Rangzen: Circumventing Government-Imposed Communication Blackouts

Giulia Fanti
Yahel Ben David
Sebastian Benthall
Eric Brewer
Scott Shenker

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-128
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-128.html

July 6, 2013

Acknowledgement

This material is based on work supported in part by NSF Graduate Research Fellowship DGE-1106400.

Rangzen: Circumventing Government-Imposed Communication Blackouts

Giulia Fanti[1], Yahel Ben-David[1,2], Sebastian Benthall[1], Eric Brewer[1,3] and Scott Shenker[1,4]

[1]Department of Electrical Engineering and Computer Science, University of California, Berkeley
[2]De Novo Group
[3]Google Inc.
[4]International Computer Science Institute

ABSTRACT

A challenging problem in dissent networking is that of circumventing large-scale communication blackouts imposed by oppressive governments. Although prior work has not focused on the need for user anonymity, we contend that it is essential. Without anonymity, governments can use communication networks to track and persecute users. A key challenge for decentralized networks is that of resource allocation and control. Network resources must be shared in a manner that deprioritizes unwanted traffic and abusive users. This task is typically addressed through reputation systems that conflict with anonymity. Our work addresses this paradox: We prioritize resources in a privacy-preserving manner to create an attack-resilient, anonymity-preserving, mobile ad-hoc network. Our prioritization mechanism exploits the properties of a social trust graph to promote messages relayed via trusted nodes. We present Rangzen,[1] a microblogging solution that uses smartphones to opportunistically relay messages among citizens in a delay-tolerant network (DTN) that is independent of government or corporate-controlled infrastructure.

Categories and Subject Descriptors

C.2.0 [**General**]: Security and protection; C.2.1 [**Network Architecture and Design**]: Distributed networks; K.4.1 [**Public Policy Issues**]: [Privacy, use/abuse of Power]

General Terms

Algorithms, Design

Keywords

Dissent networking, anonymity, censorship, opportunistic routing, DTN

1. INTRODUCTION

Recently, tyrants have attempted to subdue political turmoil and civil uprising by imposing large-scale communication blackouts. These blackouts consisted of shutting down Internet access, cellular and wired telephone systems, and at times even the power grid. Our goal is to provide an alternate communication solution that is independent of government- and corporate-controlled infrastructure. We coin the term *dissent networking* to describe communication or networking solutions that facilitate civil dissent.

1.1 Anonymity

Anonymity is a critical property for dissent networking due to fear of persecution; we consider intimidation and punishment of the authors of subversive content as part of our threat model. Studies in this area typically focus on secrecy and authentication [13], properties that conflict with anonymity. Some studies suggest pseudonymity[2] to preserve user privacy [10], but pseudonymity has been shown relatively easy to deanonymize, especially through correlation with external information [19]. This approach has been moderately fruitful in purportedly anonymous systems like the Bitcoin network [21], and the dangers are even greater in a dissent networking context.

1.2 Anonymity-preserving prioritization

A key challenge for any community-owned, decentralized, communication network is that of resource al-

[1] Rangzen is the Tibetan word for freedom or liberty.

[2] Decoupling of real users from their network identities.

location and control. The network's finite resources must be shared among citizens in a manner that mitigates the effects of unwanted traffic and abusive users while allotting higher capacity to desired content. Intuitively, the way to achieve such a prioritization mechanism is by means of community-reputation systems—content from reputable users should receive higher priority. However, such mechanisms require the network to collectively store information about individuals in order to pass judgment, thereby reducing the degree of anonymity within the system.

Balancing the need for anonymity with prevention of network abuse and attack resiliency is the principle design challenge that we tackle in this paper. We see this as the first of many pertinent research challenges on the road to implementing a Rangzen network.

1.3 Risks of exotic hardware

Several systems have addressed similar challenges in the past by proposing hardware-dependent solutions like rooftop antennas or improvised towers [9, 14, 22]. However, given the harsh restrictions that are typically enforced by oppressive governments, there are significant dangers in the setup and operation of solutions relying on alternate infrastructure elements. To avoid these dangers, we maintain that a dissent networking solution should be solely comprised of regular smartphones loaded with an enabling software application. In doing so, Rangzen leverages existing communication capabilities within phones (e.g. WiFi, Bluetooth) while removing further dependencies on infrastructure like the cellular network.

1.4 Fundamentality of DTN

We acknowledge the numerous pilots and vast body of research on mobile-mesh networks—projects that failed to scale beyond the lab. The majority of failures have stemmed from attempting to support Internet-like, online, end-to-end connectivity that conflicts with the store-and-forward communication paradigm. The store-and-forward paradigm, on which peer-to-peer ad-hoc meshes are based, increases resource contention exponentially with every added node, thereby extending latency and limiting scalability. Moreover, it is unlikely that our target localities will be dense enough to provide the desired end-to-end coverage, even if we had a way to avoid contention. High node mobility and churn also detract from the network's ability to establish end-to-end connectivity.

These fundamental challenges in supporting Internet-like connectivity over a mobile mesh led us to focus on a disruption- and delay-tolerant network (DTN) paradigm. Although ill-suited for many Internet applications, it provides a robust packet delivery fabric that is grounded on an extensive body of work, primarily from the sensor network research community.

In a DTN-mesh, phones exchange traffic when they opportunistically come within radio range of each other and collaboratively relay messages on behalf of other members. Such epidemic-like diffusion of content embraces mobility to overcome wide geographic gaps[3] and does not depend on high node density for delivery. The DTN-mesh framework is naturally conducive to broadcasting messages, which motivates our choice to design a microblogging tool.

1.5 Threats and Goals

We assume throughout this paper that the proportion of government agents in the system is very low compared to the number of citizens ($\approx 0.1\%$). We speculate that no system dominated by adversarial agents could reliably protect its legitimate users. Despite this low ratio of agents, Rangzen is designed explicitly to circumvent government censorship, so we anticipate a number of unique threats.

Radio Jamming: Traditionally, the first threat that comes to mind when discussing any sort of wireless communications is of radio jamming. We believe that Rangzen is resilient to such an attack, given the close physical proximity that is required for two mobile phones to establish a direct wireless connection. Generating a jamming signal that is strong enough to overshadow a transmission of a nearby sender is unfeasible at scale. The government may apply powerful and focused jamming at key locations where people congregate, which indeed would disturb and spoil the plentiful opportunistic data exchanges that would otherwise occur in such locations. However, given the mobility of devices, such focused disturbances are likely to be insignificant for the overall Rangzen network and costly for the government.

DoS, Information Poisoning, and Sybils: Another threat is denial of service attacks that flood the messaging system. These may come from an oppressive government or a more mundane attacker like spammers. This attack involves the malicious use of devices that spread nonsense or misleading messages. Malicious devices might be active government agents or artificial Sybil (fake) devices. The government can set up wireless routers in various places as a means of interacting with citizens' devices. These routers may impersonate agents, essentially making them omnipresent throughout the country.

We have designed our message prioritization algorithm in Section 3 with these attacks in mind. As we introduce features and complexity to our design, we will note where we see other possible threats and how our design provides means of resistance.

Goals: Any solution to this problem should exhibit

[3] A smartphone that travels on a bus may link remote locations or even countries.

the following essential properties: It should enable communication with low latency while providing resilience to the aforementioned threats. It should scale gracefully to support hundreds of thousands of nodes. It should function in a distributed, infrastructure-independent, and delay-tolerant fashion without sacrificing anonymity of users or leaking information about whom they trust. Various systems in the literature address subsets of these requirements, but Rangzen is unique in that it addresses all of them. Since stringent anonymity constraints pose the main challenges to our design, our primary technical contribution is an algorithm that prioritizes messages in a privacy-preserving and decentralized manner.

2. DESIGN PRINCIPLES

The microblogging network consists of citizens with smartphones. Whenever two citizens encounter, their phones automatically exchange stored messages, leading to epidemic message distribution. This setup facilitates both decentralization and independence of infrastructure, and microblogging is inherently delay tolerant. Additionally, epidemic routing serves to minimize latency in a DTN setting. Thus we are left to address three remaining goals: anonymity, resistance to flooding/misinformation, and scalability. We handle these requirements by means of a prioritization algorithm, which is the core of Rangzen.

The two limited resources most relevant to our system's performance are the storage capacity of every node (flash memory on each smartphone) and the capacity to exchange messages during opportunistic encounters between devices. Storage considerations depend on the type of messages transmitted (e.g. multimedia vs. text), but we expect the storage on modern smart-phones to suffice for supporting a vast microblogging network.[4] However, the typically short opportunistic encounters between mobile devices may prove problematic,[5] and each device's mobility and battery power are likely to additionally restrict the available bandwidth. Therefore, Rangzen emphasizes the transmission and storage of trusted messages, where trust is determined by our prioritization algorithm. Devices transmit messages with high trust rating first upon establishment of an opportunistic connection. These messages are then most likely to propagate through the network. Similarly, messages with low trust ratings are less likely to get transmitted, except when the duration of the opportunistic encounter is long. Least trusted messages would be deleted from a device's message pool first to make space for incoming trusted messages. The point of prioritization is therefore to assign trust ratings in a reliable yet privacy-preserving way.

2.1 Message prioritization through trust

In the Rangzen network, pairs of users establish trust relationships, which are intended to reflect real trust between the devices' operators. To accomplish this, establishing trust relations should rely on out-of-band verification. For example, users might need to read each others' screens or establish recognition over voice telephony. In this paper, we assume that trust relationships are symmetric, and both parties must confirm trust to establish a link. Borrowing a term from social networking, we will sometimes refer to devices that trust each other as friends.

The network of devices and their trust relations have an implicit graphical structure that we call the *trust graph.* We refer to nodes separated by a path of length ℓ as ℓ-hop friends. Messages can flow between any two network nodes that opportunistically encounter one another, even if the nodes are not single-hop friends; Rangzen therefore depends on inferred, imperfect trust to prioritize message flow. Our objective is to build trustworthiness scores from devices' trust relations in a way that discounts messages generated by agents and/or associated Sybil identities.

We rely on a key assumption that is often exploited in Sybil defense literature [31,32]: We assume that agents have difficulty establishing trust with citizens. This limits the number of links on the trust graph between agents and citizens. These links are often referred to as *attack edges* in the trust graph. Even with arbitrarily many Sybil identities in the network, the number of attack edges is still small compared to the number of citizens, as shown in Figure 1. Our design leverages this limited resource to filter messages from attackers.

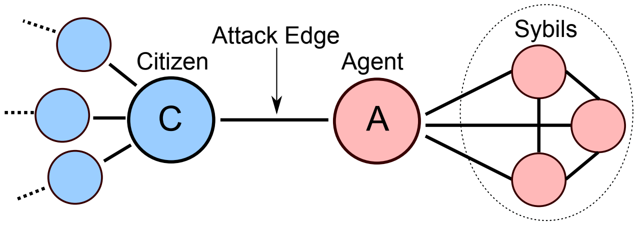

Figure 1: Trust graph structure.

To preserve anonymity, Rangzen eliminates the notion of authorship. Without this information, messages are instead prioritized by the trustworthiness of a traversed route.[6] Because of the limited number of at-

[4] Modern smart phones have 64GB of storage. If we use half of that for Rangzen and expect an average message size to be 1000 Bytes (for comparison, SMS is limited to 190 Bytes), then each node could store 32 million messages, or 320 messages per user in a network of 100,000 nodes.

[5] The lowest bitrate supported by WiFi is 1 Mbps, which means a theoretic transfer rate of 125 messages per second (for 1000 byte messages). Two smart-phones that are relatively static and in proximity may exceed 50 times that bandwidth.

[6] We expect to give users the option of signing and endorsing

tack edges, citizens and agents will have relatively few friends in common. The prioritization algorithm exploits this property by assigning trustworthiness between two nodes proportionally to the number of their shared friends. Intuitively, the more common friends two people have, the more reason they have to trust each other's information. Using friendship to infer trust has been proposed by Trifunovic *et al.* as a defense against Sybil attacks [27]; however, in Rangzen, we extend these ideas to address our strong anonymity requirements.

3. ALGORITHMS

3.1 Core algorithm

This section introduces the fundamentals of Rangzen's prioritization algorithm. We demonstrate how Rangzen securely computes trustworthiness based on the trust graph. Later in this paper, we expand on this simplified algorithm to improve performance.

Each node in the social network maintains a list of its trusted friends (first-hop neighbors on the trust graph) as well as a set of stored messages. Every message m has an associated priority $p_m \in [0, 1]$, which defines the order in which messages are transmitted during opportunistic encounters. Messages with $p_m = 1$ are transmitted first; messages with low p_m are transmitted last and deleted first if the message cache is full. Each message enters the network with rank $p_m = 1$, since authors value their own messages highly.

Two arbitrary nodes, Alice (A) and Bob (B), have associated trustworthiness scores $T(A, B) \in [0, 1]$ indicating how much Alice trusts Bob, and $T(B, A)$ indicating the opposite. This trustworthiness score is a function of the number of mutual friends between them, and it is used to determine the priority of incoming messages. Whenever a message passes from Bob to Alice, she multiplies the incoming message's priority score by the value of $T(A, B)$; thus if she finds Bob untrustworthy, the incoming message will have low priority. The priority of a message is therefore a nonincreasing function of the number (and the trustworthiness) of links it has traversed. In practice, we want to allow a user to manually upvote a message so that popular messages can propagate quickly.

Let $T_1(A, B)$ denote Alice's trust for Bob using only information about single-hop friends, and let $\mathcal{A}_1$ denote the set of Alice's single-hop friends. We define

$$T_1(A, B) = \max\left(\frac{|\mathcal{A}_1 \cap \mathcal{B}_1|}{|\mathcal{A}_1|}, \epsilon\right) \qquad (1)$$

where $|\cdot|$ denotes the size of a set and $\epsilon > 0$ is a threshold that ensures nonzero priority even for untrusted messages. Giving a nonzero value to each trustworthiness link preserves the ordering of incoming untrusted messages instead of collapsing them all to priority zero; this allows users to prioritize even among untrusted messages.

Given our assumptions on the trust graph structure, messages from agents are given low priority. Suppose Bob is an agent with a single attack edge to Charlie (C), and Alice is an arbitrary honest citizen. When messages pass from Bob to Alice, we have

$$T_1(A, B) = \begin{cases} 1/|\mathcal{A}_1| & \text{if } C \in \mathcal{A}_1 \\ \epsilon & \text{if } C \notin \mathcal{A}_1 \end{cases}$$

where the latter case is more probable. This example suggests why messages originating from agents are likely to have their priorities downgraded and malicious messages are unlikely to ever dominate an honest device's message pool.

Note that $T_1(\cdot, \cdot)$ is an asymmetric function; if an attacker were to add Alice as his only friend in the network, then he would trust Alice completely, but she would trust him only a little bit because he is one of many friends. Thus the best way for the attacker to fool Alice is by making as many friends as possible. Out-of-band trust validation makes it difficult to establish trust links in Rangzen, so the agent must either truly befriend nodes or coerce their cooperation.[7]

Because leaking private information is a threat to users' personal security, Alice and Bob must compute trustworthiness without revealing details about whom they trust. To this end, Rangzen employs a form of private set intersection (PSI) that allows parties with distinct information sets to learn the number of common elements without revealing either party's information.

We note the scalability of a network using this algorithm. New nodes enter the network organically by establishing trust with other Rangzen devices. Each device can update its list of friends when trust is established, and the network continues to function as expected.

3.2 Private Set Intersection-Cardinality

Private set intersection-cardinality (PSI-CA) allows two nodes to learn how many friends are held in common without learning which friends are common. It is conducted whenever two devices opportunistically meet; each node's private data set consists of its own friend list.

We assume government agents will misbehave in any way possible to learn information about the other party; PSI-CA cannot provide guarantees against opponents who arbitrarily choose their private friend sets or refuse to participate. In our setup, the adversary (an agent)

messages, but at its core, Rangzen supports purely anonymous communication.

[7] We discuss plausible deniability as a means to address this threat in section 6.

is trying to earn the trust of citizen nodes, thereby disincentivizing non-participation. On the other hand, the agent has an incentive to falsely augment the size of his friend list to appear more trustworthy. It is therefore difficult by design to learn the "friend IDs" of nodes without actually being friends; these IDs are protected by the PSI-CA protocol and by the lack of authorship tracking.

Suppose a node somehow obtains a large set of friends. We institute an upper limit F on the number of friends that can be compared in a single PSI-CA exchange to prevent the node from appearing trustworthy to everyone. This protects against both agents who coerce many trust links and citizens with lax trust standards. Equivalently, we posit that legitimate citizens will have at most F friends for some $F > 0$. If a party submits more than F elements to the comparison, that party is automatically mistrusted. This forces people with excessive numbers of friends to select only a subset thereof for the comparison. Similarly, if a node has few friends, it pads its list with randomly drawn values to obtain a list of length F. The probability of a randomly chosen filler ID coinciding with another node's friend set is low, and fixing F prevents adversaries from learning the size of a friend set during an encounter.

There are several PSI-CA algorithms in the literature, including [5, 7, 16]. To the best of our knowledge, only [16] addresses malicious adversaries, and our approach is similar to their Cardinality-Mal protocol. It relies on concepts such as homomorphic evaluation of polynomials and zero-knowledge proofs, which we will cover briefly.

An *additively homomorphic cryptosystem* has the property that

$$\mathcal{E}(a+b) = \mathcal{E}(a) \cdot \mathcal{E}(b) \tag{2}$$

where $\mathcal{E}(\cdot)$ denotes encryption using said cryptosystem. This key property implies that for any constant c,

$$\mathcal{E}(ca) = \mathcal{E}(a)^c. \tag{3}$$

Additively homomorphic cryptosystems are commonly used in private set intersection algorithms because they facilitate computation in the encrypted domain. In particular, they make it easy to evaluate polynomials in the encrypted domain given the encrypted polynomial coefficients. In this PSI-CA protocol, we will utilize an important example of such a cryptosystem called the Paillier cryptosystem [20].

We use zero-knowledge proofs and privacy-preserving protocols to deal with malicious adversaries. A zero-knowledge proof is a method for proving that a party knows a secret without revealing the secret to the verifying party. Efficient implementations for the Paillier cryptosystem rely on proving knowledge of discrete logarithms [2]. We will utilize two such functions. Proof of plaintext knowledge ($\mathrm{PK}\{v \mid \mathcal{E}(v)\}$) shows that the prover knows the plaintext identity r given that the encryption $\mathcal{E}(v)$ that is visible to the verifier [4]. Proof of correct polynomial evaluation ($\mathrm{PE}\{(v,r) \wedge r \neq 0 \mid \mathcal{E}(r \cdot f(v))\}$) shows that the prover knows the plaintext values v and r and $r \neq 0$, given the encrypted polynomial f and the encrypted evaluation of the polynomial $\mathcal{E}(r \cdot f(v))$ [4, 15].

Protocol Description Suppose Alice and Bob each possesses a set of friend keys, denoted $\mathcal{A} = \{a_1, ..., a_{|\mathcal{A}|}\}$ and $\mathcal{B} = \{b_1, ..., b_{|\mathcal{B}|}\}$ respectively. Let a_i denote the ith element of set $\mathcal{A}$. The algorithm consists of two rounds of PSI-CA. In the first round, Alice learns the number of shared friends, and in the second, Bob does. Since the iterations are identical except with switched roles, we will only explain the case in which Alice is trying to learn the number of common friends. The steps are as follows:

1. Alice performs the following:

   (a) She generates a secret-key/public-key pair for the homomorphic encryption scheme.

   (b) She generates a polynomial $f_{\mathcal{A}}(x)$, with the elements of $\mathcal{A}$ as roots:

   $$\begin{aligned} f_{\mathcal{A}}(x) &= \prod_{k \in \{1,2,\ldots,|\mathcal{A}|\}} (x - a_k) \\ &= \eta_0 + \eta_1 x + \ldots + \eta_F x^F \end{aligned}$$

   The degree of $f_{\mathcal{A}}$ is F because there are exactly F elements in each private set by construction.

   (c) She sends the encryption of each coefficient of $f_{\mathcal{A}}$ (except η_F) to Bob, along with proof of plaintext knowledge ($\mathrm{PK}\{\eta_i \mid \mathcal{E}(\eta_i)\}$) for each coefficient. η_F is always assumed to equal 1.

2. Bob executes the following:

   (a) Using homomorphic encryption properties, he evaluates the polynomial F times: once for each of his entries, giving $\mathcal{E}(f_{\mathcal{A}}(b_i))$.

   (b) For each set element b_i, he multiplies the encrypted evaluation of $f_{\mathcal{A}}(b_i)$ by a distinct, randomly drawn number r_i, giving $\mathcal{E}(r_i \cdot f_{\mathcal{A}}(b_i))$.

   (c) He generates proof of correct polynomial evaluation $\mathrm{PE}\{(b_i, r_i) \wedge r_i \neq 0 \mid \mathcal{E}(r_i \cdot f_{\mathcal{A}}(b_i))\}$.

   (d) He returns the F randomized polynomial evaluations and proofs of correct construction to Alice.

3. Alice decrypts the F polynomial evaluations. The number of zeros is the number of common elements.

If both parties execute this procedure, each will obtain the number of common elements. In 2a, if B_i is a shared friend, the polynomial evaluates to $\mathcal{E}(0)$; otherwise it evaluates to the encryption of some nonzero value. Bob cannot determine if the result is an encryption of zero because he lacks the private key. Since the Paillier cryptosystem is randomized, multiple instances of $\mathcal{E}(0)$ will look different with high probability. Finally, step 2b prevents Alice from learning about Bob's friends. If the argument was initially a zero, it will remain a zero, indicating a mutual friend. Otherwise, the argument is scaled by the random quantity r_i. This scaling prevents Alice from solving the polynomial for $f_A(b_i)$.

The scheme requires three total rounds of communication: one transmission in which Alice sends her encrypted polynomial, one in which Bob returns the evaluated polynomials as well as his own encrypted polynomial, and one final transmissions as Alice returns her evaluations of Bob's polynomial. Each transmission is $O(F)$ in size.

Security and Correctness The proof of security for this scheme is analogous to that of Cardinality-Mal in [16]; it shows that for each participant in this scheme, there exists a participant G in the ideal model such that the views of the participants in the real and ideal models are indistinguishable. However, the danger in this scenario does not stem from the security of the scheme, since the two parties only transmit semantically secure encryptions of their data. Instead, we must ensure that the adversary cannot impact the correctness of the scheme.

The provided protocol protects against two types of misbehavior: Alice encrypting $f_{\mathcal{A}}$ improperly and Bob evaluating Alice's polynomial improperly. Step 1c forces Alice to set $\eta_F = 1$ to prevent her from misrepresenting her polynomial as $f_{\mathcal{A}}(x) = 0$, which has every number as a root. In step 1c, inability to provide such a proof of knowledge corresponds to faking a friend set by using an incorrect polynomial $f_{\mathcal{A}}$, from a previous encounter for instance. Faking a friend set without knowing the underlying IDs will only affect how much agent trusts the citizen, not the other way around; however, it does allow the agent to learn about trust relationships in the network. Bob is forced to evaluate the polynomial properly by providing proof of plaintext knowledge of r_i and b_i, and by ensuring that $r_i \neq 0$, which would always result in an encryption of zero. The zero-knowledge proofs therefore ensure correct execution of the protocol.

3.3 Multi-hop extension

In the basic form of our algorithm, each node only knows the identities of its own trusted friends (first degree neighbors on the trust graph). This high-privacy setting is good for protecting the trust graph, but it also diminishes the receiver's ability to prioritize messages relayed via distant nodes. Such nodes may be agents or honest citizens, but with no knowledge of the trust graph, the receiver cannot make such a distinction.

We address this issue by allowing each node to maintain a 'sketch' of the local trust graph. This sketch manifests itself as a list of all the friends within an ℓ-hop neighborhood on the trust graph, for some $\ell \geq 1$. In this case, trustworthiness becomes a function of multiple hops of friendship.

The number of hops in this neighborhood, ℓ, is a function of both the privacy desired by an individual user as well as the overarching anonymity settings in the deployment environment; a larger neighborhood gives a better estimate of incoming message reliability at the risk of reduced privacy. The size of a local neighborhood should be upper bounded by some hard threshold to preserve a certain minimum level of privacy (see section 3.4 for a discussion on how to decide and set such system-wide parameters).

If we store $\ell > 1$ hops of friends in the trust graph, then the greater the separation between two nodes on the trust graph, the less they should trust one another. Therefore, each node stores a list of IDs for each friend in the local trust graph, and we call this set a *neighborhood*. Concretely, Bob's neighborhood ID for a friend named Alice who is located i hops away corresponds to a cryptographic hash of the tuple (Alice, i). This ID, denoted $k(\text{Alice}, i)$, is generated by Alice, and it cannot be used to identify Alice without access to the cryptographic key she used to generate it. Bob will not store only $k(\text{Alice}, i)$, but also all tuples from i to ℓ, which we will refer to as the *sketch set* for (Alice, i):

$$S(\text{Alice}, i) = \{k(\text{Alice}, i), k(\text{Alice}, i+1), \ldots \\ \ldots, k(\text{Alice}, \ell)\}.$$

Note that every instance of $k(\text{Alice}, i)$ is identical, regardless of who possesses it, so Alice need only generate her full sketch set $S(\text{Alice}, 1)$ once upon joining the network. From a privacy standpoint, sketch sets do not provide complete information about a trust graph neighborhood because the trust relationships between neighbors are lost. Additionally, an attacker viewing a neighborhood set has no way of understanding which IDs belong to the same sketch set unless the node has only a single friend.

Link trustworthiness was previously a function of the number of mutual friends between two entities. Since neighborhoods describe classes of friends defined by separation distance on the trust graph, different classes should be weighted differently. The heuristic we consider is a weighted sum of the proportion of common elements in each class. Instead of computing $T_1(A, B)$ as before, we consider a weighted sum of $T_j(A, B)$, which finds the intersection proportion from the jth ring of friends, for $j \leq \ell$:

$$T_j(A,B) = \max\left(\frac{|\mathcal{A}_j \cap \mathcal{B}_j|}{|\mathcal{A}_j|}, \epsilon\right) \tag{4}$$

where ϵ is the same as in equation 1 and $\mathcal{A}_j$ is the set of Alice's j-hop friends. Therefore, the reliability of an edge from Alice to Bob is determined as

$$T(A,B) = \min\left(\sum_{j=1}^{\ell} \alpha_j \cdot T_j(A,B), 1\right) \tag{5}$$

where α is a system-wide vector of parameters that weights the different levels of separation in the trust graph. Therefore, $\alpha_j \in [0,1]$ and $\sum_j \alpha_j = 1$; also, $\alpha_i > \alpha_j \quad \forall i > j$, since closer friends should count more than distant friends. The two parties will execute ℓ rounds of PSI-CA in total.

This formulation clarifies why each node should store a sketch set for each member of its neighborhood: Not storing the sketch set would result in distant nodes on the graph appearing as attackers. For example, consider the scenario in Figure 2, with $\ell = 2$. Suppose Charlie is a 2nd-hop friend of Alice's and a 1st-hop friend of Bob's. Then if Alice and Bob were only storing $k(\text{Charlie}, 2)$ and $k(\text{Charlie}, 1)$ respectively, Charlie would not show up as a common friend. Therefore, storing the sketch sets of these respective keys allows two nodes to find common elements among their friend sets with some (slightly skewed) perception of separation distance, even if the generating keys are not the same.

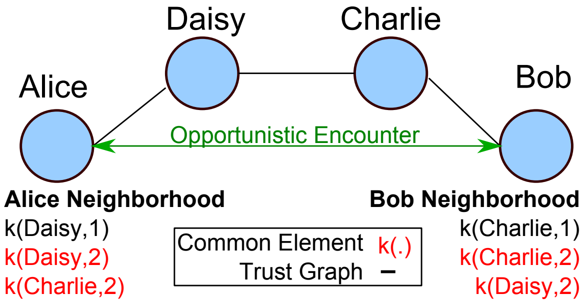

Figure 2: Sample encounter. Alice and Bob meet each other, as indicated by the green edge. After calculating their 2nd degree common friends, they each see that they have two 2nd-hop common neighbors.

3.4 Before Internet Blackout

During this phase, the initial social network of Rangzen is established. While Rangzen is designed to operate in the absence of communication infrastructure, we leverage the pre-blackout phase to speed up the creation of the network and the distribution of the Rangzen software application with the use of a central server. Through Internet access prior to the disconnection event, we overcome the delays introduced by the opportunistic DTN and allow for real-time propagation of messages. The server also selects some preliminary parameters, such as the default neighborhood size ℓ and the multi-hop weight parameters α. We envision the server being located outside the target country, and consequently beyond the control of the malicious government. The government may fully or partially block access to such a server, in which case blocked nodes can simply assume the disconnected phase has begun. Censored access could also be circumvented using traditional overlay networks or proxies.

Users join the Rangzen social network by generating a full sketch set and establishing friendships with people already in the network. Explicitly, for each new user v, the central server distributes appropriate subsets of v's full sketch set to every member of v's neighborhood. For instance, suppose Bob joins the network by becoming friends with Alice. The server will start by transferring Bob's full sketch set to Alice and vice versa. Let $\mathcal{A}_i^j$ denote the set of all Alice's friends that are between i and j hops away, inclusive. Then for each of Alice's friends $f_j \in \mathcal{A}_1^\ell$, if f_j is located $i < \ell$ hops away from Alice on the trust graph, the server will give to Bob the sketch set $S(f_j, i{+}1)$, and to f_j the sketch set $S(\text{Bob}, i{+}1)$. In doing so, the server gives Bob a sketch of his multi-hop neighborhood, and also updates the friend lists of everyone in Bob's neighborhood.

In terms of message dissemination, the central server will handle the entire prioritization pipeline. Having a global view of the trust graph, the server can avoid conducting a private set intersection for every pair of nodes.

3.5 After Internet Blackout

After the Internet blackout occurs, prioritization relies entirely on the private set intersection computations described earlier.

One of the difficult parts of the offline phase is scaling up the network. That is, if a new member joins the network, how does the system update the appropriate nodes' sketches? Due to reduced connectivity, the server can no longer take care of updating all the appropriate neighborhood lists, therefore all neighborhoods must be transmitted from device to device. Suppose that Bob joins the network by becoming friends with Alice. Bob starts by transferring his full sketch set to Alice and vice versa. Then for each of Alice's friends $f_j \in \mathcal{A}_1^{\ell-1}$, Alice will give to Bob the sketch set $S(f_j, i{+}1)$, which she possesses by construction. In doing this, Alice gives Bob his full multi-hop neighborhood (assuming that Alice knows her full neighborhood). The main imbalance in this scenario is that Alice cannot inform her neighborhood of the new addition, because the central server is not accessible and she is presumably not within transmitting distance of everyone in her neighborhood. Thus we wait for an opportunistic encounter. The next time Alice comes into contact with one of her neighbors f_j that is strictly fewer than ℓ hops away, she will transmit an appropriate subset of Bob's sketch set. This effectively informs f_j that Bob is now part of f_j's extended neighborhood. Note that f_j cannot deduce the identity of Bob from the received sketch set; the sketch set received by f_j will never be inserted into messages in any way—it will only be used in the context of the private set intersection protocol, which makes it difficult to correlate hashed keys with real identities.

4. EVALUATION

In evaluating our prioritization algorithm we emphasize two key properties: malicious message infiltration and degree of message diffusion. Obtaining realistic datasets for evaluation of Rangzen is challenging because we need information on human mobility as well as interpersonal trust. Data from typical social networks has little relevance to our scenario of strict trust relations. Similarly, we expect common mobility traces, such as of vehicles or students on a university campus, to poorly represent our use cases. Finally, since trust relations and human mobility are correlated, it is unrealistic to model Rangzen by mapping unrelated social networks and mobility traces. For these reasons, we do not validate Rangzen using real datasets, but instead develop synthetic, conservative datasets to give a worst-case notion of system performance.

4.1 Simulated environment

To synthesize a social graph, we build a small-world network according to the construction by Watts and Strogatz, which relies on adding random edges to circulant graphs [28]. Small-world networks exhibit similar properties to social networks—namely, short average path length between nodes and high local clustering [28]. Our social graph is further augmented by adding a small number of agent nodes and adding edges from each agent node to uniformly-selected-at-random citizen nodes in the graph. We previously assumed the number of agents would be 0.1 percent of the number of citizens; to give a conservative estimate of system performance, we increase this proportion by an order of magnitude in simulation and set the number of agents to be two percent of the number of citizens. An instance of this graph construction with 2^7 citizen nodes is shown in Figure 3; the outer ring of red squares represents agent nodes, while the inner ring of blue circles consists of all the citizen nodes. Edges represent trust relationships. Although this graph size is unrealistic, it serves to demonstrate trends in the system.

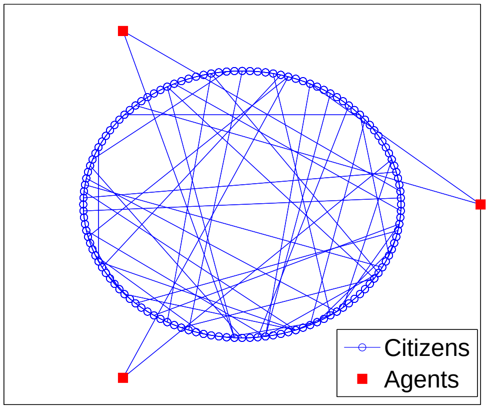

Figure 3: Sample social network graph, with 2^7 citizen nodes, and 3 agent nodes. The inner ring represents citizens and their trust relationships, while the outer nodes represent agents.

With regards to connectivity, we assume that any node in the network will meet some other node in a given time interval with a fixed probability. This encounter rate is higher for agents, as they will attempt to use the infrastructure under their control to impersonate citizens nodes. This Bernoulli random process model of encounters is a discrete approximation of a Poisson arrival process, which is often used to model memoryless random processes like human arrivals [29]. This connectivity model lacks a number of real-life dependencies, including time and location. However, by uniformly pairing nodes for opportunistic encounters, the model disproportionately favors encounters with untrusted nodes; this slows the propagation of messages and gives a conservative estimate of communication performance. We also assume that an agent's cache is always full of agent messages with priority 1. In contrast, each honest citizen will generate a new message at a given timestep with a small probability. This corresponds to authoring a new message or upvoting an existing one, with the end result that the node's cache contains an honest message of priority 1.

4.2 Malicious message infiltration

In measuring malicious message infiltration, we cannot completely eradicate malicious messages from citizens' caches, since the agents' caches are constantly filled with high priority malicious messages. The important notion is that malicious messages should be concentrated at the bottom of a citizen's message pool. Therefore, the first positions of a message pool should be full of honest messages, while the last positions (i.e. the lower priority ones) do not matter. To measure this, we simulated system operation over a number of time steps, and look at the average proportion of honest messages in each cache position. Cache index 1 is the most trusted, so we would like a high proportion of trusted messages at low cache indices. The result of this simulation for the single-hop algorithm at different time iterations is shown in Figure 4. As desired, the lower indices contain more honest messages on average, while the proportion of nodes with honest messages decreases as the priority decreases. Moreover, as time progresses, the proportion of honest messages in citizens' message pools throughout the cache converges to the shape at $t = 80$ iterations in Figure 4, modulo the randomness in the system.

This figure gives a conservative picture, since it only captures the ordering of messages. Due to the prioritization protocol, malicious messages have a lower priority score on average than honest messages; in our simulation after stabilization, malicious messages in citizens' caches had an average reliability of 0.080, while citizens' messages have an average reliability of 0.17; while the latter number may seem small, note that this includes messages coming from completely unknown nodes in the graph, which are mistrusted by the prioritization algorithm as much as messages from malicious nodes.

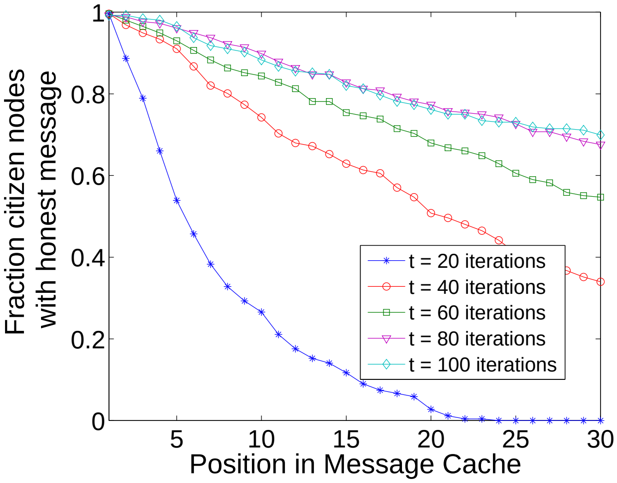

Figure 4: Average proportion of honest messages in citizens' message pools as a function of priority in the pool. Lower indices indicate higher priority. Values are averaged over 2^7 citizen nodes.

The shapes of the equilibrium curves in Figure 4 are dependent on a variety of parameters—one of the most important of these parameters is the agent encounter rate. We assume that agents will transmit messages at a higher rate than ordinary citizens to maximize the number of honest citizens reached. Figure 5 shows the equilibrium curves for various agent encounter rates ranging from 0.1 (same as citizen encounter rate) to 1.0 (constantly exchanging messages). As expected, this shows that as the agent encounter rate grows, the proportion of agent messages in citizens' message pools increases significantly. However, even in the worst case scenario of agents successfully transmitting messages all the time, the first indices in the cache are still primarily occupied by honest nodes.

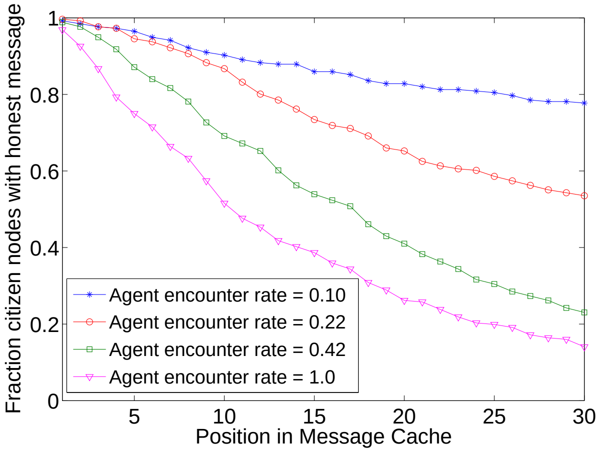

Figure 5: Average proportion of honest messages in citizens' message pools, indexed by priority in the pool. Lower indices indicate higher priority. Values are averaged over 2^7 citizen nodes.

4.3 Message diffusion

An equally important performance aspect is the degree to which messages are able to spread in the system. In practice, this will depend primarily on human mobility patterns, which we do not know. However, we can lower bound message spread using our pessimistic mobility model of uniform encounters. We simulate the diffusion of a single message, assuming no upvotes. We also assume the author posts the message in a highly trafficked area such as a shopping mall to encourage maximum dispersion; this is realistic if an individual wishes to reach many people.

With synthetic data, simulated message diffusion times mean little; however, we can observe the effect of the multi-hop extension on trust levels in the network. In some sense this is more fundamental than observed diffusion times because it is independent of mobility models. The trustworthiness function allows us to upper bound the priority of a received message, which determines how far the message can reach.

In Figure 6, we show the mean priority of the received message as a function of the receiver's distance (in number of hops) from the author. These curves confirm the algorithm intuition that using larger friendship neighborhoods enables greater message spread. Moreover, the parameter vector α, which determines the weight of various friendship levels in the trustworthiness function, allows us to shape the curves in Figure 6 as desired.

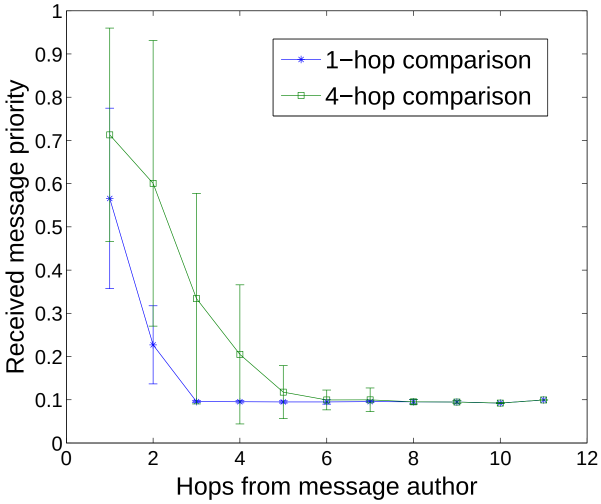

Figure 6: Incoming message priority as a function of separation distance (in hops on the trust graph) from the author.

5. RELATED WORK

Inspired by SumUp [26], GateKeeper [25], Sybilinfer [6] and Sybillimit [32], we base our defenses on the properties of the social network. The number of social links connecting agents to citizens is limited by the number of real friends the agents have—the *attack edges*. Moreover, since the citizens in Rangzen only create social links with trusted friends, the agents cannot create attack edges arbitrarily, resulting in a small number of attack edges compared to the number of citizens. A fundamental departure from these studies is our ability to perform similar defenses without leaking information about the social trust graph, thanks to privacy-preserving set intersections [7, 16].

Although security issues in mobile ad-hoc networks (MANETS) have been studied extensively, anonymity concerns remain relatively unexplored. Exceptions include [3], which quantifies common interests between MANET nodes in a privacy-preserving way, and [18], which emphasizes the anti-localization of authors in a communication network. Despite high-level similarities, [3] focuses on designing primitives for multi-party interest-casting, while we focus instead on inferring trust and resisting network attacks. Regarding [18], we are less concerned about localization since our strong anonymity properties make attribution of messages to authors difficult.

In this study, we address primarily the challenges stemming from our unique threat model; as such, we leave underlying communication foundation issues for future work. Our design is nevertheless shaped by the fundamental constraints of opportunistic and delay-tolerant networking. These considerations were motivated in part by recent work on neighbor discovery in mobile ad-hoc networks [11, 17, 33] as well as canonical DTN literature [12]. Naturally, we are also inspired by recent studies to combat information censorship [1, 23].

6. FUTURE WORK

Implementing a viable Rangzen network is a complex challenge. In this work, we focus on the least explored problem: anonymity-preserving, social graph-based message prioritization. However, there remain many obstacles to be resolved.

User security: While our core algorithm is privacy-preserving, leakage of trust graph information could occur in the multi-hop scenario. In particular, if the government were to forcibly access the devices of many citizens, it could learn each device's neighborhood. Given enough devices in addition to correlation with external information, the government could make inferences about the trust graph structure. Of course, the smaller the neighborhood size (in hops), the more phones required to rebuild the graph. The degree of anonymity reduction in such a scenario needs to be further evaluated beyond our initial exploration in this study.

On a related note, plausible deniability is an important property for our network. We wish to provide a mechanism by which citizens can safely signal to the network that an establishment of trust has been made under duress. Traditionally, this is done by entering a special password different from the user's regular one, indicating that the user is being forced. Preventing agents from detecting the activation of such a hidden mechanism is challenging.

Resource awareness: Since the algorithm is run on mobile devices, it is important to consider resource costs. We anticipate that certain aspects of the algorithm, including neighbor discovery and duplicate message transmission, will prove particularly costly.

During network operation, devices must automatically and efficiently detect the presence of physically close nodes (and communicate with the central server in the pre-blackout phase). Modern smartphones support many modes of communication for doing so, including access to cellular infrastructure, WiFi, Bluetooth, and even physical transportation of memory cards. A multimodal connectivity-seeking networking layer should be designed to gracefully degrade across these modes of connectivity. The Hercules project addresses some of these issues [30], but it focuses on real-time modes while neglecting opportunistic DTN and real-time ad-hoc modes. These have been studied in depth elsewhere but also require heavy alterations for use in Rangzen [11, 17]. We also anticipate popular messages being circulated widely, causing redundant message transmissions. To save bandwidth during opportunistic encounters, nodes need only incrementally replicate messages that are already stored by the receiving node. Studies such as TIERstore [8] and Haggle [24] address these issues in similar environments.

Alternate use cases: We would like to explore uses for Rangzen beyond the specific application presented in this paper. In the microblogging application space, users might wish to distribute multimedia as well as text, which could present additional resource allocation questions. Additionally, for the transmission of confidential messages between friends, we envision the need for message encryption. Public keys could be exchanged during establishment of trust, which could then be used to encrypt messages intended for the corresponding friend; exchanging public keys also implies the ability to sign and authenticate tamper-proof messages. Communication between nodes that do not trust one another is more challenging because it requires key distribution over a DTN mesh without a centrally trusted certificate authority.

More broadly, we wish to address non-dissent use cases. This is important for two reasons: It would provide a cover story for the application, preventing it from being outlawed, and it would encourage the general public to download and use the application. The latter reason is important for studying system functionality at scale. We envision disaster preparedness as such a plausible use case. While the anonymity guarantees and attack resiliency properties of Rangzen might be less important for such scenarios (at least when natural disasters are the concern), the robust, opportunistic distribution qualities over a DTN are highly attractive.

Usability: Care should be taken to ensure Rangzen is user-friendly. It is particularly important for the system to guide users in making informed decisions that may affect their security. Open questions include how to establish trust relations among peers, especially when they are not physically close.

7. CONCLUSIONS

Dissent networking is a relatively unexplored territory that presents extreme challenges. Designing a communication network for citizens in the face of an adversarial government is a major undertaking that can lead to devastating consequences if done poorly. We have presented Rangzen: an anonymity-preserving microblogging tool designed for circumvention of government-imposed communication blackouts and censorship. Our goal was to present a decentralized, delay-tolerant communication system that is both resilient to network attacks and anonymity-preserving for users. We addressed this problem by designing an algorithm that exploits social graph structure and privacy-preserving set intersections to prioritize messages. We have simulated this algorithm on synthetic data and found that on average, it filters out malicious messages so that users see primarily honest messages in the top slots of their message caches.

8. REFERENCES

[1] S. Burnett, N. Feamster, and S. Vempala. Chipping away at censorship firewalls with user-generated content. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 29–29, Berkeley, CA, USA, 2010. USENIX Association.

[2] J. Camenisch and M. Stadler. Proof systems for general statements about discrete logarithms, 1997. Technical report, ETH Zurich.

[3] G. Costantino, F. Martinelli, and P. Santi. Privacy-preserving interest-casting in opportunistic networks. In *Wireless Communications and Networking Conference (WCNC), 2012*, pages 2829–2834. IEEE, 2012.

[4] R. Cramer, I. Damgård, and J. Nielsen. Multiparty computation from threshold homomorphic encryption. *Advances in cryptologyEUROCRYPT 2001*, pages 280–300, 2001.

[5] E. D. Cristofaro, P. Gasti, and G. Tsudik. Fast and private computation of cardinality of set intersection and union. Cryptology ePrint Archive, Report 2011/141, 2011.

[6] G. Danezis and P. Mittal. SybilInfer: Detecting sybil nodes using social networks. In *NDSS*, 2009.

[7] E. De Cristofaro, J. Kim, and G. Tsudik. Linear-complexity private set intersection protocols secure in malicious model. *Advances in Cryptology-ASIACRYPT 2010*, pages 213–231, 2010.

[8] M. Demmer, B. Du, and E. Brewer. Tierstore: a distributed filesystem for challenged networks in developing regions. In *FAST*, volume 8, pages 1–14, 2008.

[9] J. Dibbell. The shadow web. *Scientific American*, 306(3):60–65, 2012.

[10] B. Dodson, I. Vo, T. Purtell, A. Cannon, and M. Lam. Musubi: disintermediated interactive social feeds for mobile devices. In *Proceedings of the 21st international conference on World Wide Web*, pages 211–220. ACM, 2012.

[11] P. Dutta, D. Culler, and S. Shenker. Asynchronous neighbor discovery: Finding needles of connectivity in haystacks of time. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 531 –532, april 2008.

[12] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM.

[13] P. Gardner-Stephen and S. Palaniswamy. Serval mesh software-wifi multi model management. In *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief*, pages 71–77. ACM, 2011.

[14] J. Glanz and J. Markoff. US underwrites internet detour around censors. *The New York Times*, 1, 2011.

[15] M. Jakobsson and A. Juels. Mix and match: Secure function evaluation via ciphertexts. *Advances in CryptologyASIACRYPT 2000*, pages 162–177, 2000.

[16] L. Kissner and D. Song. Private and threshold set-intersection. Technical report, DTIC Document, 2004.

[17] J. A. B. Link, C. Wollgarten, S. Schupp, and K. Wehrle. Perfect difference sets for neighbor discovery: Energy efficient and fair. In *Extremecom*, 2011.

[18] X. Lu, P. Hui, D. Towsley, J. Pu, and Z. Xiong. Anti-localization anonymous routing for delay tolerant network. *Computer Networks*, 54(11):1899–1910, 2010.

[19] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE, 2009.

[20] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in CryptologyEUROCRYPT99*, pages 223–238. Springer, 1999.

[21] F. Reid and M. Harrigan. An analysis of anonymity in the bitcoin system. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 1318–1326. IEEE, 2011.

[22] A. Reynolds, J. King, S. Meinrath, and T. Gideon. The commotion wireless project. In *Proceedings of the 6th ACM workshop on Challenged networks*, pages 1–2. ACM, 2011.

[23] Y. Sovran, A. Libonati, and J. Li. Pass it on: social networks stymie censors. In *Proceedings of the 7th international conference on Peer-to-peer systems*, IPTPS'08, pages 3–3, Berkeley, CA, USA, 2008. USENIX Association.

[24] J. Su, J. Scott, P. Hui, J. Crowcroft, E. De Lara, C. Diot, A. Goel, M. Lim, and E. Upton. Haggle: Seamless networking for mobile applications. *UbiComp 2007: Ubiquitous Computing*, pages 391–408, 2007.

[25] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control. In *The 30th IEEE International Conference on Computer Communications (INFOCOM 2011)*, Shanghai, P.R. China, 4 2011.

[26] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 15–28, Berkeley, CA, USA, 2009. USENIX Association.

[27] S. Trifunovic, F. Legendre, and C. Anastasiades. Social trust in opportunistic networks. In *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, pages 1–6, March.

[28] D. Watts and S. Strogatz. An undirected, unweighted network representing the topology of the western states power grid of the united states. *Nature*, 393:440–442, 1998.

[29] R. Wolff. Poisson arrivals see time averages. *Operations Research*, 30(2):223–231, 1982.

[30] K.-K. Yap, T.-Y. Huang, M. Kobayashi, Y. Yiakoumis, N. McKeown, S. Katti, and G. Parulkar. Making use of all the networks around us: a case study in android. In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, pages 19–24. ACM, 2012.

[31] H. Yu. Sybil defenses via social networks: a tutorial and survey. *SIGACT News*, 42:80–101, October 2011.

[32] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A near-optimal social network defense against Sybil attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17. IEEE Comoputer Society, 2008.

[33] G. Zyba, S. Ioannidis, C. Diot, and G. M. Voelker. Dissemination in opportunistic mobile ad-hoc networks: The power of the crowd. In *The 30th IEEE International Conference on Computer Communications (INFOCOM 2011)*, Shanghai, P.R. China, 4 2011.