Rangzen: Anonymously Getting the Word Out in a Blackout

Abstract

Recent years have seen the rise of a new type of state-level adversary. Governments have shown themselves willing to both surveil their citizens using Internet infrastructure and impose blackouts to shut off that very infrastructure during times of civil strife. However, it is exactly during such strife that citizens need reliable and anonymous communications the most.

In this paper, we present Rangzen, a system for anonymous broadcast messaging during Internet and cellular network blackouts. Rangzen is distinctive in both aim and design. Our aim is to provide an anonymous, one-to-many messaging layer that requires only users' smartphones and can withstand network-level attacks. Our design is a delay-tolerant mesh network which deprioritizes adversarial messages by means of a private social graph. We built a complete prototype app that runs on Android smartphones, present benchmarks of its performance and functionality, and analyze the challenges we faced building such an app on a modern mobile platform. Additionally, we present extensive simulation results demonstrating Rangzen's efficacy at scale.

1 Introduction

Over the past decade, the balance of power between citizens and governments has tilted inexorably in the latter's direction. Though there was once a perception of the Internet as a land of radically free communication and activism, it has become clear that like any key infrastructure, the Internet exists under centralized control [60]. While this has proved to be a boon to the average user under placid conditions—large companies are able to deliver efficient and reliable services and governments are able to police networks for criminal activity—in times of unrest, this has proven dangerous to those who would use the Internet as a forum to speak out.

Indeed, during societal unrest, centralized infrastructure can be easily co-opted. In recent years, authorities in Egypt, Syria, and Iran, among others, have shut down their already heavily-surveilled Internet access during times when citizens were questioning those very authorities' political legitimacy [18, 25, 59]. However, it is exactly in such moments that citizens need the ability to communicate without restriction and risk of retribution.

In this paper, we present Rangzen, an architecture for anonymously spreading information at a large scale during a communications blackout. We also present an open-source implementation of Rangzen as an Android smartphone app utilizing Bluetooth and Wifi Direct to communicate from pocket to pocket between stock, non-rooted, Android 4.0 and greater devices.

Two main challenges characterize our scenario: a) a lack of functioning infrastructure (neither Internet nor cell networks), and b) powerful adversaries intent upon widespread interference. We address the first challenge by broadcasting messages over a delay-tolerant, smartphone-based, mobile mesh network. This architecture enables reliable (albeit high-latency) communication. We address the second challenge with a social trust-based prioritization algorithm. Counterintuitively, our prioritization approach allows users to receive trustworthy messages without knowing their origin or author; this property preserves authors' anonymity.

More concretely, our solution leverages the real-world properties of social trust and human mobility, each of which is represented as a graph over individuals. The social trust graph is built by users via pairwise user trust establishments ("friending"), while the physical proximity graph is implicit in the users' time-varying locations. Given this foundation, during message propagation we attempt to separate the wheat from the chaff. Such a task is difficult in any decentralized, anonymous system—adversaries can flood the network with bogus messages, and the lack of point-to-point communication makes organizing content difficult. We address this by assigning trust to messages proportional to the number of *common friends* we have with the node that forwarded the messages. This social trust information is compared in a pri-

vate protocol, by cryptographically computing the number of common friend IDs. Communication partners remain anonymous to one another, and do not learn one another's friends, or even which friends they have in common. The trust information derived from exchanges with other nodes is used to prioritize messages, determining which are forwarded, which are dropped from the system, and if or how these messages are shown to users. Communicating the system's trust in messages to users, or not displaying low trust messages to users at all, is critical to preventing the intrusion of propaganda messages into users' communications. However, eavesdroppers and adversarial nodes participating in the network can see message priority scores. To avoid leaking information about users' trust patterns, we add noise to each message's priority score before transmitting it, so that the scores do not reveal too much about the true message priority on the sender's phone. We evaluate the privacy implications of this in §7.3.

Our contributions are threefold. First, we present the Rangzen architecture, carefully defining the interfaces presented to the user and used between devices to achieve the larger goal of enabling trustworthy communication while ensuring user anonymity. Second, we present a working prototype of Rangzen, implemented as an Android app, which demonstrates all aspects of the message passing and anonymity protocol. We measure the performance of this prototype and find that its behavior in terms of latency, bandwidth, range of operation, and battery usage are sufficient for Rangzen to be used in practice. Third, we show results from at-scale simulations that suggest Rangzen's viability as a city or regional scale communications medium.

Designing the Rangzen architecture itself gave us important insights into this modality of anonymous communications. The protocol represents a realistic system that addresses challenges real people face in the modern world. Our implementation process led to a deep understanding of the challenges faced when developing these types of systems on modern mobile platforms. While the hardware is willing, the operating system often presents roadblocks. Many of these challenges consist of important limitations of the operating system and application programming environment that were presumably included by designers to improve user experience in the common case, but which make building unusual apps like Rangzen cumbersome and lead to unsustainable software engineering practices like relying on hidden APIs. We discuss the compromises we made building our prototype in §5 and some lessons we learned for building unusual apps on mobile platforms.

Our benchmarks suggest Rangzen works in practice: devices in proximity for under 10 seconds can find one another, compare friends, and exchange several hundred messages. Even in the unrealistic worst case of constant communication, Rangzen uses only about 5.5% of the battery of a Nexus 5 phone per hour, suggesting that under realistic conditions its battery load would be quite reasonable. Finally, our simulations suggest that a peer-to-peer messaging system with an implementation like our prototype may be able to provide useful messaging within a city or region. Specifically, they suggest that Rangzen is capable of reaching over 90 percent of a city-wide userbase within 24 hours of the message's composition. Additionally, Rangzen effectively hampers the spread of propaganda; in simulation, a propagandist's messages reached 30 percent fewer people than a legitimate user's messages over a period of 48 hours.

2 Background and Motivation

The design of a communication system for use in a government-imposed Internet blackout is a complex undertaking. Social and political dynamics constrain both user desires and government actions.[1] Understanding these constraints is crucial to designing and building a system that is realistic in its aims and limits. In this section we present an abridged analysis of relevant sociopolitical turmoil and its implications for Rangzen.

2.1 Technology's Role in Revolution

Recent years have seen both successful and failed political revolutions in Tunisia, Libya, Egypt, Syria, Iran, Thailand, and Ukraine [7, 8, 30, 69], and large-scale social demonstrations in Greece, Turkey, Spain, Canada, Hong Kong, and the United States [9, 10, 15]. In most (if not all), smartphone- and Internet-based organizing was prevalent across social demographics. Governments responded in kind, leveraging the same communications networks not only to surveil [33] but also to communicate directly with the population [67].

Implication: Smartphones are now sufficiently widespread that they can serve as democratic means for communication, though the communication they enable must be secured.

2.2 The Need for Anonymity

Technology's role in the spread of dissent is somewhat ambiguous. Although it certainly enables the spread of information, in many cases that very information has helped governments to target activists to an unprecedented degree. This targeting ranges from identifying dissenters online to harassing citizens who possess out-of-the-ordinary communication hardware [31].

Implication: During periods of social unrest, communication systems should protect individuals from being directly linked to objectionable content.

[1] We are in active contact with dissidents and activists in countries that restrict political speech and organizing to gain a better understanding of their needs; this has informed our design choices for Rangzen.

2.3 The Need for Robustness

The communications blackouts we consider in this paper are not accidents, and are due to a desire to stem the tide of public discontent. Nevertheless, the means by which blackouts have been implemented and their degree of totality have varied considerably. Some blackouts have involved BGP route withdrawals [21], while others have been more severe cuts [61]. In addition, total blackouts are made easier for governments to impose by a non-diverse network infrastructure with few providers and/or heavy government control. During these blackouts, governments are likely to try to thwart any temporary workarounds used by the population.

Implication: The cause or mechanism of a blackout should not impact the subsequent operation of the system, and the system should resist unilateral shutdown or co-option.

2.4 The Paradox of Fast Communication

Recently, political theorists have studied the impermanence of so-called "Twitter revolutions", commenting on the absence of slow, steady community organizing and in-person contact that has led to lasting political movements in the past [64]. Indeed, movements over the last century have often built momentum over decades, only to come to a sudden point of crisis involving drastic government actions such as the imposition of martial law or communications blackouts. The determining factor in the outcomes of such crises is largely an enigma, though the strength of the underlying movements has been a crucial factor [37]. While only history will adjudicate the impact of modern communications tools in political organizing, we believe the evidence suggests that rapid communication is not crucial.

Implication: The system need not enable rapid communication, but it should enable trustworthy and reliable communication.

3 Related Work

The literature contains a wide body of work on anonymous communication systems. Most existing work targets point-to-point communication over dedicated infrastructure such as Tor, Tarzan, Crowds, and Free Haven [1, 5, 26, 27, 34, 58]. Another, less well-known branch of work focuses on point-to-point anonymous communication in the *absence* of dedicated infrastructure [39, 46, 52]. Our work differs from both of these areas by focusing on *broadcast* communication in the absence of dedicated infrastructure. To the best of our knowledge, existing work does not explicitly consider this problem space, though individual components have been studied in depth. Algorithmically, the observations described previously require Rangzen to address a tension between anonymity (which demands that users' identities be hidden) and robustness to network-level attacks (which requires some notion of reputation or identity). We subdivide this body of literature into anonymous broadcast communication, and DTN security. A number of cryptographic, anonymous broadcast protocols exist [14, 16, 20, 45], but it is unclear whether such protocols are well-suited for large-scale adoption in resource-starved environments. The problem is particularly challenging in a blackout, which prevents the use of online trust mechanisms, such as Bitcoin-based protocols [50]. On the other hand, much of DTN security research has focused on point-to-point communication [12, 29, 32]; our problem is more related to filtering content in a distributed and privacy-preserving way. Any solution must be lightweight to function during short opportunistic encounters. Many practical and academic anonymity systems attempt to resolve this tension using pseudonymous reputation systems [6, 39], but using pseudonyms increases susceptibility to side-channel correlation attacks [24, 51, 57].

Our approach relates to Sybil defense [55, 66, 68, 71] through our use of social graph structure to distinguish users. Similarly, our approach builds on a rich literature on distributed, privacy-preserving trust [35, 38, 62], and more specifically, private set intersection over social data to achieve privacy and anonymity [43, 44, 63].

4 Architecture and Protocol

Each aspect of Rangzen's design is based on the principles we discussed in §2. The prevalence of smartphones and their use in community organizing enables the basic architecture: a mesh composed entirely of smartphones. The delay tolerance of this system emerges directly from the need for robustness and the "Paradox of Fast Communication" (§2.4). Real-time networks over meshes are difficult and potentially fragile, while our delay tolerant system is both robust and provides sufficiently fast (messages propagating within tens of hours or a couple days) communication to enable community organization. Rangzen provides anonymity (§2.2) to authors to enable free speech without persecution, and it sidesteps attacks based on pseudonymity by providing full anonymity instead. To provide trust in messages and suppress propaganda and spam, we filter and prioritize based on social trust between mesh nodes at propagation time, rather than directly between readers and authors. Borrowing from sybil defense schemes [71], Rangzen assumes that messages are more trustworthy if they reach a reader via a path of trusted nodes. In Rangzen, we determine trust without knowing the identity of the node which forwarded the message. Using a privacy-preserving set intersection protocol, we learn how many common friends we have with each forwarding node a client communi-

cates with; the more friends in common, the more trust we have in the node, and in the messages it forwards.

4.1 Threat Model

Our adversary is a state-level actor capable and willing to disable infrastructure such as cellular networks and ISP networks providing links to the Internet. The adversary's goals are to *disrupt communication* of its enemies, or to inject false information (i.e. *propaganda*) into their communications. The adversary may have significant monetary resources but it lacks a widespread ability to befriend or infiltrate dissident social circles. In this respect the attacker is like a typical adversary which deploys sybils against a system – although it may be physically well distributed and possess a high level of technological and financial resources, its weakness is an inability to socially infiltrate its enemies. The adversary's sybils will nearly always have fewer friendships with honest nodes than the honest nodes have among themselves. This is a common assumption in the Sybil defense literature [66, 71]. A worst-case adversary might violate this assumption by recruiting spies from the general population. However, even the most heavy-handed regimes in history—such as the Stasi—recruited a fraction of citizen informants no larger than one in fifty [41]. Moreover, if there are no characteristics distinguishing adversaries from regular citizens, early work implies that the Sybil detection problem cannot be solved in the first place [28].

Non-Goals.

We assume that the adversary has the resources to single out individual users and perform severe, targeted, violent or social attacks against them. We do not protect against such targeted attacks. Rangzen aims to make it infeasible for the adversary to *scale* their attacks, making it impossible for them to deceive a large percentage of the population, or to disrupt communications on a large scale. We consider preventing the *scaling* of privacy violations beyond a small percentage of citizens to be a fundamental win, as have some privacy scholars [36].

Message content isn't private in Rangzen. As with a public system like Twitter, messages are public. (By contrast, authorship is confidential and protected). Further, the use of Rangzen is detectable by an attacker. The attacker can participate in the system, or eavesdrop on those who do. Rangzen doesn't seek to hide the fact that Rangzen is being used. Instead, it relies on decentralization to protect it from attack. Hiding the identities of devices from an eavesdropper is orthogonal to our goals. While our current implementation doesn't itself hide the device's identifiers (e.g. Bluetooth and Wifi MAC addresses), a rooted device could randomize the MAC address. This type of privacy may be needed in some but not all circumstances to protect the safety of users, and we will incorporate it into deployments as appropriate.

4.2 Design Overview

Delay-Tolerant Mesh Network.

Rangzen forms a mobile, ad-hoc, delay-tolerant network of smartphones. The network propagates microblogs—broadcasted, public datagrams, much like Tweets—which are passed opportunistically between devices in physical proximity. Messages will spread rapidly through a crowd of people who have Rangzen installed on their phones over the course of tens of seconds or minutes. Messages need not spread in real time; they are stored on devices and forwarded opportunistically when another device running Rangzen is encountered. Users need not actively participate in forwarding messages, as Rangzen runs in the background, periodically searching for nearby devices and sending and receiving messages to and from those devices. Thus messages will also spread over time through a city or region as people move, passing on the street, riding transit, or spending time together. We refer to the opportunistic exchange of messages between two devices, including the comparing of friends for trust establishment, as an *exchange*.

Prioritization Based on Mutual Friendships.

Rangzen clients trust messages forwarded to them by other clients with whom they share *mutual friends*. Each client stores a priority value in $[0, 1]$ with each message, which is displayed to users and used to order messages in the UI. Low-priority messages are displayed to the user at the bottom of the feed, or not at all. When two users meet and exchange of messages, low trust messages are the last to be forwarded. Low trust messages may be dropped entirely from storage by Rangzen clients.

Clients determine the trust to put in each message based upon their trust of the device which forwarded the message. Since Rangzen is a fully anonymous system, no authorship information is stored in messages. Trust must be based entirely upon relationships between the clients, who store and forward the messages. Trust between devices is based upon mutual friendships. When Alice's client receives messages from another user's client, Alice's client assigns trust to those messages proportionally to the number of friends they have in common.

These friendships are real-life relationships, and they must be formed in person through the exchange of secrets between the friends' smartphones. In our prototype, QR codes are used. Rangzen displays a QR code containing a hash of the user's Rangzen public key, which can be scanned to form a friendship. When two devices have an opportunistic exchange, they compare friends via a private set intersection protocol, learning the number of friends they have in common without revealing who their friends are, or even which friends are shared. Each client then assigns trust to messages received in the exchange

based on the number of common friends.

Suppressing Propaganda via the Friend Graph. Friendships in Rangzen can be viewed as forming a *trust graph* in which each user is a node, and graph edges represent real-life trust relationships between people. We leverage this trust graph to suppress the adversary's propaganda. Since the adversary cannot form friendships with real users on a widespread basis (see our threat model in §4.1), propaganda sent by adversarial nodes will reach genuine users through message exchanges including very few common friends. Thus Rangzen clients will assign lower degrees of trust to adversarial messages, compared to messages from genuine users. Propaganda messages will tend to be dropped automatically or hidden from user view, reducing their influence on the communications happening through Rangzen.

4.3 Peer Exchange Protocol

Rangzen's mesh communication is achieved via lightweight, pairwise message exchanges. Here we define the protocol two peers use when they encounter one another to propagate messages, intersect friend sets, and finally how each uses the information it has received to quantify social trust and discriminate between messages.

4.3.1 Lightweight Protocol

The Rangzen protocol attempts to be lightweight and simple, with the goal of enabling devices to perform it as quickly as possible over the longest possible ranges and without requiring excessive bandwidth. The protocol requires only 1 round trip, assuming that messages cross in flight, since each device first sends a "client" message and then a "server" reply to complete the bi-directional private set intersection. We leverage the PSI-Ca (Private Set Intersection Cardinality) protocol of Cristofaro *et al.*, which is ideal for our implementation context because it only relies upon standard groups and group operations that are available in standard crypto libraries [22]. This lets two nodes compute the number but not the identity of their mutual friends. The client message includes not only friendship set information but also the set of messages and priorities that the device knows of.

4.3.2 Social Trust Metric

Our basic assumptions are that a) people mostly believe and want to see messages from people socially well connected to them, and b) adversarial nodes are have a difficult time befriending people to share many mutual friends with many honest nodes. To capture these ideas, each pair of nodes computes a social trust score during each opportunistic encounter. This trust score is simply the number of shared mutual friends between the two nodes divided by each node's total number of friends. 0 friends result in a very small ε trust rather than 0, to maintain ordering between messages with very low trust and to ensure that new users can communicate. We let $T(a,b)$ denote how much a trusts b:

$$T(a,b) = \max\left(\frac{F(a) \cap F(b)}{F(a)}, \varepsilon\right)$$

where $F(a)$ denotes the set of a's friends, and ε is a small positive constant that ensures that ordering is preserved—even if the nodes share no mutual friends.

We expect users to have around 30 trusted friends, and limit the number of friends that can be submitted to any PSI interaction accordingly, since we consider the trust between people with (e.g.) 10 and 100 common friends to be similar. This restricts numbers of common friends to small integers, reducing the extent to which common friend degree can distinguish unique users.

4.3.3 Mapping from trust to priority

After receiving messages from another node, Alice's client must decide where to insert those messages in her feed. In our prototype, she simply multiplies the priority score of each message according to the sender by her trust of the sender. Our simulations explore the use of more complicated mechanisms for scaling priority of messages based on the trust of other users. These mechanisms may be more effective for ensuring message propagation and the effective filtering of propaganda. In our simulations, we actually use a distorted function that assigns more trust to senders with a threshold number of mutual friends. Additionally, to give message authors increased deniability, each sender adds noise to each message's priority score *before* sending it to a new node. We will see in §7.3 that this improves deniability.

To make this more precise, if a receives a message from b with priority $0 \leq p_o \leq 1$, then a will insert the message into her queue with a priority that is a sigmoidal function of the trust score:

$$Tr_0^1[(p_{b,a}(T(a,b)) \times p_o) + z_a],$$

where $z_a \sim \mathcal{N}(\mu, \sigma^2)$ is additive Gaussian noise used to improve message propagation,[2] $Tr_0^1[x]$ is a threshold forcing x to be in the range $[0,1]$, and

$$p_{b,a}(T(a,b)) = \frac{1}{1+\exp\{-\rho(T(a,b)-\tau)\}}. \quad (1)$$

Approximately, this means a will trust b fully if the ratio of mutual friends is greater than τ. We used $\rho = 13$ and $\tau = 0.3$ for a sigmoid that transitioned sharply in the range $[0,1]$. These constants would need to be tuned in a real deployment based on real mobility and friendship patterns. If a device runs out of storage, the lowest-priority messages get dropped first.

[2] This noise parameter helps unpopular nodes spread content by randomly increasing (or decreasing) priority scores, but it also improves author anonymity. We demonstrate these claims in §7.3.

4.4 Authoring and Message Management

Users are presented with an ordinary microblogging messaging interface. New messages written by the local user are initialized to have priority 1. If a user particularly likes a message from another node, she can manually "upvote" the message, which resets its priority score to 1. Note that since messages are anonymous, upvoting a message is equivalent to reauthoring it.[3] Finally, Rangzen decays the priority of messages over time so that out-of-date content gradually leaves the system.

4.5 Security Discussion

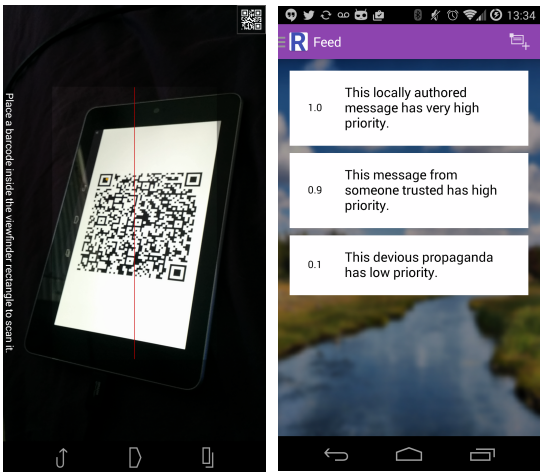
Rangzen combines friend establishment and private set intersection of friend identities into a protocol for anonymous message dissemination. An attacker spreading propaganda must do so via the Rangzen protocol. A Rangzen user will only store a new message if the message is authored by the node itself or received during a peer encounter. Attackers that do not corrupt the client software must therefore attack the peer encounter protocol to spread messages.

Unique device identifiers. An attacker may attempt to identify and reidentify individual devices and their owners that are using Rangzen. On rooted devices, Rangzen could be modified to randomize the device's MAC addresses on each exchange. This could be implemented if appropriate in a particular deployment.

Propaganda spread. Rangzen clients reject messages from peers with whom they cannot complete private set intersection. If the attacker performs the PSI protocol correctly, then their success at spreading propaganda relies on their ability to form trust relationships with users, which is their weakness (§4.1) Otherwise, the attacker can attempt to misbehave during the PSI protocol.

Attacking friend addition. Rangzen only adds friend IDs via in-person exchanges. Thus attackers must either capture devices or socially engineer targets to befriend users. An attacker who learns friend IDs can store them, forming directed edges in the graph. But since the protocol never shares friend identities, the attacker would have to befriend or harass a user to learn real identities. If the attacker frequently confiscates devices, encrypting friend IDs with a password could protect from the attacker gaining friends via copying IDs from confiscated devices. Alternatively, an ID revocation protocol could be built atop the message sending primitive of Rangzen. These enhancements are orthogonal to our design and could be implemented if the deployment setting called for them.

[3]It is imperative that users not inadvertently reveal their identities through message content; unfortunately, this is very difficult to prevent with technological solutions alone. As with all anonymous communication systems, user education is critical and interfaces must be designed in ways that make doing the safe thing easy.



(a) Adding a friend. (b) The feed of messages.

Figure 1: Rangzen screenshots.

Attacking trust computation. Our choice of PSI algorithm is a modular one — any other protocol could be substituted to compare friend sets. The protocol we used is only provably secure against semi-honest adversaries, but we believe there is nothing an attacker can do to further their goals under Rangzen's threat model.

Chosen-Input Attack. Adversaries can learn social graph edges by submitting a *single* real ID friend to the PSI protocol per encounter. If the intersection is cardinality 1, the attacker learns that their communication partner is friends with that ID. We call this a chosen input attack on the trust computation. If an adversary can identify peers it meets, this would at most allow the adversary to gradually learn as much of the social graph as described in Appendix B. It is unclear that this is the most efficient way for any adversaries to learn social connections between people; they might instead examine online social media. If this type of attack is a concern, clients can rate limit encounters to restrict the amount of information leaked through this channel. Rate limiting is also necessary for battery conservation, so some protection against this attack comes bundled with better battery life. Policies for this restriction are an area of future work on Rangzen. This attack requires prior knowledge of real client friend IDs.

Denial of Service. Attackers might attempt to launch denial of service attacks by overwhelming the system with messages. Neither storage nor bandwidth can be overwhelmed by such a flood of messages since the prioritization mechanism applies to all such messages, and clients can terminate exchanges that go on too long. We discuss attackers that jam the airwaves in §7.2.4.

5 Implementation

We implemented a prototype of Rangzen as an Android app. In building Rangzen, we had several goals, includ-

ing that it be easy for users to use correctly, efficient in terms of battery life, and able to propagate messages as quickly and at as large distances as possible. We used a combination of several technologies available on Android phones since OS version 4.0 (Ice Cream Sandwich) including Wifi Direct (known as Wifi P2P in Android) and Bluetooth. In building Rangzen, we discovered that Android puts up significant hurdles for implementing a mesh-based messaging app like this one. In this section we discuss some of those difficulties and the ways in which they limited and challenged us. We show screenshots from our app in Figure 1.

Our implementation consists of an Android app written in over 3600 lines of Java. Our test suite contains over 2100 lines of Java. We use JUnit (a standard Java unit testing framework) and Robolectric (a testing framework for Android that enables tests to run against the Android APIs without running the tests on an actual Android device or emulator) for our test suite.

We ran Rangzen on 7 models of Android devices: A Nexus 5, a first-generation Nexus 7, a second-generation Nexus 7, a Nexus 4, a Samsung Galaxy S4, a Samsung Galaxy S5, and an HTC One X. These devices were running stock Android 5.0, stock Android 4.4.4, stock Android 4.4.2, Cyanogenmod 11, a Verizon build of Android 4.3, and an AT&T build of Android 4.0.3 at various points in the testing process. We found that each new phone and OS version exhibited new bugs, but were able to work around those bugs with simple fixes once we understood the differences between devices and OS versions. Testing unusual apps like Rangzen on many devices is extremely important for robustness, as non-standard features have inconsistent platform support.[4]

5.1 Assumptions

We assumed modern but not cutting-edge devices and operating systems for the users of Rangzen. Our prototype functions on any version of Android which supports Wifi Direct (4.0 and greater). Android 4.0 was released in 2012 and our target versions include over 92% of Android devices operating today, according to Google [4].

We assumed that uptake for a system like this would be significantly lower if it required the user to root their device, to perform significant configuration, or if its passive operation required frequent user input. Thus we rejected approaches which burdened the user in any of these ways. This limited us to certain unconventional uses of networking technologies which, while functional, illustrate the fact that that Android makes it complicated to do what Rangzen does.

[4]We aim to have the Rangzen source undergo a professional security audit and release the app to the Play Store this summer.

5.2 Engineering Challenges

We found that it was difficult to implement mesh networking capabilities in Android. The hardware present supports a variety of protocols which in principle offer convenient peer-to-peer capabilities, such as Bluetooth, Bluetooth Low Energy, ordinary Wifi, hotspot mode Wifi, and Wifi Direct. However, the OS limits the ways we can use these technologies. Here we offer several short case studies of these limitations to illustrate the challenges of implementing this type of system on modern mobile platforms.

Ad-hoc WiFi Requires Root.

Ad-hoc wifi, a classic approach for peer-to-peer communication over wifi enabled devices, requires a rooted phone. We opted not to require the user to root their phone for the sake of deployability.

Bluetooth Discoverability Requires User Input.

Bluetooth offers the ability to *discover* other Bluetooth devices, connect to them and exchange messages. In order to be discovered, a device must become *discoverable*, a state in which it broadcasts its presence. For security and user experience reasons, the developers of Android chose to require direct user input any time a device wishes to become discoverable. Additionally, devices can only become discoverable for a short period of time (a minute or two), to conserve battery. This model would prevent Rangzen from operating independently from the user's pocket. While we ended up using Bluetooth for data transfer, we do not use it to discover peers.

Wifi Direct Data Requires User Input.

Wifi Direct is a peer-to-peer protocol using wifi chips which enables devices to discover peers, connect, and exchange data. Unlike Bluetooth, Android's Wifi Direct implementation does not require user intervention for devices to discover one another. However, it does require user input to connect and transfer data with a newly discovered peer.[5] Thus while other devices can be discovered over Wifi Direct, a data connection can't be formed without user input. We eventually settled on using Wifi Direct to find peers, but not to exchange data.

5.2.1 Our Approach

Wifi Direct enables us to discover other devices without user intervention, while Bluetooth allows connections and data transfer without user input. To work around these limitations, we use both protocols in combination to form a full discovery and data path, using Wifi Direct for peer discovery and Bluetooth as a data channel.

[5]The reason for this difference between user input in Wifi Direct and Bluetooth is unclear to us. It may be intentional or accidental. From our perspective as developers, differences like these significantly increased development time, since they are largely undocumented.

Devices searching for peers over Wifi Direct send out beacons to other devices. One of the fields of this beacon is a *name* field, and it is settable in software via a hidden API. We set this name to be the *Bluetooth* MAC address of the local device. Essentially we use the Wifi Direct discovery layer to communicate a small amount of information—the Bluetooth MAC address—which is required to bootstrap a Bluetooth connection. Clients now can bypass the discovery portion of the Bluetooth stack, using the fact that no user interaction is required if the Bluetooth MAC address of the remote device is already known, to connect and exchange messages. This full process can be completed without any user interaction and without the requirement of rooting the device.

This approach has significant limitations: Wifi Direct has significantly greater range than Bluetooth, but our effective range is limited by Bluetooth since we must be in range of both technologies[6]. This illustrates the significant limitations to building this type of system on Android. Even when the raw hardware and protocols are capable of performing the operations necessary to support a system like Rangzen, well-intentioned limitations of the programming environment and OS can require significant hacks to build a working system.

5.3 Engineering Lessons

Building Rangzen was far more difficult than we had expected, especially given the purported support for phone-to-phone communication protocols in modern Android devices. We discuss a few lessons for building these types of systems which are technologically feasible but difficult in practice. in which APIs and OSes are used and built to be used in the common case.

Design Choices for Common Apps Lead to Difficulties for Outliers.

Android's design choices emphasize user experience in the common case. We presume that (legitimate) concerns about apps using Bluetooth and Wifi Direct subsystems maliciously or irresponsibly led to the restrictions we encountered. For example, to limit the battery usage associated with being in Bluetooth's discoverable mode, the system restricts the time that an app can cause the system to remain discoverable and prevents apps from doing so without user intervention.

These limitations don't interfere with most apps in the current ecosystem, which don't involve frequent, opportunistic pairings with many different devices. We note then that our difficulties stemmed primarily from the mismatch between Rangzen's mode of operation and the ways that the majority of smartphone apps behave (typically using infrastructure-based wifi or connecting to a small set of known devices over Bluetooth).

[6]Even still, we found that Rangzen is able to function at ranges up to 40-50m.

Step	Avg	StdDev	Med	90th%
Peer Discovery	1.18	0.70	0.83	2.18
BT Connect Delay	2.30	0.80	2.19	3.29
BT Latency	0.18	0.11	0.18	0.29
Data Tx	1.52	0.18	1.56	1.75
Crypto	0.61	0.01	0.61	0.62
Other	0.82	—	0.81	0.60
Total (measured)	6.61	1.42	6.18	8.37

Table 1: Breakdown of the time spent during a Rangzen exchange between two peers in seconds.

Unfortunately, this mismatch leads to significant difficulty in implementing novel uses of mobile technologies. We suggest that this produces a feedback loop: the difficulty of implementing new types of apps results in a relatively homogeneous ecosystem of apps. Since few use those APIs, platform vendors don't feel the need to support them well. This in turn reduces the likelihood of new developers to use those features.

Working with hidden APIs.

Our reliance on hidden APIs to change the Wifi Direct name of the device to communicate the device's Bluetooth MAC address to peers is one example of the unsustainable practices that derive from a lack of support for the use of available features. In the future, an elimination of this hidden API could break Rangzen. Despite this danger, we were forced to rely upon it for now. We hope to transition to using Bluetooth Low Energy (BTLE), which would come with the benefit that it is designed to have much larger ranges than classic Bluetooth [2]. However, the ability for a phone to act as both sides of a BTLE communication has only been added in Android 5.0, and thus it is likely to be several years before there is widespread support for this mode. We have already experimented with BTLE as an alternative communication mode for Rangzen, and plan to enable it as BTLE support becomes more widespread.

Increased flexibility from rooting/ROMing trades off with fewer users/high deployment costs.

While we rejected rooting or ROMing our phones for this implementation, we acknowledge that a legitimate tradeoff exists between the size of achievable user base and the requirement that users root or reimage their phones. If Rangzen were deployed within a specific, small community, for example, it could be possible to provide a greater degree of privacy and functionality by, for example, deploying a set of pre-imaged, pre-rooted phones.

6 Microbenchmarks

Table 1 depicts our benchmarks of our implementation. The *Total* row represents the total time for an exchange plus the time to locate a peer. Each other row represents a small experiment we did to measure individual factors

that contribute to the full time in an exchange. The *Other* row represents time measured in a full exchange not accounted for by our measurements of individual factors.

6.1 Methods

All network benchmarks were performed at a 10 meter distance between a Nexus 5 and a Nexus 7. We found that even using Bluetooth, Rangzen can operate between devices out to around 40-50 meters, but limited our experiments to 10 meters since we believed that approximated average distances communicating devices were likely to experiment. For each measurement we performed at least 100 trials.

Measuring an entire exchange.

We measured the time from the beginning to end of an exchange between two devices, after they had discovered one another. The devices were preloaded with 100 messages (140 characters each) and 30 friends, which resulted in the transmission of approximately 23.5 KB of data in each direction. In Table 1, we add the peer discovery time to these values to form our "Total" row.

Cryptographic Operations.

We measured the runtime of the computations performed for the PSI protocol over 15 friends. Initialization of the PSI protocol takes 350ms on average. This can be done offline, but we didn't perform this optimization in our prototype. The online portion takes 260ms on average.

Peer Discovery.

We measured the wallclock time between calling the Android API that starts a peer-finding scan until the time our application located a nearby peer.

Delay of Bluetooth Socket Connection.

After peer discovery, the devices involved are aware of each others' presence but must form a Bluetooth RFCOMM connection before transmitting data. We measured the time between requesting such a connection and being informed by the OS that the connection was ready.

Latency of the Bluetooth Data Channel.

We measured the round-trip time between two devices which were already connected over Bluetooth. We sent an integer nonce (4 bytes) back and forth over the channel a single time, counting the time between the measuring node's transmission and its receipt of the echo.

Bandwidth of the Peer-to-Peer Channel.

We measured the achievable bandwidth over a peer-to-peer Bluetooth link. In this experiment we measured the raw bandwidth available to us, which can be viewed primarily as a limiting factor on the number of broadcast datagrams which we can transmit in a single encounter between peers. These speeds were measured over a payload size of 150KB. We measured an median 15.09 KB/s. The 90th percentile worst bandwidth was 13.41 KB/s. In Table 1 above, we converted these bandwidths into the amount of time required to send 23.5 KB at that bandwidth. This corresponds to the amount of data sent in our integrated benchmark for 100 messages and 30 friends.

Battery Usage.

One concern for an application like Rangzen is that it will not be adopted due to detrimental impacts it has on the battery life of devices which run it. We measured the additional battery load imposed on a device, as reported by Android's battery manager. These tests were performed on the Nexus 5 and Nexus 7. We found that on the Nexus 5, Rangzen consumed 5.5% of the device's battery when performing communication with a nearby device every 10 seconds. This represents the worst case for battery usage, since it involves constant communication. Going forward, we intend to benchmark measure power usage more precisely and to to explore power-saving protocols that communicate less frequently to save battery while preserving message propagation (e.g. [70]). However, even at our current prototype's tested numbers, we find that while an additional load of 5.5% per hour will noticeably reduce the battery life of a phone, the load is small enough that we believe it represents a manageable tradeoff.

6.2 Discussion

We note that Bluetooth's slow bandwidth and connection delay account for about 1/3rd each of the duration of an exchange; if we were able to use Wifi Direct (see §5) with its better bandwidth and performance at range, Rangzen's performance would be significantly better. Nevertheless, these numbers combine to form a picture of an app capable of discovering peers and communicating hundreds of messages with them over opportunistic encounters of less than 10 seconds. Even passers-by on a street will often be in range long enough to permit an exchange. We believe our prototype demonstrates that even in the face of the compromises we were forced to make, Rangzen can uphold the promises of its protocol.

7 Simulations and Analytical Results — Rangzen at Scale

The previous microbenchmarks indicate that our Rangzen implementation is robust and fast enough to physically disseminate messages. In this section, we evaluate the anonymity and message-spreading properties of Rangzen at the network level. Ideally, we would have liked to test this using our Rangzen implementation; however, Rangzen's anonymity and reliability depends on large-population statistics. Running small-scale experiments with real subjects would not be indicative of performance at scale, and getting many hundreds of people to participate in an experiment is extremely challeng-

ing. Therefore, we have instead simulated Rangzen operating at city-scale over real mobility traces. We also derive anonymity properties theoretically, and evaluate our expressions for the datasets considered. Due to the lack of public datasets containing social and mobility data, we have used a large-scale mobility dataset and imposed a social graph using known methods.

For simulating message spread, we used real-world datasets, including mobility traces (EPFL Cabspotting [54], St. Andrews Locshare [13], University of Milano PMTR [49], and Technicolor SIGCOMM [53]) and social graphs (two subgraphs of the Facebook social graph [48, 65]). We also tested our algorithms on datasets of mobility *and* social connections [13, 17] (in principle, this is what we want), but we found the datasets to be too sparse in time (Gowalla) and space (St. Andrews) for effective evaluation.

Our simulator consists of 2100 lines of Java code built upon MASON [47], a discrete-event multiagent simulation library. Our simulator accepts social network data inputs or can generate scale-free random social graphs when needed [11]. The simulator supports various mobility datasets. It replays agent locations over time and agents within 20 m[7] are made to encounter one another with some small probability (we chose 0.05). This is meant to simulate unreliable message exchanges for worst-case evaluations.

Nodes can also be "adversarial", which causes them to perform physical/MAC layer attacks. We model these attacks in a worst-case analysis by assuming that all nodes within range of the attacking node are unable to communicate at all. We do not allow honest nodes to upvote messages, to ensure that simulation results are lower bounds on message propagation speed.

7.1 Summary of Results

Our results indicate that Rangzen and Rangzen-like systems could continue to deliver predominantly legitimate messages during an Internet blackout while protecting the anonymity of message authors.

Message propagation. In simulation, Rangzen delivered messages from honest nodes to over 80 percent of the population within 24-48 hours, depending on the prioritization noise parameters (Figure 2). Figure 4 indicates that messages from popular nodes may spread up to 33 percent more than those from adversarial nodes.

Robustness of the network. Figure 5 indicates that Rangzen is robust to localized denial of service attacks (e.g. jamming) when 10 percent of the population is an attacker, using devices with ranges up to 1.3 km. We believe such an attack to be beyond the capabilities of any adversary. However, Figure 4 suggests that coalitions of adversarial nodes cannot dominate network resources as long as they have few friends. A coalition of 6 adversarial nodes in a network of 400 nodes performed only marginally better on average than individual honest nodes selected uniformly at random.

Protection for users.
Authorship deniability. Uses can deny authorship of any message with non-negligible probability (§7.3).
Device Capture. If an adversary captures b's device, b's friend IDs are password-protected. Without input from b, an adversary can only learn mutual friends via the chosen-input PSI-Ca attack. Even with b's password, friend IDs are not stored—only hashes are.
Trust Graph Extraction. A resource-limited adversary cannot learn a significant portion of the trust graph. This effect can be amplified by randomly adding and deleting friends in PSI interactions, and limiting the maximum number of friends that can be fed to the PSI-Ca protocol.

7.2 Message Propagation

Our metrics of success for message propagation are a) the time required for a tagged message to reach 90 percent of the honest population, and b) the fraction of honest nodes that have received a message by a given time. These metrics are chosen for use cases like protest organization, in which mobilization depends on a large portion of the population cooperating. All plots are averaged over 40 runs. We use epidemic propagation over infinite-storage devices as an upper bound on the spread rate since a mesh DTN cannot disseminate content faster than message flooding if storage is unconstrained. Ideally, we would have used a dataset with both social connections and mobility traces, but those datasets were too sparse to effectively capture the population dynamics needed for a mobile mesh network. Since mobility is harder to model than social relations, we used the Cabspotting mobility dataset [54] with a randomly-generated Albert-Baràbasi social graph [11]. The Albert-Baràbasi generative model lets us create arbitrarily-sized social networks, and it displays some common properties of social networks, like high clustering-coefficient, power-law degree distribution, and short path lengths between nodes. However, it does not capture other properties of social graphs, such as community development. Also, true social graphs are typically correlated with mobility patterns.

7.2.1 Propagation without an adversary

Figure 2 shows the propagation of legitimate messages with no adversary. The curves represent different distributions of the noise parameter z_i in our trust metric. Figure 2 suggests that even using random social graphs, **Rangzen can reach at least 80 percent of the population within 24 hours and 90 percent of the population**

[7] Our prototype is able to communicate at ranges up to 40-50m, and sometimes farther.

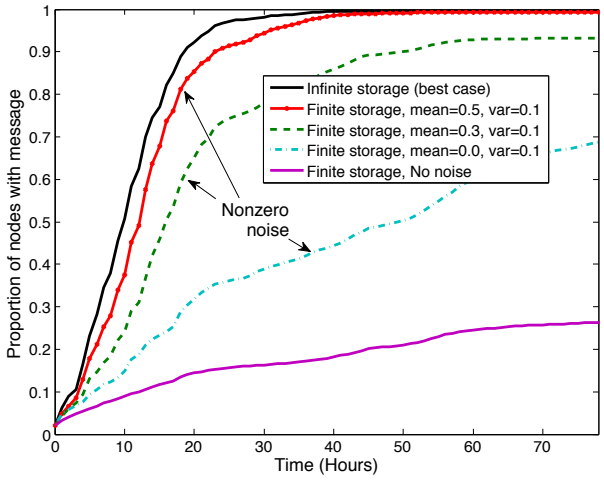


Figure 2: Impact of the Rangzen protocol on legitimate message propagation without an adversary. The additive Gaussian noise in priority scores clearly improves propagation, but may hamper the system's ability to filter out adversarial messages.

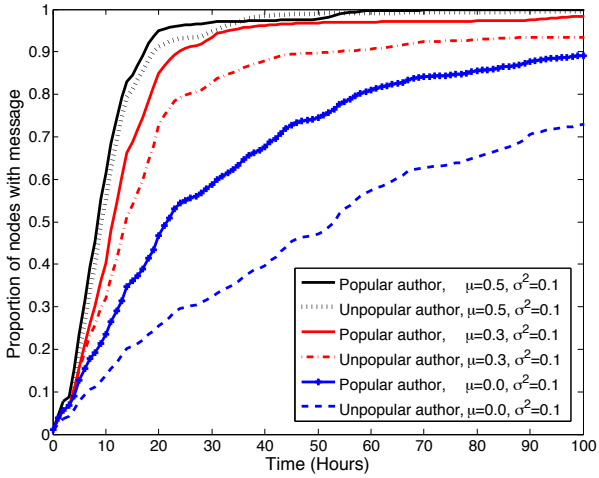


Figure 3: Popular nodes can spread messages faster than unpopular nodes. This effect is more pronounced when nodes add less noise prior to transfers (e.g. lower μ). We expect adversarial nodes to be unpopular.

20 hours after infinite-storage epidemic routing does so. Since taxis move city-wide (unlike most citizens), this experiment is most representative of the time it takes a message to span a city. Again, there is nothing special about taxis—the Cabspotting dataset was just the largest, densest mobility dataset we could find for simulation.

7.2.2 *Propagation with a passive adversary*

Next, we demonstrate the performance of Rangzen under a passive adversary, which deploys devices that follow the Rangzen protocol, but may also disseminate their own content. Distinct groups of friends may wish to emphasize their own content internally without directly attacking others' communications. A node with few connections to a social graph cluster can therefore be considered a passive adversary; its goal is not explicitly to hinder message propagation within the cluster, but messages from more popular nodes in the cluster should be prioritized. This is not a key part of our threat model, but it relates nonetheless to reducing spam in broadcast networks. Figure 3 shows the effects of node popularity on propagation speed. Here, (un)popular nodes were selected randomly from the 5 percent worst- or best-connected nodes in the social network. The figure shows that **messages from popular nodes reach 90 percent of the population as much as 40 hours earlier than messages from unpopular nodes, for certain noise levels**. This model of communication is consistent with natural human communication patterns, which tend to favor people with more social connections.

7.2.3 *Propaganda Message Spread*

We now consider an active adversary that controls a small fraction of nodes. Adversaries have few friends, but they can share friend IDs. The adversarial coalition spreads only its own messages. It can create Sybils, but this is of limited use, since Sybils don't help to better befriend honest nodes. We used noise parameters $\mu = 0.0$ and $\sigma^2 = 0.1$ with a community of 400 nodes, utilizing almost the entire Cabspotting dataset.

Figure 4 illustrates the propagation time of messages originating from popular, average, and unpopular honest nodes, as well as the adversarial coalition; we assume that 1.5 percent of the population belongs to the adversarial coalition.[8] The figure shows that **the adversarial coalition can spread messages a little bit better than average nodes, but at least 30 percent worse than *individual* popular nodes**. This happens because the coalition of malicious nodes continuously broadcasts high-priority propaganda, while an average node has no such group to help with propagation. Thus as long as the adversary has few friends within a coalition, popular nodes can use Rangzen to spread information reliably. At very small scales (50 nodes), we observed that average and unpopular nodes actually performed *better* than the adversarial coalition for the first 48 hours. This suggests that the app can still be used within tighter social circles to communicate, though one should be well-connected to communicate at a large scale.

If an adversary were to corrupt popular nodes it could infiltrate a given social circle. Even offline, this is impossible to prevent. Here we must rely on nodes to gradually unfriend corrupted nodes. Similarly, if the adversary corrupts a significant fraction of the population, Rangzen cannot defend against it. We believe that this will be true

[8] At its height, the Stasi employed 0.6% of the East German population as agents and 1.5% of the population were either employed or in collaboration [41].

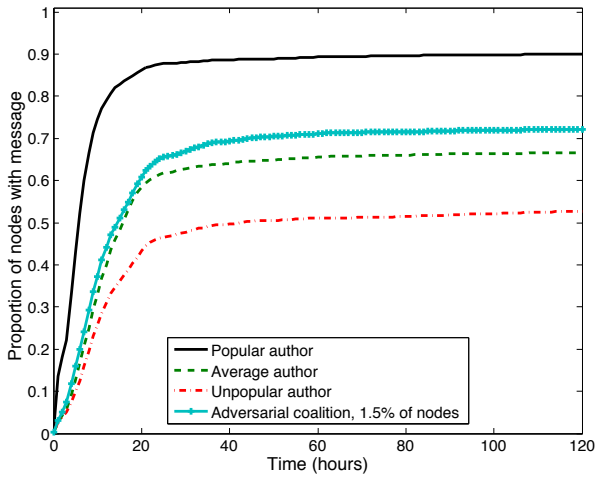


Figure 4: Adversary propaganda spread.

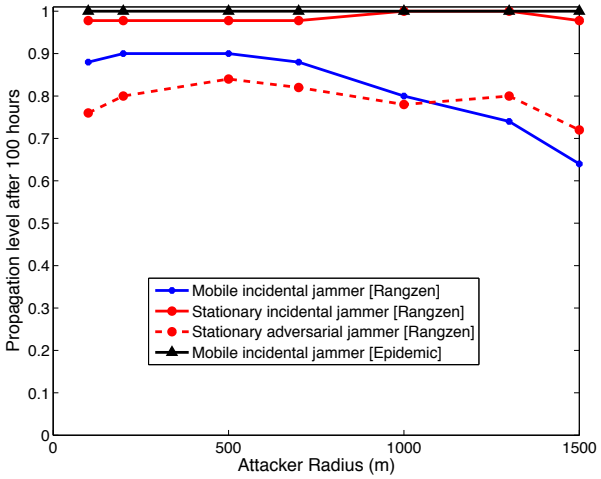


Figure 5: Propagation impact of physical/MAC attacks.

of any decentralized, mobile-mesh-based solution.[9]

7.2.4 *Robustness to physical/MAC attacks*

For a worst-case estimate of physical or MAC-layer attack effects, we first consider a physical-layer attacker (e.g., a jammer). We model it as a point source of radiation in one of the WiFi frequency bands (20 MHz bands at either 2.4 GHz or 5 GHz), as a best-case for the attacker. The received radiation power is assumed to follow the path loss formula: $P_R = P_T \left(\frac{c}{4\pi d f}\right)^2$, where c is the speed of light, f is the signal frequency in Hz, d is the distance traversed, and P_T and P_R are the transmitted and received power, respectively (we assume equal antenna gains). We ignore factors like reflection, diffraction, and absorption, which would significantly weaken a jamming adversary. We estimate the transmit power of a smartphone to be 251 mW (corresponding to average output power over the 5.4 GHz band), and we estimate the maximum output power of a stationary jammer to be 20 W in the same WiFi band (based on commercial jammers). Under these assumptions, a jammer would need to be within 180 m of the receiver with a line-of-sight connection to jam transmissions between nodes 20 m apart. An attacker might extend this range with MAC-layer attacks—e.g., by sending messages that cause other nodes to not transmit. Figure 5 shows the impact of such geography-based attacks on message propagation.

Figure 5 considers mobile and stationary adversaries, both optimally and non-optimally placed. We model mobile, non-optimal adversaries as nodes in the mobility trace. Stationary non-optimal attackers are placed uniformly within the simulation area. A simulated annealing algorithm was used to place optimal, stationary adversaries [40]. An optimal *mobile* attacker would have to know the entire population's location at every instant in time, and solve an NP-hard problem [40], so we don't believe such adversaries pose a greater risk than nodes traversing popular routes regularly, like taxi cabs.

Figure 5 shows that **even when physical/MAC-layer attackers have ranges up to 1000 m, the system propagates at least 80 percent as well as in a best-case scenario**. Such an attack is unlikely, but this highlights Rangzen's robustness to localized attacks.

7.3 Analytical Privacy Guarantees

In this section we evaluate Rangzen's anonymity properties and its resistance to message author identification and social trust graph extraction.

Claim: Sending a high-priority message doesn't necessarily make you look like the author.

If the adversary receives a high-priority message from an honest node, Rangzen should enable the sender to plausibly deny authorship. Here, we derive a distribution for the *anonymity set*, or the set of nodes that could have plausibly authored a particular message. Specifically, we estimate how many hops a message took since inception, and then estimate how many nodes are that many hops away for a fixed confidence level. Recall that random noise is added to message priority scores before each transmission. This noise enlarges the anonymity set.

We compute the pmf of the number of hops a message traversed before reaching a target node, given the observed priority score. Suppose node A receives a message from B. Let N denote the number of hops the message traversed *before* reaching B. $S \in [0,1]$ denotes the priority with which A receives the message from B (before considering their mutual friends). Ω denotes the event that the message is observed by a randomly-selected node (in this case, A). For a worst-case analysis, assume that A receives the message with priority $S = 1$.

[9] Note that 1.5 percent of a city like Damascus, Syria consists of a nontrivial 25,665 spies in a single city.

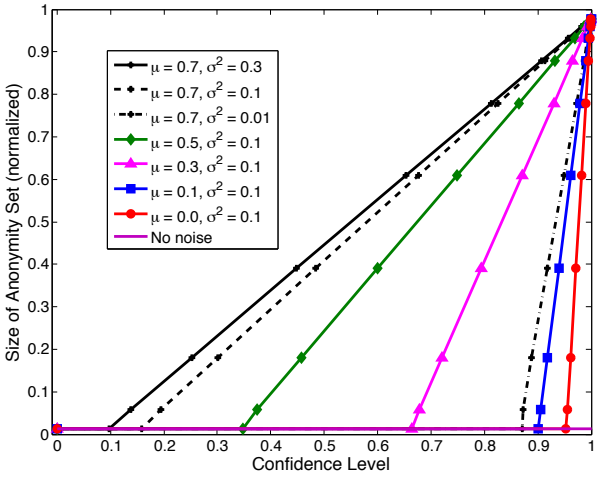


Figure 6: Author anonymity set size (fraction of population) as a function of the estimator's confidence level, given a node stores the message with priority 1. The point $(0.5, 0.2)$ means that the smallest set of nodes to include the author with probability 0.5 contains at least 20 percent of the network.

We want $P(N = n|\Omega, S = 1) = P(S = 1|N = n, \Omega)P(N = n|\Omega)$.

Our modeling of $P(N = n|\Omega)$ and $P(S = 1|\Omega, N = n)$ appears in Appendix A. Using these models, we estimate $P(N = n|\Omega, S = 1)$ as a function of n. Combining this with mobility data, we numerically estimate the anonymity set size for a given trace.

Figure 6 shows the size of the author's anonymity set as a function of the estimator's confidence level—the probability that the true author is in the anonymity set—for the SIGCOMM dataset [53]. Using $\mu = 0.3$ and $\sigma^2 = 0.1$, the 90% confidence anonymity set contains 80 percent of network nodes. More noise significantly increases the anonymity set size. The correct noise parameters should be selected empirically to balance anonymity with message propagation.

Claim: The amount of information the adversary can learn about the social trust graph is limited.

Here we consider an adversary who aims to learn global information about the Rangzen trust graph, such as which pairs of nodes are friends. This information can lead to deanonymization through correlation with other social graphs (e.g., Facebook, Twitter) [51]. In Appendix B, we show analytically that due to Rangzen's node IDs protection, this will be a costly endeavor at scale.

8 Future Work

While we find that Rangzen can enable anonymous messaging in the presence of active attacks, there are a number of avenues for future exploration.

From an engineering perspective, we intend to get the Rangzen codebase professionally audited. On the deployment side, we'll continue talking with groups that may need a system like Rangzen to make versions of our design and implementation that work for them. We also hope to leverage work on battery aware scheduling in delay tolerant networks to improve our power consumption.

We would like to explore the possibilities and tradeoffs involved in using pre-existing social networks or in establishing friendships without physical proximity, such as over the phone. We believe that analogously to key distribution in public key based systems, the bootstrapping of trust in Rangzen is a key feature which may reduce its deployability and usability. Incorporating existing social networks will require thinking about the relatively trustworthiness of online friends and the adversary's ability to make online friends more easily than in person. Additionally, it would require managing levels of uncertainty about users' online friends. However, this work would have the ability to address one of the largest sources of difficulty for deploying systems like Rangzen. Potentially, key distribution services like Keybase could provide a building block [3]. Enabling non-in-person friending requires composing existing authentication approaches to defuse man-in-the-middle risks.

We also hope to further investigate issues we have identified, such as the trade-off between prioritization and anonymity, and Rangzen's robustness to jamming.

9 Conclusion

Since the advent of the Internet and the rise of democratized communication there has been a tension between the communication wants and needs of the many and the prerogatives of the few in control of the means of communication. Our aim has been to evade this tension by designing and building a robust, anonymous communication substrate to evade the shutdown of communications infrastructure. We did this by designing a lightweight, anonymous communications protocol; by implementing that protocol in a prototype Android app which demonstrates the protocol's practical feasibility; and by examining the behavior of the protocol and of our implementation in a series of benchmarks and simulations, showing that Rangzen's design and practice have the potential to deliver on their promises.

How this tension evolves remains to be seen. An arms race naturally follows the use of circumvention technology like Rangzen. Our hope is that Rangzen provides both a useful means of communication that is difficult to shut down and provides sufficient protection to the average user to prevent retribution by an adversarial government. We are currently working with partners on the ground in several medium-risk deployment environments to evaluate Rangzen further and to spur adoption among dissident communities.

References

[1] Anonymizer. https://www.anonymizer.com/.

[2] Bluetooth smart (low energy) technology. https://developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx.

[3] Keybase. https://keybase.io.

[4] Platform versions. https://developer.android.com/about/dashboards/index.html.

[5] Private Internet Access. https://www.privateinternetaccess.com/.

[6] Tavern. https://tavern.com/.

[7] Libya protests: 84 killed in growing unrest, says HRW. *BBC News* (February 19, 2011).

[8] Why is Ukraine in turmoil? *BBC News* (February 22, 2014).

[9] Greece protest against austerity package turns violent. *BBC News* (June 28, 2011).

[10] Chinese Web Censors Struggle With Hong Kong Protest. *New York Times* (September 30, 2014).

[11] ALBERT, R., AND BARABÁSI, A.-L. Statistical mechanics of complex networks. *Reviews of modern physics 74*, 1 (2002), 47.

[12] ASOKAN, N., KOSTIAINEN, K., GINZBOORG, P., OTT, J., AND LUO, C. Applicability of identity-based cryptography for disruption-tolerant networking. In *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking* (2007), ACM, pp. 52–56.

[13] BIGWOOD, G., REHUNATHAN, D., BATEMAN, M., AND BHATTI, S. CRAWDAD data set st_andrews/sassy (v. 2011-06-03). Downloaded from http://crawdad.org/st_andrews/sassy/, June 2011.

[14] BONEH, D., AND HAMBURG, M. Generalized identity based and broadcast encryption schemes. In *Advances in Cryptology-ASIACRYPT 2008*. Springer, 2008, pp. 455–470.

[15] BUCKLEY, C., AND DONADIO, R. Buoyed by Wall St. Protests, Rallies Sweep the Globe. *New York Times* (October 16, 2011).

[16] CHAUM, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 24*, 2 (1981), 84–90.

[17] CHO, E., MYERS, S., AND LESKOVEC, J. Friendship and Mobility: Friendship and Mobility: User Movement in Location-Based Social Networks. In *ACM KDD* (2011).

[18] CHULOV, M. Syria shuts off internet access across the country. *The Guardian* (November 29, 2012).

[19] CLAUSET, A., SHALIZI, C. R., AND NEWMAN, M. E. Power-law distributions in empirical data. *SIAM review 51*, 4 (2009).

[20] CORRIGAN-GIBBS, H., AND FORD, B. Dissent: accountable anonymous group messaging. In *Proceedings of ACM CCS* (2010).

[21] COWIE, J. Egypt Leaves the Internet. *Renesys* (January 2011). http://www.renesys.com/2011/01/egypt-leaves-the-internet/.

[22] DE CRISTOFARO, E., GASTI, P., AND TSUDIK, G. Fast and private computation of cardinality of set intersection and union. In *Cryptology and Network Security*. Springer, 2012, pp. 218–231.

[23] DEHMER, M., AND MOWSHOWITZ, A. A history of graph entropy measures. *Information Sciences 181*, 1 (2011).

[24] DIAZ, C., TRONCOSO, C., AND SERJANTOV, A. On the impact of social network profiling on anonymity. In *Proceedings of PETS* (2008).

[25] DIAZ, J. Iran Shuts Down Google, Will Completely Cut Citizens Off the Internet. *Gizmodo* (September 24, 2012).

[26] DINGLEDINE, R., FREEDMAN, M. J., AND MOLNAR, D. The free haven project: Distributed anonymous storage service. In *Designing Privacy Enhancing Technologies* (2001).

[27] DINGLEDINE, R., MATHEWSON, N., AND SYVERSON, P. Tor: The second-generation onion router. In *Proceedings of USENIX Security* (2004).

[28] DOUCEUR, J. R. The sybil attack. In *Proceedings of IPTPS* (2002).

[29] EL DEFRAWY, K., SOLIS, J., AND TSUDIK, G. Leveraging social contacts for message confidentiality in delay tolerant networks. In *Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International* (2009), vol. 1, IEEE, pp. 271–279.

[30] FAHIM, K. Violent Clashes Mark Protests Against Mubarak's Rule. *New York Times* (January 26, 2011).

[31] FANTZ, A. Son: Iranian dad arrested for my facebook posts. *CNN* (July 12, 2012).

[32] FARRELL, S., AND CAHILL, V. Security considerations in space and delay tolerant networks. In *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on* (2006), IEEE, pp. 8–pp.

[33] FASSIHI, F. Iranian Crackdown Goes Global. *Wall Street Journal* (December 3, 2009).

[34] FREEDMAN, M. J., AND MORRIS, R. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of ACM CCS* (2002).

[35] GARMAN, C., GREEN, M., AND MIERS, I. Decentralized anonymous credentials. In *Proceedings of ISOC NDSS* (2014).

[36] GRAY, D. C., AND CITRON, D. K. The right to quantitative privacy. *Minnesota Law Review 98* (2013).

[37] HEDGES, C. *Death of the liberal class*. Nation Books, 2010.

[38] ISDAL, T., PIATEK, M., KRISHNAMURTHY, A., AND ANDERSON, T. Privacy-preserving P2P data sharing with OneSwarm. In *Proceedings of ACM SIGCOMM* (2010).

[39] JANSEN, R., AND BEVERLY, R. Toward anonymity in delay tolerant networks: threshold pivot scheme. In *IEEE MILCOM* (2010).

[40] KEUNG, G. Y., ZHANG, Q., AND LI, B. The base station placement for delay-constrained information coverage in mobile wireless networks. In *Proceedings of IEEE ICC* (2010).

[41] KOEHLER, J. O. *STASI: The untold story of the East German secret police*. Basic Books, 1999.

[42] KOSSINETS, G., AND WATTS, D. J. Empirical analysis of an evolving social network. *Science 311*, 5757 (2006), 88–90.

[43] LI, M., CAO, N., YU, S., AND LOU, W. Findu: Privacy-preserving personal profile matching in mobile social networks. In *Proceedings of IEEE INFOCOM* (2011).

[44] LIANG, X., LI, X., ZHANG, K., LU, R., LIN, X., AND SHEN, X. Fully anonymous profile matching in mobile social networks. *IEEE JSAC* (2013).

[45] LIBERT, B., PATERSON, K. G., AND QUAGLIA, E. A. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *Public Key Cryptography–PKC 2012*. Springer, 2012, pp. 206–224.

[46] LU, X., HUI, P., TOWSLEY, D., PU, J., AND XIONG, Z. Anti-localization anonymous routing for delay tolerant network. *Computer Networks 54*, 11 (2010).

[47] LUKE, S., CIOFFI-REVILLA, C., PANAIT, L., SULLIVAN, K., AND BALAN, G. Mason: A multiagent simulation environment. *Simulation 81*, 7 (2005).

[48] MCAULEY, J., AND LESKOVEC, J. Learning to discover social circles in ego networks. In *Proceedings of NIPS* (2012).

[49] MERONI, P., GAITO, S., PAGANI, E., AND ROSSI, G. P. CRAWDAD data set unimi/pmtr (v. 2008-12-01). Downloaded from `http://crawdad.org/unimi/pmtr/`, Dec. 2008.

[50] MIERS, I., GARMAN, C., GREEN, M., AND RUBIN, A. D. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Proceedings of IEEE Security and Privacy* (2013).

[51] NARAYANAN, A., AND SHMATIKOV, V. De-anonymizing social networks. In *Proceedings of IEEE Security and Privacy* (2009).

[52] NEEDLEMAN, R. Firechat network-free chat could be big. and now it's on android. *Yahoo News* (April 3, 2014).

[53] PIETILAINEN, A.-K. CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15). Downloaded from `http://crawdad.org/thlab/sigcomm2009/`, July 2012.

[54] PIORKOWSKI, M., SARAFIJANOVIC-DJUKIC, N., AND GROSSGLAUSER, M. CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.org/epfl/mobility/, Feb. 2009.

[55] POST, A., SHAH, V., AND MISLOVE, A. Bazaar: Strengthening user reputations in online marketplaces. In *Proceedings of USENIX/ACM NSDI* (2011).

[56] RASHEVSKY, N. Life, information theory, and topology. *The bulletin of mathematical biophysics 17*, 3 (1955).

[57] REID, F., AND HARRIGAN, M. An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*. 2013.

[58] REITER, M. K., AND RUBIN, A. D. Crowds: Anonymity for web transactions. *ACM TISSEC 1*, 1 (1998).

[59] RHOADS, C., AND FOWLER, G. Egypt Shuts Down Internet, Cellphone Services. *The Wall Street Journal* (January 29, 2011).

[60] SCHNEIER, B. The Battle for Power on the Internet. *The Atlantic* (October 24, 2013).

[61] SHACHTMAN, N. Syria's internet blackout explained. *Wired* (November 30, 2012).

[62] TRIFUNOVIC, S., KURANT, M., HUMMEL, K. A., AND LEGENDRE, F. Preventing spam in opportunistic networks. *Computer Communications 41* (2014), 31–42.

[63] TRIFUNOVIC, S., LEGENDRE, F., AND ANASTASIADES, C. Social trust in opportunistic networks. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010* (2010), IEEE, pp. 1–6.

[64] TUFEKCI, Z. After the Protests. *New York Times* (March 20, 2014).

[65] VISWANATH, B., MISLOVE, A., CHA, M., AND GUMMADI, K. P. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks* (August 2009).

[66] VISWANATH, B., POST, A., GUMMADI, K. P., AND MISLOVE, A. An analysis of social network-based sybil defenses. In *Proceedings of ACM SIGCOMM* (2011).

[67] WALKER, S., AND GRYTSENKO, O. Text messages warn Ukraine protesters they are 'participants in mass riot'. *The Guardian* (January 21, 2014).

[68] WOLINSKY, D. I., SYTA, E., AND FORD, B. Hang with your buddies to resist intersection attacks. In *Proceedings of ACM CCS* (2013).

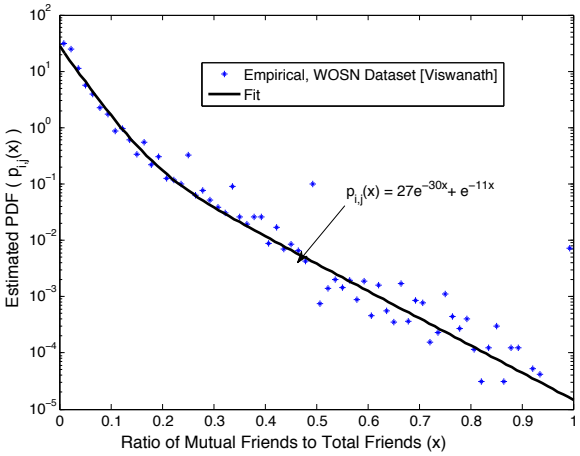


Figure 7: Estimated probability density function of the ratio $p_{i,j}$ over all node pairs. A sum of exponentials over the range $x \in [0,1]$ models the data in [65] well. This model of pairwise trust is used to estimate anonymity set sizes.

[69] WORTH, R., AND FATHI, N. Violent Clashes Mark Protests Against Mubarak's Rule. *New York Times* (June 14, 2009).

[70] YOUNIS, M., YOUSSEF, M., AND ARISHA, K. Energy-aware routing in cluster-based sensor networks. In *Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on* (2002), IEEE, pp. 129–136.

[71] YU, H., KAMINSKY, M., GIBBONS, P. B., AND FLAXMAN, A. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of ACM SIGCOMM* (2006).

[72] ZHANG, X., NEGLIA, G., KUROSE, J., AND TOWSLEY, D. Performance modeling of epidemic routing. *Computer Networks 51*, 10 (2007).

A Anonymity Set Details

A message's priority score S depends on the number of hops the message took. In particular, we can define the received priority after $N = n$ hops (S_n) recursively as follows:

$$\begin{aligned} S_n &= p_n \cdot S_{n-1} + z_n \\ S_0 &= 1 \end{aligned}$$

where z_i is the noise added by the ith node, and p_i is the priority score at the ith node. z_i's distribution is designed, but priority scores p_i depend on graph and mobility properties.

This priority depends on $p_{i,j}$, the scaling factor when messages pass from j to i, as defined in Equation 1. Empirical evidence shows that degree distributions in social networks obey a power law [19]. We found that in the Facebook WOSN dataset [65], mutual degree distributions also obey a power law tail distribution, but the *ratio* p_{ij} across *all node pairs* appears to be better-modeled by

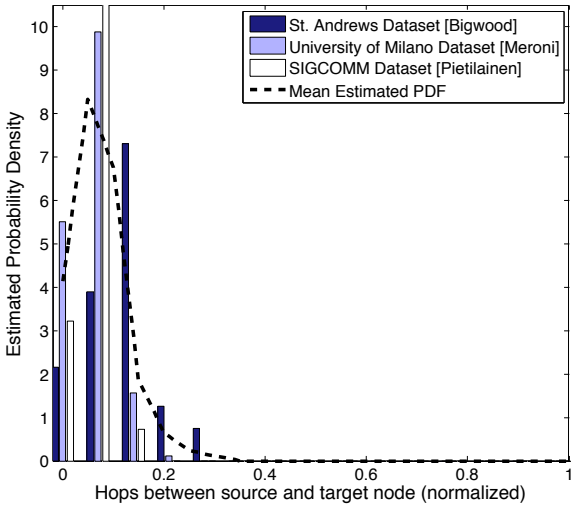


Figure 8: Empirically-estimated PDF for the minimum number of hops between pairs of nodes in mobility traces [13, 49, 53]. Most nodes are within a few hops of one another. We use the mean CDF measurement to model $P(N \leq n|\Omega)$.

a truncated sum of exponentials (Figure 7).[10] This trust metric is not heavy-tailed, so the fraction of nodes with highly overlapping friend sets is vanishingly small. This motivates the sigmoid in equation 1, which assigns high trust even if nodes don't share a large fraction of mutual friends.

We estimated $P(N = n|\Omega)$ empirically from several datasets (Figure 8). For every pair of nodes in the dataset (i, j), we measured the minimum number of times a message would need to be forwarded before reaching target j from source i. This measurement gives an estimated lower bound on how many hops in the (time-varying) connectivity graph separate an arbitrary message from its creator. Figure 8 illustrates these measurements, normalized by the total network nodes. Most pairs are a few hops apart, in part due to the small scale of these mobility datasets.

B Deanonymizing the Social Graph

Our goal is to quantify how much information the adversary learns about the true social graph through attacks on the private set intersection protocol. Throughout, we will assume the adversary knows the vertex set V of the social graph, since use of Rangzen is not assumed to be inherently incriminating.[11] There are many definitions in the literature for graph information content (see [23] for a review). None of the definitions is clearly superior, so we

[10] Technically, this probability is only defined over rational values, but we approximate the function as having a continuous domain.

[11] A dissent network can only be useful if many people participate, which suggests that most network nodes are using it for benign purposes.

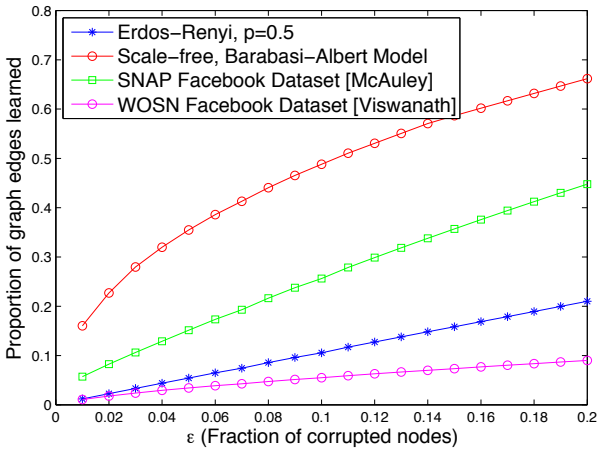


Figure 9: Proportion of graph edges learned by the adversary (d_ε) as a function of ε (proportion of corrupted nodes).

use the proportion of common edges as a heuristic metric. That is, if the original graph is denoted $G = (V, E)$ and the subgraph is denoted $G_s = (V, E_s)$ with $E_s \subseteq E$, then our similarity metric is $d_\varepsilon(G, G_s) = \frac{|E_s|}{|E|}$.

This metric is closely related to the definition of graph entropy by Rashevsky et al [56] and was also shown to be strongly correlated with deanonymization success in [51]. Assuming the adversary can corrupt at most fraction ε of the nodes, we wish to upper bound d_ε as a function of ε. This section demonstrates that the adversary will be unable to learn more than 15 percent of the graph edges by corrupting up to 5 percent of the nodes, and this quantity can be further limited by artificially adding and removing edges from the social graph during PSI-Ca interactions.

Static graph. We start by assuming the trust graph does not change. As time tends to infinity, we assume that the adversary can learn all edges emanating from nodes corrupted by the adversary. This is a worst-case estimate, because it assumes that the adversary knows how to align its learned subgraph within the larger trust graph (or a similar social graph from a different domain). In practice, subgraph alignment is not trivial.

Figure 9 illustrates the proportion of edges learned as a function of the proportion of nodes corrupted. The SNAP dataset is a Facebook ego-social-circle dataset [48], and the WOSN dataset contains social connections between 55,000 nodes in the Facebook New Orleans network as of 2009 [65]. The figure suggests that as long as the adversary cannot corrupt more than about 5 percent of nodes, it can learn at most 15 percent of the social graph. This estimate is worst-case; along with the subgraph alignment issues mentioned earlier, corrupting nodes is difficult, and we expect trust establishment to be less promis-

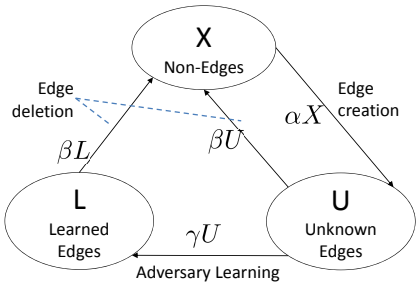


Figure 10: Adversarial learning of a dynamic trust graph.

cuous in Rangzen than in Facebook.

Dynamic graph. Next, we assume that the graph is changing with time. Consider three bins: one with edges that have been learned by the adversary (L), one with edges that have not been learned by the adversary (U), and a final bin containing edges that are not in the graph (X)—i.e., pairs of nodes that are not connected. Each time a new trust relationship is created in this trust subgraph, another edge is added to the U bin, and each time an edge is deleted (i.e. someone "unfriends" an acquaintance) an edge is removed from the L or the U bin. For a worst-case estimate of privacy, we assume the adversary knows when edges are deleted. Edges move from U to L whenever the adversary learns another edge in the graph. Thus we wish to characterize $|L|/|L+U|$. Recall that with a static graph, the adversary could learn at most a small fraction d_ε of the total edges in the graph. As such, our dynamic model operates within a restricted space of nodes and edges. For instance, if the adversary corrupts 5 percent of network nodes, then $N_E = X+L+U$ equals the number of edges possible between the corrupted 5 percent of nodes and the rest of the network. Any equilibrium value of d_ε in our dynamic model should therefore be multiplied by the results for the static graph.

Our underlying model for this system is a continuous-time Markov chain with Poisson events. The state space of this chain grows exponentially in the number of total possible edges (N_E), so we use a mean-field approximation, much like [72]. Figure 10 illustrates our model of the system. αX is the rate of edge creation, $\beta(U+L)$ is the rate of edge deletion, and γU is the rate at which the adversary learns new edges.

We know that $X(t) = N_E - L(t) - U(t)$ where N_E describes the number of total possible edges. Letting $V(t) = [L(t)\ \ U(t)]^T$, we have a nonhomogeneous time-invariant linear system:

$$\frac{dV(t)}{dt} = \begin{bmatrix} -\beta & \gamma \\ -\alpha & -(\alpha+\beta+\gamma) \end{bmatrix} V(t) + \begin{bmatrix} 0 \\ \alpha N_E \end{bmatrix} \quad (2)$$

Observation B.1. *Let $V(t) = [L(t)\ \ U(t)]^T$, with dynamics described in Equation 2. Then* $\lim_{t\to\infty} \frac{L(t)}{L(t)+U(t)} = \frac{\gamma}{\gamma+\beta}$.

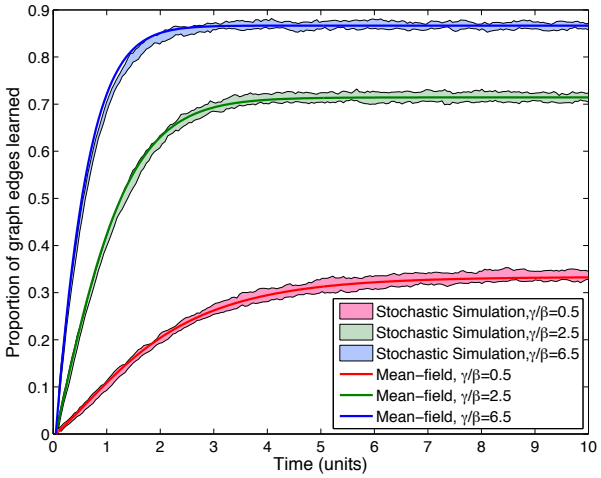


Figure 11: Adversarial graph learning over time, parameterized by the adversary's learning rate γ. Asymptotically, the leaked proportion of graph edges depends exclusively on the adversary's learning rate γ and the network-wide edge deletion rate β. We can increase privacy by systematically omitting trust graph edges.

Proof. (Sketch) It is straightforward to show that dynamical system (2) is internally stable, with exact solution

$$\begin{bmatrix} L(t) \\ U(t) \end{bmatrix} = \begin{bmatrix} \frac{\alpha\gamma N\left(\alpha-\gamma+(\beta+\gamma)e^{-(\alpha+\beta)t}-(\alpha+\beta)e^{-(\beta+\gamma)t}\right)}{(\alpha-\gamma)(\alpha+\beta)(\beta+\gamma)} \\ \frac{\alpha N\left(\beta(\alpha-\gamma)+-\alpha(\beta+\gamma)e^{-(\alpha+\beta)t}\gamma(\alpha+\beta)e^{-(\beta+\gamma)t}\right)}{(\alpha-\gamma)(\alpha+\beta)(\beta+\gamma)} \end{bmatrix} \quad (3)$$

We then consider $\frac{L(t)}{U(t)+L(t)}$. Since the exponential terms in (3) tend asymptotically to 0, the ratio of interest converges precisely to $\gamma/(\gamma+\beta)$. □

Figure 11 illustrates our analytic results compared to simulated results. The colored bands are inter-quartile ranges over 40 trials. These results affirm our mean-field approximation, not the assumption of constant-rate learning and social graph alterations. However, our model does capture the observation that social graph properties stabilize globally over time, despite continuing to change locally [42].

This result says two things: 1) if no edges are deleted, the adversary eventually learns the entire graph, and 2) asymptotic behavior is independent of the edge creation rate. Over a long time scale, we cannot rely on natural social graph growth to limit the adversary's knowledge. Instead, we should artificially simulate the deletion of edges by (for instance) including random subsets of users' friend sets in each private set intersection.