability exists even when devices do not themselves mix traffic. Setting $\varphi = 0$ gives a random walk and lets us quantify the unlinkability gained due to randomness in the routing strategy. Essentially the time to deliver a packet is the first passage time from the source to destination for a random walk parameterized by φ. If the packet reaches its destination before mixing has occurred, then unlinkability will not be maximized. If the packet reaches its destination after mixing has occurred, then unlinkability will be maximized but possibly using more transmissions than necessary.

When setting the entries of $\mathbf{B}$, however, the adversary assumes only shortest path routing is used and has no knowledge of φ. The adversary does, however, know the true network topology. Let $src(j)$ be the source device of flow j and let $dst(j)$ be the destination device. Let $snd(i)$ be the sending device on link i and let $rcv(i)$ be the receiving device. $Nbr(k)$ indicates the set of neighbor devices for device k and $R\{src : dst\}$ indicates the set of devices comprising the shortest route between a source device, src, and destination device, dst. We then set the entries of $\mathbf{B}$ as follows.

$$\mathbf{B}_{ij} = \begin{cases} 1, & \text{if } snd(i) \neq dst(j),\ rcv(i) \in Nbr(snd(i)), \\ & \text{and } rcv(i) \in R\{src(j) : dst(j)\} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Intuitively, for each possible sending device $snd(i)$, $\mathbf{B}$ gives the next hop receiving device $rcv(i)$ for packets on flow j.

4.2 Flow Inference

The Kalman filter lets us compute the maximum likelihood estimates of the flow state given the observed link transmissions, assuming the linear Gaussian assumptions hold.

Given a Kalman filter with its parameters set for a network scenario and a sequence of observations $\mathbf{y}_{1:T}$, we can recursively

compute the probability distribution $P(\mathbf{x}_t|\mathbf{y}_{1:t})$. This distribution remains Gaussian and the computation remains tractable even with many observations. Let $\bar{\mu}_F$ be the mean of this distribution and let μ_F be the vector containing the first N^2 values of $\bar{\mu}_F$, corresponding to the estimates of the total traffic on the N^2 possible flows. We use μ_F to derive a probability distribution over flows, P_F.

We perform one post-processing step: any entries in μ_F that are negative are set to zero, since negative amounts of traffic on a flow are not feasible. We conjecture several reasons for the presence of negative entries. First, from inspecting the values of μ_F, negative entries seem to arise in part for flows that don't exist but that are sub-flows of flows that do exist, and so may capture the removal of traffic at one device and the transfer to another device. Second, our problem formulation is unlikely to strictly satisfy the linear, Gaussian assumptions of the Kalman filter and so negative entries may be a consequence of numerical approximations.

We compute the flow distribution P_F as follows, where $\mu_F(i)$ is the total traffic on flow i and $P_F(i)$ is the probability that the ith flow had traffic.

$$P_F(i) = \frac{\mu_F(i)}{\sum_{j=1}^{N^2} \mu_F(j)} \quad (5)$$

As the network size increases, the probability mass is more finely dispersed over the possible states. We thus renormalize P_F focusing on the most likely states. To identify these states, we sort the probabilities and find the value, min_F, at index $2F$ in the sorted list, where F is the actual number of flows in the network. We set all probabilities less than min_F to zero and renormalize P_F. Knowing F is strictly not necessary and any cutoff point could be used.

5 QUANTIFYING UNLINKABILITY

Regardless of the adversary model, computing unlinkability for a given network scenario is computationally hard, given the large space of possibilities and limited adversary information. Consequently, some kind of probabilistic model is necessary. Here, we describe a new metric based on our Kalman filter adversary.

5.1 Unlinkability Metric

We derive an unlinkability metric, U, by computing the total variation distance between the flow distribution P_F and the true distribution, P_T. Total variation distance has range $[0, 1]$ and so U also has range $[0, 1]$. We use total variation distance rather than the Kullback-Liebler (KL) divergence, because the KL-divergence is not a true metric (e.g., the distance from P_F to P_T could be different than the distance from P_T to P_F), and our goal is a metric we can use to compare unlinkability in many different network scenarios.

$$U = \frac{1}{2}\sum_{i=1}^{N^2}|P_F(i) - P_T(i)| \quad (6)$$

We obtain a bound on the maximum unlinkability, U_{max}, by computing U when P_F is set to the uniform random distribution, but excluding those flows for which the source and destination are the same device. While we considered self-flows in the Kalman filter computation, we do not consider them here since we are interested only in how many "real" flows are correctly inferred. U_{max} can be viewed as a bound on the worst performance of an intelligent adversary, but not the absolute maximum unlinkability achievable, which would be achieved when all probability weight is put on flows that are not present. In our experiments, the U_{max} values are typically in the range of 0.9 to 1. Because U_{max} can be less than one, for clarity, we show the normalized unlinkability in our results computed as follows.

$$U_{norm} = \frac{U}{U_{max}} \quad (7)$$

If $U_{norm} > 1$, this indicates that the adversary's flow inference is worse than uniformly random guessing.

Our use of normalization here is to provide a bound on the performance of our adversary and give additional insight when comparing the adversary's performance in different network scenarios. In practice, normalizing the unlinkability by the performance of a uniform random adversary may not always be useful, and a more intelligent adversary could be used. For instance, if there are few flows in the network, then the adversary's random guessing could be restricted to consider only those nodes that forward any traffic for any flow.

5.2 Anonymization Efficiency

A network's characteristics impact both unlinkability and the total link transmissions used to deliver traffic. For instance, while additional transmissions from using a longer path to route traffic from source to destination increases unlinkability, it requires using network capacity above what is minimally required to deliver the traffic over the shortest path. Alternatively, while having every device retransmit every packet over a broadcast link would maximize unlinkability, it would also be inefficient.

We would thus like to quantify the gains in unlinkability at the cost of transmissions. To do this, we introduce a metric we call *anonymization efficiency*, E, computed as follows, where D_{tx} is the total packets transmitted and D_{dv} is the total packets delivered.

$$E = \frac{U_{norm}}{D_{tx}/D_{dv}} \quad (8)$$

When computing E in our simulations, we assume infinite capacities on links and infinite queues at devices. We focus solely on data traffic since the amount of control traffic generated by a routing strategy is strategy specific and a variable and hard to optimize constraint. Instead, we assume that there is no control traffic present, neither to set up routes nor any acknowledgements that might be sent in response to received data packets. It is true that knowledge of control traffic should increase an adversary's ability to accurately estimate the flow distribution and consequently decrease unlinkability. However, our interest in this work is not purely the absolute value of unlinkability or anonymization efficiency, but how they change in different network scenarios.

6 EVALUATION

Our simulations are done in R and run using the MIT SuperCloud and Lincoln Laboratory Supercomputing Center [32]. We use the FKF (Fast Kalman Filter) package [21] as our Kalman filter implementation. We next describe our simulation set-up and then overview our simulation results.

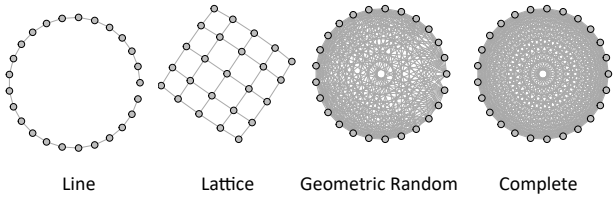


Figure 2: Network topologies used in simulations.

6.1 Simulation Setup

6.1.1 Network topology. We assume the adversary knows the network topology and whether unicast or broadcast links are present. As shown in Fig. 2, we consider four network topologies, with $N = 25$: (1) line, (2) 4-degree lattice, (3) geometric random graph, and (4) complete graph. To generate a geometric random graph, points are randomly placed in a unit square. Then any points within a given transmission radius are connected; we use relatively large radii of 0.6 and 0.85 to ensure a connected graph.

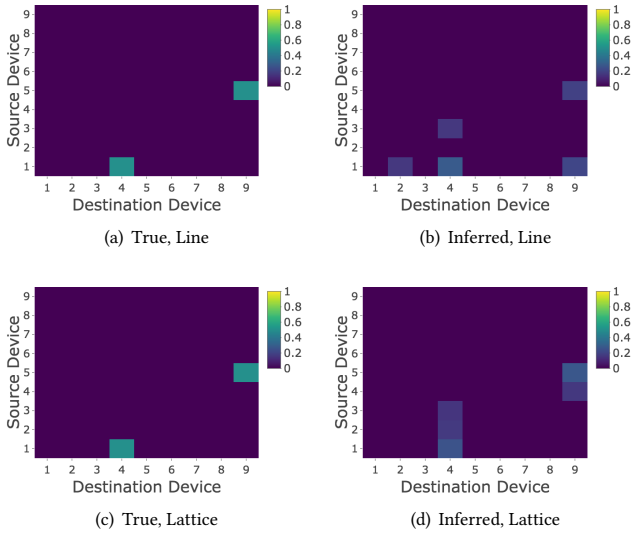
6.1.2 Link dynamics. We consider scenarios where devices are stationary but the links present may change. We use a 2-state Markov model for the link dynamics: links are i.i.d. and stay up from one timestep to the next with probability p and stay down with probability q. We initialize the up or down state of each possible link according to the steady-state distribution of the Markov model, $\pi = (1-q)/(2-p-q)$. On each timestep, we then update the state of each link according to the model.

For both unicast and broadcast links, if a link that was present disappears, no packets can be sent over that link. Since we model a broadcast link as comprising a set of unicast links, we assume the adversary can tell when any of the individual unicast links disappears. We assume, however, that the adversary does not know the probabilities with which links change state.

6.1.3 Routing Strategy. We use the φ randomized multi-hop routing strategy that we introduced in Sec. 4.1.4. The adversary knows that a shortest path-based routing strategy is being used but does not know the value of φ. If an estimate of φ were known to the adversary, this could be accounted for by changing how $\mathbf{B}$ is set. Note that the link dynamics do not change the shortest paths in the network and so do not affect $\mathbf{B}$. This is because links are i.i.d and which links will be up or down over time cannot be predicted.

6.1.4 Medium Access Control. We assume discrete time and that the duration of a timestep is long enough for every device in the network to transmit one packet.

6.1.5 Traffic. We randomly choose F flows with replacement and simulate the flows for T timesteps. We assume no control traffic is generated. We model the number of data packets that arrive on each flow according to a Poisson process with rate $\lambda = 0.5$. The adversary does not know the number of flows present nor which subsets of devices comprise sources or destinations of flows. If such information were known, it could be used to set the initial state. While the traffic generated on each flow is random, the rate of traffic is the same for all flows. This scenario should be more difficult for the adversary than one with heterogeneous traffic arrivals.



(a) True, Line (b) Inferred, Line

(c) True, Lattice (d) Inferred, Lattice

Figure 3: Examples of true (P_T) vs. inferred (P_F) flow distributions in line and lattice topologies for $N = 9$. Results are for static ($p = 1$ and $q = 0$) unicast links, $\varphi = 1$, $\lambda = 0.5$, and $T = 200$. The y-axes indicate the 9 possible source devices and the x-axes indicate the 9 possible destination devices. The value at a square (x, y) represents the (inferred) proportion of total network traffic sent from source y to destination x. In these plots, traffic comprises 2 flows: from source 1 to destination 4, and from source 5 to destination 9. Using these distributions and Eq. 7, we get $U_{norm} = 0.526$ for the line topology, and $U_{norm} = 0.477$ for the lattice topology.

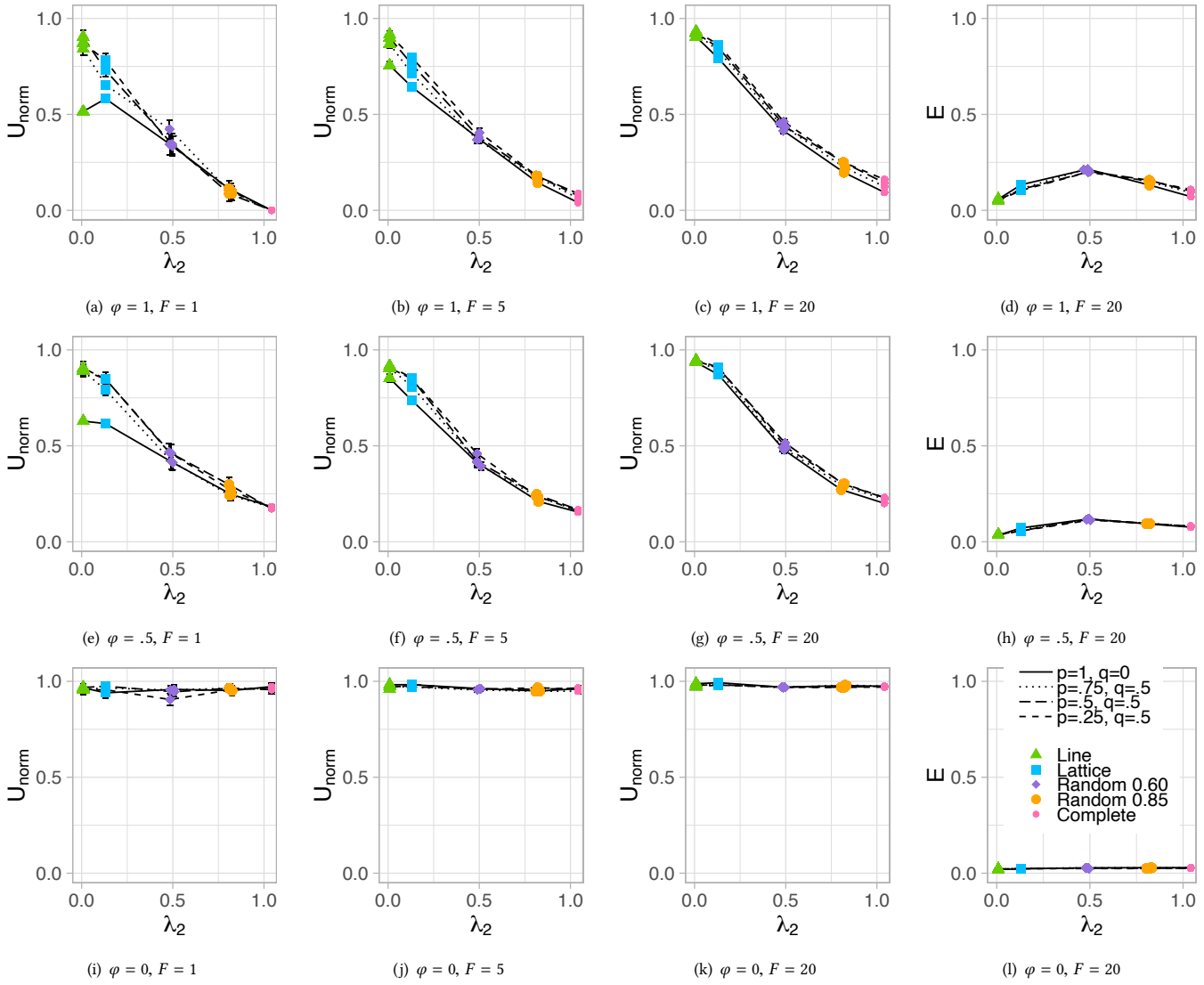
6.2 What impacts unlinkability?

Fig. 3 gives simulation examples of true and inferred flow distributions and the associated U_{norm} values to provide intuition about how we compute unlinkability. Figs. 4 and 5 give simulation results of how unlinkability changes for different network scenarios.

6.2.1 Impact of number of flows. Figs. 4 and 5 plot unlinkability as a function of algebraic connectivity, λ_2. Each simulation is executed for $T = 200$ timesteps. For each topology, we run 100 simulations, choosing different sets of F flows randomly with replacement. We show 95% confidence intervals, with the points colored according to the associated network topology. Different lines indicate different settings for the probability that links stay up, p, or down, q.

In Fig. 4, for unicast links and no link dynamics (i.e., the solid line where $p = 1, q = 0$), we see that for a given value of routing randomness φ, as the number of flows, F, increases, unlinkability increases. This is because having more flows provides more opportunities for flows to cross paths. In Fig. 5, for broadcast links, we see similar behaviour.

It is important to note that, for $F = 1$, unicast links, and $\varphi = 1$ in the line topology, unlinkability *is not* 0, even though no explicit



(a) $\varphi = 1$, $F = 1$ (b) $\varphi = 1$, $F = 5$ (c) $\varphi = 1$, $F = 20$ (d) $\varphi = 1$, $F = 20$

(e) $\varphi = .5$, $F = 1$ (f) $\varphi = .5$, $F = 5$ (g) $\varphi = .5$, $F = 20$ (h) $\varphi = .5$, $F = 20$

(i) $\varphi = 0$, $F = 1$ (j) $\varphi = 0$, $F = 5$ (k) $\varphi = 0$, $F = 20$ (l) $\varphi = 0$, $F = 20$

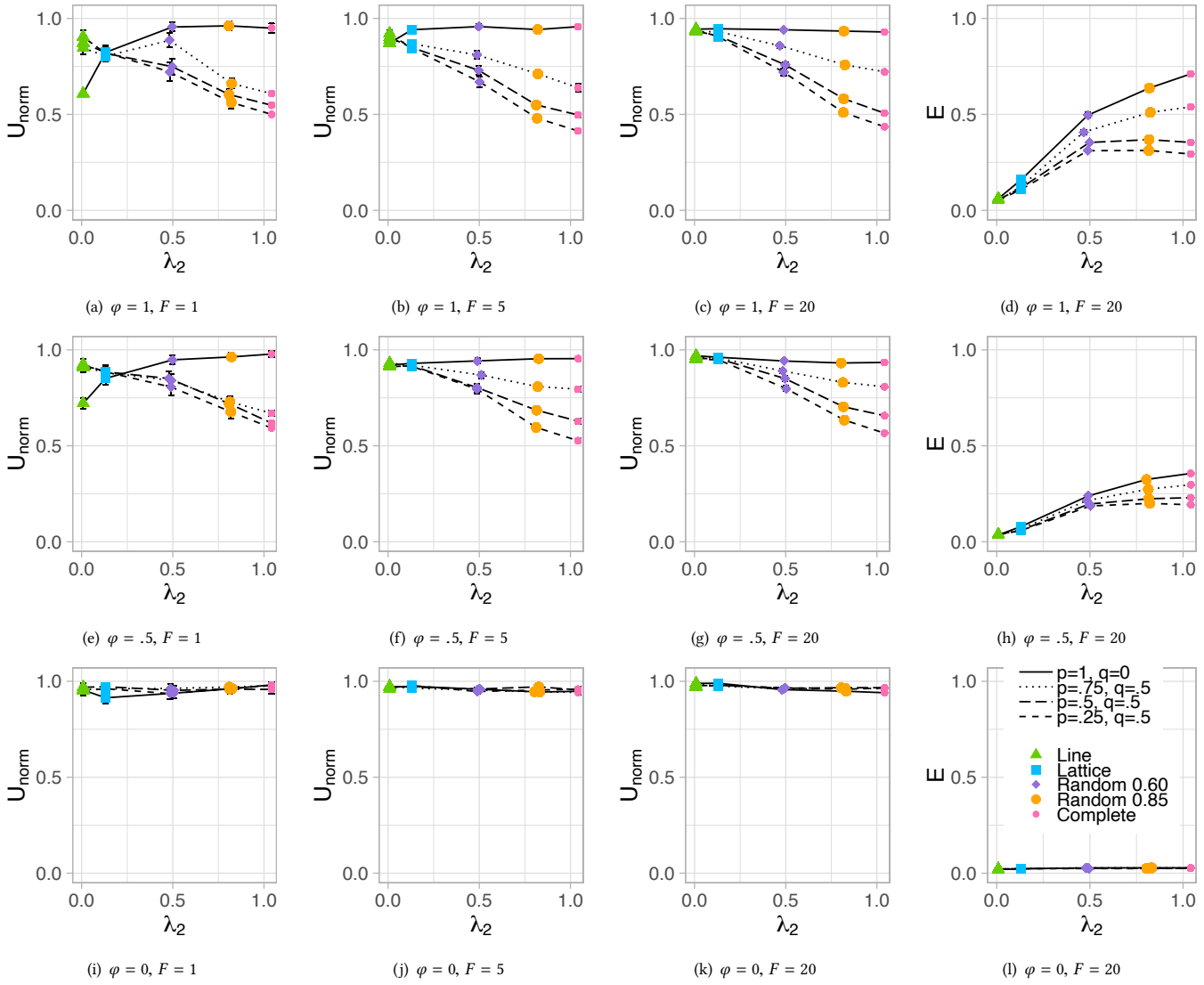
Figure 4: Network scenarios with unicast links. Plots show unlinkability, U_{norm}, and anonymization efficiency, E, as a function of algebraic connectivity, λ_2, for $N = 25$.

attempt is made to increase traffic mixing. This is due to multi-hop routing: the adversary must consider that the possible intermediate hops on the path between a potential source and destination might themselves be potential sources and destinations. With unicast links, unlinkability is greater than zero as long as there are shortest paths in the topology that are more than 1 hop long. In the case of the complete graph, the unlinkability *is* 0 because all shortest paths are only 1 hop long, so there are no potential intermediate sources and destinations, and so no mixing.

6.2.2 Impact of routing randomness. In Figs. 4 and 5, as routing randomness increases (that is, φ decreases from 1 to 0), unlinkability generally increases, regardless of the number of flows, link type, or topology. This is in part because all possible sub-paths along a path must be considered as potential flows due to multi-hop routing. For instance, with increased routing randomness, it takes longer for any packet to reach its destination, since the packet must pass through more intermediate devices that must be considered potential sources and destinations. Regardless of link type, depending on the number of flows and network topology, less routing randomness is needed to achieve the same level of unlinkability.

6.2.3 Impact of link type. Comparing Figs. 4 and 5 shows that unicast scenarios generally have lower unlinkability than broadcast scenarios. For $\varphi = 0$, i.e., maximum routing randomness, there is little difference in unlinkability between unicast vs. broadcast links.

However, for each link type there is a split based on network connectivity. For lower connectivity topologies like the line and



(a) $\varphi = 1$, $F = 1$ (b) $\varphi = 1$, $F = 5$ (c) $\varphi = 1$, $F = 20$ (d) $\varphi = 1$, $F = 20$

(e) $\varphi = .5$, $F = 1$ (f) $\varphi = .5$, $F = 5$ (g) $\varphi = .5$, $F = 20$ (h) $\varphi = .5$, $F = 20$

(i) $\varphi = 0$, $F = 1$ (j) $\varphi = 0$, $F = 5$ (k) $\varphi = 0$, $F = 20$ (l) $\varphi = 0$, $F = 20$

Figure 5: Network scenarios with broadcast link. Plots show unlinkability, U_{norm}, and anonymization efficiency, E, as a function of algebraic connectivity, λ_2, for $N = 25$.

lattice, regardless of whether unicast or broadcast links are used, high unlinkability is possible only when there are many randomly chosen flows or routing is via a highly random walk. For high connectivity topologies like the geometric random graph or complete graph, high unlinkability is only possible when either broadcast links are used or routing is via a highly random walk.

Thus, broadcast transmissions are primarily beneficial when the topology is well-connected. If the topology is not well-connected then broadcast is not sufficient by itself: instead, many randomly chosen flows or very random walks are needed. While other work has shown the power of broadcast transmissions [36], here we see with multi-hop routing that the benefits of broadcast are dependent on additional characteristics of the network scenario.

6.2.4 Impact of network topology. In Fig. 4, for unicast links and no link dynamics (i.e., $p = 1, q = 0$), as algebraic connectivity, λ_2 increases, unlinkability generally decreases except when $\varphi = 0$. This is because when connectivity is higher, paths are shorter, and so there are fewer devices potentially involved in flows, and so there are fewer possible flows to consider which lowers unlinkability. When connectivity is lower, so paths are longer, there are more devices potentially involved in flows which means there are more flows to consider which increases unlinkability. Note that for the random topologies, the mean λ_2 is plotted, since each simulation is for a different randomly generated topology.

Conversely, in Fig. 5, for broadcast links and no link dynamics, as λ_2 increases, unlinkability increases except when $\varphi = 0$ or $F = 20$.

Now, each broadcast increases the number of devices that receive a transmission, thus increasing the number of possible flows that must be considered. For networks with higher connectivity, each broadcast reaches more devices, increasing unlinkability. For networks with lower connectivity, even though each broadcast reaches fewer devices, flows must use longer paths, which means more devices are still reachable and must be considered in possible flows.

6.2.5 Impact of link dynamics. The dotted and dashed lines in Figs. 4 and 5 are for scenarios when links are dynamically changing. In Fig. 4, unicast link dynamics result in higher unlinkability but only when connectivity is low. Essentially, when the network is sparsely connected, unicast link dynamics may prevent a device from making a transmission, and as a result, can change the probabilities that packets are destined to that device or its neighbors.

Conversely, in Fig. 5, broadcast link dynamics result in higher unlinkability when connectivity is low and lower unlinkability when connectivity is high. Essentially, when a given device makes repeated broadcast transmissions over time, if different subsets of the component unicast links to the device's neighbors are down due to link dynamics, then this can give information about which neighbor a transmission is intended for. When the network is sparsely connected, the broadcast scenario is similar to the unicast scenario, with link dynamics potentially preventing a device from making any transmission.

6.2.6 Anonymization efficiency. The last columns of Figs. 4 and 5 plot anonymization efficiency as a function of algebraic connectivity, λ_2, for unicast and broadcast links respectively. We show only the anonymization efficiency results for $F = 20$ since the $F = 1$ and $F = 5$ results are very similar. In Fig. 4 for unicast links, anonymization efficiency is highest for the lattice and random graph topologies, except when $\varphi = 0$. The lattice, however, achieves significantly higher unlinkability. We conjecture that in terms of efficiency, the lattice best trades-off having paths that are not too short so that intermediate hops must be considered as potential sources and destinations, with having paths that are not too long and thereby incurring too many transmissions. In Fig. 5 for broadcast links, anonymization efficiency increases as λ_2 increases, unless $\varphi = 0$. Generally, anonymization efficiency is higher for broadcast links than unicast links, except when $\varphi = 0$.

6.3 Discussion

Our results confirm that traffic mixing is possible from multi-hop routing even when devices themselves do not reorder traffic. Our results also give insight into how best to control the network structure on which unlinkable communication protocols might run. That is, depending on the network characteristics, it may not always be necessary to delay and mix traffic at devices in order to increase unlinkability.

In our simulations, we observe that the lattice topology most consistently supports high unlinkability (with U_{norm} never less than about 0.6) regardless of link type, routing randomness, number of flows, or link dynamics. This suggets that when it is possible to control network connectivity, a lattice is a good target topology when other network conditions are unknown.

We did, however, observe significant differences between unicast and broadcast links. Sparsely connected topologies improved unlinkability in unicast scenarios but degraded unlinkability in broadcast scenarios. This suggests that depending on the link type, controlling the topology to be more or less sparse is beneficial. Because link dynamics improved unlinkability in sparsely connected topologies for both unicast and broadcast scenarios, this suggests that artificial link dynamics could be beneficial. Delay tolerant networks which have broadcast links that are mostly down (large q and small p link dynamics) should have good unlinkability.

Routing randomization was also shown to be consistently helpful at increasing unlinkability. In practice for scenarios with low unlinkability, rather than routing all flows by (mostly) random walks there may be benefits to a more fine-grained approach. For instance, when there are few flows, routing can be done by a random walk. As the number of flows increases, only a subset of flows need be routed randomly. An alternative approach would be to add an additional set of cover traffic flows to the network that are long-lived and routed randomly, with the number of cover traffic flows changing as some function of the number of real flows in the network.

6.4 Scalability

We chose to model states and observations as multivariate Gaussian random variables to reduce the number of dimensions. The Kalman filter implementation we use, FKF [21], was chosen for its ability to work with large state spaces. The programming language R, however, itself has a maximum vector length and array dimension limit of $2^{31} - 1$. Experimentally, we have found that the largest networks for which we have been able to construct a Kalman filter and simulate before hitting this limit have been for $N = 64$ devices. For $N = 64$, the $\mathbf{A}$ matrix is of size 8192×8192 and cannot be represented sparsely due to the Kalman filter computations. Since we require 18,000 simulations to obtain the data to make our plots (from 5 topologies times 3 values of φ times 3 values of F times 4 sets of p and q values times 100 simulations for statistical significance). Thus, due to the memory and simulation time required to run simulations with larger N values, even using cloud resources, the largest N value that we simulate in this work is $N = 25$.

Our goal, however, is not an algorithm to run in real-time for large networks, but instead to quantify what impacts unlinkability in multi-hop wireless networks which are typically smaller in size. While we consider small networks, they still give insight, such as how to prevent poor unlinkability subnetworks in large networks.

7 CONCLUSIONS

In this work, we have quantified the unlinkability achievable when traffic mixing is due to multi-hop routing and broadcast transmissions, rather than mixing at individual devices. To do this, we formulated a Kalman filter adversary who passively observes all packet transmissions that occur in a multi-hop wireless network in which devices also act as anonymizing routers. The adversary uses these transmissions to compute a probability distribution over the possible flows present in the network. From this flow distribution we derived an unlinkability metric that we analyzed in simulation. We showed that i) for unicast links, as network connectivity increases, unlinkability decreases since less traffic mixing is possible,

while for broadcast links as connectivity increases unlinkability increases, ii) link dynamics increase unlinkability in poorly connected topologies regardless of link type, iii) more well-connected topologies are able to achieve the same level of unlinkability with fewer transmissions per packet delivered, and iv) a lattice topology has consistently good unlinkability in different network scenarios.

In future work, we would like to scale our simulations by either approximating the Kalman filter state space or using approximate inference methods. The applicability of non-linear Kalman filters as well as particle filters would also merit investigation. We would also like to explore the impact of more realistic physical and link layers as well as mobility models, which would enable analysis and infrence of performance metrics like throughput and delay. Finally, we would like to consider adversaries that may only have partial information about the network topology, as well as active adversaries able to intelligently jam transmissions to decrease unlinkability. Our long-term goal is to devise unlinkable communication protocols using our insights.

8 ACKNOWLEDGEMENTS

The authors are grateful to Danny Krizanc for many helpful discussions, and thank Amir Herzberg and Bing Wang for helpful feedback on the paper. The authors also thank the anonymous reviewers for their helpful comments. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC and consultation resources that have contributed to the research results reported within this paper.

REFERENCES

[1] Amos Beimel and Shlomi Dolev. 2003. Buses for Anonymous Message Delivery. *Journal of Cryptology* 16, 1 (2003).
[2] Ron Berman, Amos Fiat, Marcin Gomulkiewicz, Marek Konowski, Miroslaw Kutylowski, Tomer Levinboim, and Ammon Ta-Shma. 2015. Provable Unlinkability Against Traffic Analysis with Low Message Overhead. *Journal of Cryptology* 28 (2015), 623–640.
[3] Matt Blaze, John Ioannidis, Angelos D Keromytis, Tal G Malkin, and Avi Rubin. 2009. Anonymity in wireless broadcast networks. (2009).
[4] David L. Chaum. 1981. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Commun. ACM* 24, 2 (Feb. 1981).
[5] Chen Chen, Daniele E Asoni, David Barrera, George Danezis, and Adrain Perrig. 2015. HORNET: High-speed onion routing at the network layer. In *ACM SIGSAC Conference on Computer and Communications Security*.
[6] Fan RK Chung. 1996. Lectures on spectral graph theory. *CBMS Lectures, Fresno* 6 (1996), 17–21.
[7] George Danezis. 2003. Mix-networks with restricted routes. In *International Workshop on Privacy Enhancing Technologies*. 1–17.
[8] Jing Deng, Richard Han, and Shivakant Mishra. 2006. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. *Pervasive and Mobile Computing* (2006).
[9] Claudia Diaz, Steven Murdoch, and Carmela Troncoso. 2010. Impact of Network Topology on Anonymity and Overhead in Low-Latency Anonymity Networks. In *Privacy Enhancing Technologies*.
[10] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. *Tor: The second-generation onion router.* Technical Report. Naval Research Lab Washington DC.
[11] Roger Dingledine, Vitaly Shmatikov, and Paul F Syverson. 2004. Synchronous Batching: From Cascades to Free Routes. In *PETS*, Vol. 4. Springer.
[12] Lars Fischer, Stefan Katzenbeisser, and Claudia Eckert. 2008. Measuring unlinkability revisited. In *ACM workshop on Privacy in the electronic society*.
[13] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. 2004. Universal Re-encryption for Mixnets. In *Topics in Cryptology – CT-RSA 2004*. 163–178.
[14] Thaier Hayajneh, Razvi Doomun, Prashant Krishnamurthy, and David Tipper. 2011. Source destination obfuscation in wireless ad hoc networks. *Security and Communication Networks* 4, 8 (2011), 888–901.
[15] Dijiang Huang. 2008. Unlinkability measure for IEEE 802.11 based MANETs. *IEEE Transactions on Wireless Communications* 7, 3 (2008), 1025–1034.
[16] Rudolph Emil Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering* 82, Series D (1960), 35–45.
[17] Jiejun Kong and Xiaoyan Hong. 2003. ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *ACM International symposium on Mobile ad hoc networking & computing*.
[18] Stefan Köpsell and Sandra Steinbrecher. 2003. Modeling unlinkability. In *Proceedings of the Third Workshop on Privacy Enhancing Technologies*.
[19] Brian N Levine, Michael K Reiter, Chenxi Wang, and Matthew Wright. 2004. Timing attacks in low-latency mix systems. In *International Conference on Financial Cryptography*. Springer, 251–265.
[20] Yunzhong Liu, Rui Zhang, Jing Shi, and Yanchao Zhang. 2010. Traffic inference in anonymous manets. In *IEEE SECON*.
[21] David Luethi, Philipp Erb, and Simon Otziger. 2018. FKF: Fast Kalman Filter. R package version 0.1.5. *https://cran.r-project.org/web/packages/FKF/index.html* (2018).
[22] Alberto Medina, Nina Taft, Kave Salamatian, Supratik Bhattacharyya, and Christophe Diot. 2002. Traffic matrix estimation: Existing techniques and new directions. In *ACM SIGCOMM Computer Communication Review*, Vol. 32. ACM, 161–174.
[23] Prateek Mittal and Nikita Borisov. 2009. Shadowwalker: peer-to-peer anonymous communication using redundant structured topologies. In *ACM conference on Computer and communications security*.
[24] Prateek Mittal, Matthew Wright, and Nikita Borisov. 2012. Pisces: Anonymous communication using social networks. *arXiv preprint arXiv:1208.6326* (2012).
[25] Marie Elisabeth Gaup Moe. 2009. Quantification of anonymity for mobile ad hoc networks. *Electronic Notes in Theoretical Computer Science* 244 (2009), 95–107.
[26] Vakul Mohanty, Dhaval Moliya, Chittaranjan Hota, and Muttukrishnan Rajarajan. 2010. Secure anonymous routing for MANETs using distributed dynamic random path selection. In *Pacific-Asia Workshop on Intelligence and Security Informatics*. Springer, 65–72.
[27] Shishir Nagaraja. 2007. Anonymity in the wild: Mixes on unstructured networks. In *International workshop on privacy enhancing technologies*. 254–271.
[28] Open Garden. 2019. Firechat Messaging App. https://en.wikipedia.org/wiki/FireChat.
[29] Andreas Pfitzmann and Marit Hansen. 2010. Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management, Internet Draft (Expired). https://tools.ietf.org/id/draft-hansen-privacy-terminology-00.html.
[30] Yang Qin, Dijiang Huang, and Bing Li. 2013. STARS: A statistical traffic pattern discovery system for MANETs. *IEEE Transactions on Dependable and Secure Computing* 11 (2013).
[31] Charles Rackoff and Daniel Simon. 1993. Cryptographic Defense Against Traffic Analysis. In *STOC*.
[32] Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, et al. 2018. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*. IEEE, 1–6.
[33] Stefaan Seys and Bart Preneel. 2006. ARM: Anonymous routing protocol for mobile ad hoc networks. In *International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)*.
[34] Vitaly Shmatikov and Ming-Hsiu Wang. 2006. Measuring relationship anonymity in mix networks. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*. ACM, 59–62.
[35] Augustin Soule, Kavé Salamatian, Antonio Nucci, and Nina Taft. 2005. Traffic matrix tracking using kalman filters. *ACM SIGMETRICS Performance Evaluation Review* 33, 3 (2005), 24–31.
[36] Frank Stajano and Ross Anderson. 1999. The cocaine auction protocol: On the power of anonymous broadcast. In *International Workshop on Information Hiding*.
[37] Carmela Troncoso and George Danezis. 2009. The bayesian traffic analysis of mix networks. In *ACM conference on Computer and communications security*. ACM, 369–379.
[38] Validity Labs. 2020. HOPR Messaging App. https://hopr.network/.
[39] Jelle Van Den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. 2015. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Symposium on Operating Systems Principles*.
[40] Yehuda Vardi. 1996. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American statistical association* 91, 433 (1996), 365–377.
[41] G. Welch and G. Bishop. 1995. *An Introduction to the Kalman filter.* Technical Report TR95-041. U of North Carolina at Chapel Hill, Dept. of Computer Science.
[42] Ye Zhu, Xinwen Fu, Bryan Graham, Riccardo Bettati, and Wei Zhao. 2009. Correlation-based traffic analysis attacks on anonymity networks. *IEEE Transactions on Parallel and Distributed Systems* 21, 7 (2009), 954–967.