Anix: Anonymous Blackout-Resistant Microblogging with Message Endorsing

Sina Kamali
University of Waterloo
sinakamali@uwaterloo.ca

Diogo Barradas
University of Waterloo
diogo.barradas@uwaterloo.ca

***Abstract*—Repressive governments are increasingly resorting to Internet shutdowns to control the flow of information during political unrest. In response, messaging apps built on top of mobile-based mesh networks have emerged as important communication tools for citizens and activists. While different flavors of these apps exist, those featuring microblogging functionalities are attractive for swiftly informing and mobilizing individuals. However, most apps fail to simultaneously uphold user anonymity while providing safe ways for users to build trust in others and the messages flowing through the mesh.**

We introduce Anix, a blackout-resistant app with two novel features: remote trust establishment and anonymous message endorsing. Anix also leverages a set of identity revocation primitives for the fine-grained management of trust relationships and to provide enhanced anonymity. Our evaluation of Anix through comprehensive micro-benchmarks and simulations showcases its practicality and resilience in shutdown scenarios.

1. Introduction

Aiming to control the narrative in times of political unrest, an increasing number of repressive governments are resorting to regional [1] or country-wide [2], [3] Internet shutdowns. These shutdowns, also commonly referred to as "blackouts", deprive citizens of accessing the Internet for a set period and have been observed to be instated from just a few hours up to several weeks in a row [4], [5].

To reclaim their ability to communicate freely during shutdowns, Internet users have resorted to delay-tolerant messaging applications based on mobile mesh networks [6]. These applications, which we refer to as *blackout-resistant communication apps*, convey messages in a "gossip" fashion between end-user equipment devices (often smartphones) through wireless technologies such as Bluetooth and Wi-Fi Direct [7], obviating the need for an active data connection to relay messages. For this reason, these apps have become an attractive tool not just for citizens in general, but also for human rights activists interested in sharing information and organizing urban protests [8]. After their popularization during the 2014 pro-democracy outcries in Hong Kong (through FireChat [9]), blackout-resistant communication apps (e.g., Briar [10], Bridgefy [11]) have been consistently improved, scrutinized [12], [13], and used across multiple countries upon the threats of Internet shutdowns [14], [15], [16].

In the context of activism scenarios, protesters should be able to coordinate amongst themselves and/or expose corruption while avoiding the risk of retaliation by part of the censor. Given the important role of blackout-resistant communication apps in fulfilling these goals, we argue that such apps should strive to meet three core requisites. First, the app should allow users to participate in the network *anonymously*, i.e., a message should not be easily tied to a user's real identity so that users may avoid harassment or prosecution due to dissent. Second, the app should provide a *one-to-many* communication capability to enable the swift sharing of information and facilitate the mobilization of large groups of individuals [17], [18], [19]. Third, since in this communication model any node is allowed to broadcast a message, users should have a way to both gauge the *trustfulness* of the messages they receive and be able to verify their *authenticity*. This would enable users to correctly prioritize messages and safely ignore those introduced by rogue agents; in the past, nation-states have purposely spread misinformation to foil the organization of protests [9], [20] or intimidate citizens [21].

Despite numerous academic and non-academic proposals for blackout-resistant communication apps, their functionalities are limited. After surveying prominent examples (see §2), we find that few of them simultaneously address the requisites mentioned above. For instance, while Moby [22] and ASMesh [23] provide strong anonymity guarantees and enable users to verify the authenticity of messages authored by some trusted contact, they focus on one-to-one communication, making it difficult to disseminate information widely or coordinate large groups. Conversely, some popular apps support one-to-many communication but do not offer anonymity (e.g., Firechat), or fail to offer it despite multiple attempts (e.g., Bridgefy [13]). To date, Rangzen [7] is the only app that supports anonymous one-to-many communication, presenting users with a simple microblogging platform that implements a message prioritization functionality.

Though Rangzen introduces a mutual friendship-based trust scheme for prioritizing messages exchanged within its microblogging platform, this scheme requires users to manually exchange contact information and may fail to reflect most users' practical expectations. As it stands, there is a significant disconnect between a message's trust score

and the perceived usefulness of the message's content or the trust deposited in the message's author itself. Briefly, a Rangzen node prioritizes a message by assigning it a trust score proportional to the number of mutual friends between itself and the node relaying that same message. Unfortunately, this trust scheme may prevent potentially useful messages authored by users outside a close circle of friends from achieving notoriety, potentially impacting the ability of the microblogging platform to raise awareness about specific events. Also, the scheme does not allow a user to gauge how a message is perceived by its own trusted contacts at large (e.g., how many have "upvoted" it).

In this paper, we propose Anix, a blackout-resistant communication app designed to overcome the limitations of previous work by introducing two new key features: a *remote trust establishment* mechanism that allows users to progressively build trust relationships with other users across the mesh network without the need for in-person interaction, and an *anonymous message endorsing* scheme, enabling users to share their opinions on messages with their trusted contacts while keeping their identity hidden from the rest of the network. Additionally, Anix includes *identity revocation* primitives that provide users with fine-grained control over their trust relationships and offer stronger anonymity guarantees, even if users are coerced or their devices are compromised, aligning with other state-of-the-art anonymous mesh networking solutions.

We implemented a prototype of Anix and evaluated its practicality through a set of micro-benchmarks and simulations that aim to reflect its operation in a (down-scaled) city-wide environment. Our results reveal that Anix can efficiently exchange messages and endorsements between devices, while message exchange times still dominate over Anix's lightweight local operations on mobile devices. Our simulations reveal that, even under active attack by a set of adversary nodes that compose 2% of the overall network and refuse to forward messages authored by legitimate users, Anix messages can reach over 90% of users within 23 simulation steps (roughly equivalent to 23 hours).

Contributions. We summarize our contributions as follows:

- We survey prominent blackout-resistant mesh communication apps, highlighting the disparities between their threat models and design features.
- We design Anix, a new blackout-resistant messaging app that enables users to remotely establish and manage trust relationships across the mesh network.
- We build an anonymous message endorsing system that enables Anix users to prioritize messages based on the perceptions of their trusted contacts.
- We develop a prototype of Anix [24] and evaluate it via extensive micro-benchmarks and simulations.

2. Blackout-Resistant Messaging Apps

This section surveys the landscape of existing blackout-resistant messaging apps and categorizes each solution according to four different dimensions tied to their design and operational characteristics. While there is a history of prior blackout-resistant messaging apps [7], [22], [23], [25], [26], most of them differ on a) the communication model they support; b) the anonymity guarantees they provide; c) the way they allow users to trust each other or the messages flowing through the mesh, and; d) the ability for refreshing user identities to recover from different attacks. Throughout our analysis, we contrast the features provided by existing blackout-resistant messaging apps, and make the case for specific design goals we aim to achieve with Anix. We also showcase how Anix fits amongst the existing solutions, supported by a detailed comparison shown in Table 1.

2.1. Communication Models

Messaging apps can generally support three communication models: a) *one-to-one* (O2O), or private messaging, in which two parties exchange messages directly between each other across the mesh network; b) *some-to-some* (S2S), or group messaging, in which two or more users exchange messages in a way that only they can read them; c) *one-to-many* (O2M), or broadcasting, in which a user sends messages to every other user on the mesh at once (akin to publicly posting on microblogging platforms such as X [27] or Reddit [28]). As shown in Table 1, most prior solutions are geared for a one-to-one or some-to-some scheme, in which the mesh network is used as an infrastructure to exchange private messages with other users.

Microblogging capabilities. Rangzen [7] is the single academic proposal for a blackout-resistant app that focuses on offering a one-to-many messaging scheme. Other solutions such as Bridgefy [11] and Firechat [9] support all communication models, but are engineered to provide more comprehensive support for specific models. For instance, Firechat focuses on one-to-many messaging, while Bridgefy focuses on one-to-one and some-to-some communications.

Communication scale. While solutions such as Perry et al. [26] explore domain-specific optimizations for the efficient dissemination of messages between pockets of users who are densely packed within small areas, most blackout-resistant messaging apps aim to deliver messages to large community segments, typically comprising several thousand mobile users throughout a city, with reasonable reliability.

Design goal 1. One-to-many messaging at city-wide scale, while also supporting other communication models.

2.2. Anonymity Guarantees

In the context of mesh-based messaging apps, anonymity can be mapped to three main attributes [23]: a) *sender and receiver anonymity* (SRA), which requires the app to prevent an adversary from linking a given message to its sender and intended receiver(s); b) *forward anonymity* (FA), which requires the app to uphold the sender and receiver anonymity guarantees of older messages if the state of any of the parties engaged in communication is compromised in the future; and; c) *post-compromise anonymity* (PCA), which requires

TABLE 1. COMPARISON OF BLACKOUT-RESISTANT COMMUNICATION APPS. A GREEN CHECK MARK MEANS THE FEATURE IS FULLY PRESENT, AN ORANGE ONE MEANS IT IS SOMEWHAT PRESENT OR CAN BE EASILY ADDED, AND A RED CROSS MEANS IT IS NOT PRESENT.

Application	Communication			Anonymity			Trust System			Revocable IDs	
	O2O	S2S	O2M	SRA	FA	PCA	DT	DTM	TT	SR	HR
Firechat [9]	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Bridgefy [11]	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗
Briar [10]	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗
1am [25]	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗
Moby [22]	✓	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗
Perry et. al. [26]	✓	✓	✗	✓	✗	✗	✓	✓	✗	✗	✗
ASMesh [23]	✓	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗
Rangzen [7]	✓	✗	✓	✓	✓	✗	✓	✗	✓	✗	✗
Anix	✓	✓	✓	✓	✓	✓	✓	✓✓	✓	✓	✓

the app to uphold sender and receiver anonymity for future messages exchanged after a party involved in the protocol is coerced or compromised, although the re-establishment of these guarantees may require a brief recovery period.

FA and PCA are aligned with the notions of forward secrecy and post-compromise security, which normally relate to ensuring message confidentiality [23]. As highlighted in our threat model (§3), however, the typical adversary in an internet shutdown scenario seeks to track the activities of users engaged in mesh messaging without necessarily breaching confidentiality. This means that, for the O2O and S2S communication models where messages are usually encrypted, the adversary will seek to uncover who is communicating with whom. In the O2M model, where messages are sent in the clear, the adversary may instead attempt to uncover the author of a specific message. As a result, we argue that a blackout-resistant app should, at the very least, provide basic sender and receiver anonymity. Some solutions, however, fail to satisfy this requirement.

Anonymity in non-academic solutions. These messaging apps either do not intentionally hide information about a message sender or receiver, or fail to do it successfully. In Firechat [9], users have long-term public IDs that uniquely identify their accounts and are sent along with their messages to every user they share their messages with.

Bridgefy [11] included both the sender's and receiver's long-term identifier on each message. Albrecht et al. [12] reported that this practice was unsafe, compelling Bridgefy's maintainers to enhance the app's security. Yet, the deployed upgrades proved to be insufficient [13] and, in its current state, Bridgefy cannot be considered anonymous.

Briar [10] only allows two users in close physical proximity (e.g., Bluetooth range) to establish a connection if they have previously exchanged contact information, but the app does not attempt to conceal the identities of communication parties [29]. Restrictions on connection establishment may also help a passive adversary uncover a user's social graph.

Anonymity in academic solutions. Mesh messaging apps described in the literature support different anonymity properties to varying degrees. 1am [25] does not provide any anonymity as its messages are exchanged in plaintext and no metadata protections are in place. On the other hand, Rangzen [7] achieves sender and receiver anonymity by explicitly avoiding including the sender and receiver IDs in transmitted messages. Despite its heavy focus on the microblogging scenario, Rangzen users can still find messages specifically targeted at them by actively trying to decrypt encrypted messages flowing through the mesh. (In §2.3, we address how users can exchange cryptographic material to engage in private communication over one-to-many networks.) Rangzen also naturally provides forward anonymity since messages are not linked to their corresponding authors.

Moby [22] achieves the same anonymity guarantees as Rangzen by relying on a simpler version of Signal's [30] double ratchet which enables for sender and receiver anonymity with forward anonymity. ASMesh [23] improves upon Moby by supporting sender and receiver anonymity with both forward and post-compromise anonymity.

Perry et al. [26] seek to ensure sender and receiver anonymity against an adversary that has full visibility over all communications established across the mesh network, by using cover traffic to hide users' messaging patterns. In these conditions, however, highly repressive adversaries may shift focus to outright arrest the individuals found to use a mesh networking app – while this issue is not specific to Perry et al. (and solving it remains an open challenge), it may prompt an adversary to take swifter action in this direction. Further, Perry et al. do not offer safeguards to users who are coerced or whose devices are compromised by the adversary, lacking forward and post-compromise anonymity mechanisms.

Design goal 2. Ensure anonymity despite partial network monitoring, user coercion, or device compromise.

2.3. Trust Systems

Blackout-resistant messaging apps have used different heuristic notions for defining the boundaries of how much a given user can trust messages or other users or messages.

No trust. Some apps avoid defining a trust system altogether (e.g., Firechat [9]). These apps place every single message received by the user at the same level of importance regardless of their originating user, or legitimacy, and as a result, cannot distinguish spam or misinformation from otherwise.

Direct trust. Another group of mesh messaging apps defined trust as simply knowing (with a reasonable certainty) from whom the incoming message is coming from (i.e., a trusted contact). We call these *direct trust* (DT) systems.

Apps using direct trust systems require some notion of identity to be exchanged when any two users wish to establish trust between one another. However, most apps using direct trust systems do not concern themselves with the method through which identities are shared, either mentioning that trust establishment takes place when Internet is available (as in Moby [22]), or that users can exchange keys manually or by scanning a QR code (as in Rangzen [7], ASMesh [23], or Perry et al. [26]). Bridgefy [11] uses a direct trust system in its private and group messaging but has no trust system in place for one-to-many communications.

Direct trust mediators. A third group of apps builds upon the notion of direct trust and improves it by introducing novel ways for helping users establish trust across the mesh network, even when users may not necessarily know each other. We call these *direct trust mediator* (DTM) systems.

Briar and Perry et al. support direct trust mediation by leveraging *introductions*, a scheme in which a common contact between two users can introduce both of them by privately relaying their identifiers to each other across the mesh network. While Perry et al.'s solution and Briar's ultimately translate to direct trust, they present a useful mechanism to remotely establish trust across a mesh.

Note that existing direct trust mechanisms (mediated or not) currently lack the means to remove the trust placed on certain users. This may be a liability when one wrongfully trusts (potentially malicious) users and wishes to "un-trust" them. We address potential solutions for this issue in §4.3.

Transitive trust. Trust systems gain a heightened relevance when applied to the one-to-many communication model since, in a microblogging scenario, distinguishing legitimate messages from spam becomes a significant challenge [7]. In this scenario, it would be useful for users to be able to tell whether (and whom amongst) their trusted contacts has vetted some message as being relevant, even if that message has not been authored by any of the user's trusted contacts. This requires users not only to be able to establish direct trust relationships but also to have a way to perceive how their trusted contacts assess a given message transmitted across the network. We call these *transitive trust* (TT) systems, as they allow users to propagate their positive/negative assessment of a given message to their other contacts.

Rangzen attempts to provide a notion of trust that aligns with – albeit not exactly meets – our definition of transitive trust. To determine the "trustability" of a received message, it privately computes the cardinality of the intersection [31] between the contact list of two users when their devices exchange messages with one another. Briefly, the more contacts two users have in common, the more they trust the messages they receive from one another. This is reflected in a message's *trust score*, which is a direct result of the multiplication of the trust value between two users exchanging the messages and the messages' original trust scores.

We identify several particularities of Rangzen's trust approach that prevent it from meeting our definition of transitive trust. First, the app is afflicted by issues related to *diminishing trust*, i.e., the longer the message has to travel through the network, the lower its trust score becomes, regardless of originating from a trustful source or not. Rangzen offers a way to "repost" a message by resetting its trust score back to its maximum value, but this is unreliable as a user still relies on the message's current trust score to decide whether to repost it. Additionally, reposting feeds into the issue of *path dependency*, i.e., a message's trust score can vary greatly based on the path it took to reach a given user. Thus, in Rangzen, a user has no way to gauge the overall sentiment that her trusted contacts (potentially spread across the mesh at large) placed on a given message.

Design goal 3. A network-wide and path-independent transitive trust system that improves the ability of users to place trust in messages transmitted across the network.

2.4. Revocable Identities

Existing apps tie users' identities to long-lived and static identifiers (e.g., long-term public keys). Unfortunately, this may be detrimental for the anonymity of users and for the authenticity of messages sent across the mesh. For instance, if the adversary can gain the trust of a legitimate user, it may be able to permanently track that user's activities over the mesh network. If a user's device is compromised, the adversary may also acquire cryptographic material that allows it to deanonymize users or impersonate the compromised user.

As mentioned before, direct trust mediation systems may further exacerbate anonymity breaches, as mediator nodes can establish irrevocable trust relationships between two users without their explicit consent. Indeed, Perry et al. [26] mention that should a malicious user be trusted, their solution has no safeguards to prevent de-anonymization.

We argue that, to preserve users' anonymity and ensure the authenticity of messages after a compromise, user identities can be designed to be ephemeral and do not need to be tied to a permanent long-term identity. Based on this assumption, we envision two types of identity revocation primitives, each suited for different scenarios. Next, we briefly describe these primitives and refer the reader to §4.3, where we describe how Anix realizes them in practice.

Soft revocation. A first kind of identity revocation, *soft revocation* (SR), allows a user to remove wrongfully trusted contacts from her trust circle, thus addressing the drawbacks of existing direct trust systems. Soft revocation helps a user revoke her current temporary identity, migrate to a freshly rolled temporary identity, and privately share this new identity with the users she wishes to keep within her trusted circle while excluding distrustful ones.

Hard revocation. An adversary who takes control of a user's device could henceforth impersonate that user. We envision *hard revocation* (HR) as a potential identity revocation primitive that allows a user to signal to her trusted contacts that her account has been compromised and that no further messages authored by this user should be trusted.

Design goal 4. Support identity revocation primitives that enhance anonymity and message authenticity safeguards.

With the above goals in mind, we now present the threat model Anix will operate in before addressing Anix's design.

3. Threat Model

Our threat model assumes a state-level adversary that will force users to switch to Anix for communicating during a blackout. In this setting, the goals of the adversary are threefold: a) to identify the real identity of the users responsible for creating certain public messages exchanged through Anix (e.g., the organizer of a protest); b) to manipulate the way Anix's users perceive the information shared throughout the mesh network (e.g., towards misleading a potential protest's participants with false rally point instructions), and; c) to disrupt communication through the mesh network (e.g., by introducing Sybil nodes that refuse to forward messages, or temporarily jamming wireless signals in certain areas).

Below, we outline our assumptions about the environment where Anix is expected to operate, and the main capabilities and limitations of the adversary we consider.

Assumptions on the adversarial environment. We assume that the adversary is capable of arbitrarily disabling the Internet infrastructure within the country's borders. Further, during an Internet blackout, there is no other safe external network through which users can communicate. This forces users to resort to a mesh networking solution that relies on wireless protocols, such as Bluetooth and Wi-Fi Direct, to connect with nodes within close physical proximity. Through epidemic protocols [32], a user's message will eventually be propagated throughout multiple hops within the mesh network and reach its intended receiver(s).

Like Moby [22], we assume the adversary is only able to operate locally. Thus, while the adversary may intercept or drop messages in certain locations by deploying Sybil nodes in the network [7], the adversary will not be able to globally drop or intercept Anix messages.

Lastly, we assume Anix to be deployed within regions considered to be of medium risk [7], [33] where, though there might be restrictions on free speech, governments do not rely on violence and mass arrests to repress citizens who rely on alternative means to the Internet to communicate. Similarly to Rangzen [7], we consider this to be a reasonable assumption given that solutions such as Anix may either be unnecessary in low-risk scenarios (e.g., within democratic countries whose governments avoid cracking down on protests) or a severe liability in highly repressive countries (e.g., dictatorial regimes), where the simple usage of mesh networking can enact an extreme response from authorities.

Adversary's capabilities. To accomplish its goals, the adversary can set up multiple Sybil nodes within Anix's mesh network and pose as a legitimate user. This allows the adversary to passively intercept messages flowing through the network to, for instance, learn the content of public messages or screen message headers for finding specific identifiers. By playing the role of a Anix node, the adversary can also simply refuse to forward messages. Participating in the network also allows the adversary to fabricate messages and attempt to manipulate the contents of incoming messages before relaying them to other peers in the network. We assume the adversary will also strive to build trust with other Anix users over time and try to influence the actions of benign users by purposely spreading messages carrying misinformation. In addition, we assume that there is the possibility that the adversary will coerce users or seize users' devices to compromise the state held by the Anix app (e.g., message logs or cryptographic material).

Adversary's limitations. The adversary has a limited capacity to convince users to trust its Sybil nodes, resulting in a restricted number of social connections between the adversary and the rest of the network. Thus, the adversary is expected to be less well-connected to other legitimate Anix users within the mesh network. (In §6, we expand on the advantage acquired by an adversary as the ratio of social connections to legitimate users increases.) Furthermore, the adversary is assumed to be computationally bounded and unable to break standard cryptographic primitives.

Attacks outside the scope. Anix does not prevent the adversary from learning that a given individual is using the application. Thus, Anix will not protect users against adversaries that may retaliate simply based on the fact that the app is in use. Anix also does not protect users from targeted attacks where the adversary can single out a legitimate user within a close physical vicinity. In such cases, the targeted user may be limited to directly exchanging newly produced messages with Anix nodes controlled by the adversary, thus forfeiting sender anonymity. Additionally, Anix does not explicitly aim to defend against denial of service attacks where the adversary interrupts communication between nodes in the mesh network (e.g., via jamming) or where nodes selectively refuse to forward certain messages. However, similarly to previous approaches [22], Anix leverages the propagation capabilities of epidemic protocols to deliver messages across the mesh despite local and/or temporary disruptions.

4. Anix

This section discusses the inner workings of Anix, a new blackout-resistant messaging app. We start by outlining Anix's workflow on the *one-to-many* communication model before addressing its core design features and message exchange mechanisms. Then, we detail how Anix can easily accommodate other communication models.

4.1. Overview of the Microblogging Workflow

Anix's microblogging workflow relies on three core operational constructs, which we describe below. Their synergies allow for the creation of an anonymous messaging scheme that allows users to reason about whether they can trust messages originating from potentially unknown users.

1. Pseudonyms and identity management. Central to Anix's operation is the use of one-time-use pseudonyms (PSUs), which partly act as public keys. Every time a user (e.g., Alice) wishes to send a message, she generates

a new pseudonym and places it in the "sender" field of her message. While this ensures sender anonymity for every new Alice message, Anix allows another user to track multiple messages sent by Alice, as long as Alice provides this user the necessary cryptographic material required to link multiple of her pseudonyms together. We expand on the management of pseudonyms in §4.2. Further, given that Alice may wish to prevent certain previously trusted parties from recognizing her messages (e.g., if her device is compromised and used to spread misinformation, or she outright loses trust in certain parties), Anix provides identity revocation primitives that can be used to refresh cryptographic material. We will also cover this aspect in more detail in §4.3.

2. Trust management. We say that *Alice trusts Bob* when Alice sends Bob the cryptographic material that allows him to track Alice's PSUs over various activities. This cryptographic material can be securely propagated to Bob through the mesh network by encrypting it with one of the public keys exposed in any of Bob's PSUs. Bear in mind that during this process, Alice does not necessarily know the identity of Bob. Upon receiving this cryptographic material, Bob can opt to reciprocally trust Alice immediately (e.g., if Alice and Bob are real-life friends who are meeting in person and actively seeking to establish a bi-directional trust relationship in Anix), to trust Alice after tracking her messages over time (e.g., in the case where Bob can only judge whether to trust a user by the content of her messages), or end up deciding not to trust her at all. We elaborate on Anix's trust management primitives in §4.3. Different from existing approaches, this process enables Anix users to progressively build trust in a given user based on the contents of its messages. It also allows for remotely establishing trust during a blackout, without the need to rely on in-person encounters or temporary internet access.

3. Message endorsing and ranking. Anix allows for message endorsing through "votes", as in microblogging platforms such as Reddit [28]. In a nutshell, Anix leverages the concept of *up/downvotes* in a message to attribute an importance score (i.e., rank) to this message. Votes for a given message are propagated across the mesh network when a message is directly exchanged between two nodes, but Anix allows for multiple deliveries of the same message so they can accrue votes over time. Since votes are authored by users' identity keys (§4.2), Alice may not only track the total number of up/downvotes on a message but also track the number of users she trusts who (dis)approve the message. While total up/downvotes could better reflect the overall sentiment about a message being propagated in the network, such votes may be inflated by Sybil nodes. Thus, for messages dealing with sensitive topics, Alice may prefer to reason about how many of her trusted contacts (and/or which of these contacts) upvoted or downvoted the message. We address Anix's message voting mechanisms in §4.4, and point-to-point message exchange in §4.5.

Next, we expand on each of the above three constructs and how they enable us to meet the goals outlined in §2.

Algorithm 1 Generate pseudonym (PSU)

Input: $PubKS_{ID}$, $PrivKS_{ID}$, random r, r'
1: $PubKE_{OTU}$, $PrivKE_{OTU}$ ← generateRandomKeyPair(r)
2: $PubKS_{OTU}$, $PrivKS_{OTU}$ ← generateRandomKeyPair(r')
3: $bsig \leftarrow \texttt{BSign}_{PubKS_{ID},PrivKS_{ID}}(PubKE_{OTU}||PubKS_{OTU})$
4: $PSU \leftarrow PubKE_{OTU}||PubKS_{OTU}||bsig$
5: **return** $PSU, PubKE_{OTU}, PrivKE_{OTU}, PubKS_{OTU}, PrivKS_{OTU}$

4.2. Pseudonyms and Identity Management

Anix nodes rely on a set of cryptographic material to establish their identity in the network and communicate with one another via (potentially anonymous) pseudonyms. This cryptographic material also allows users to identify whether and how their trusted contacts voted on a given message. Specifically, Anix relies on a two-tiered set of *hierarchical keys* to manage user identities within the network.

Cryptographic material and operations. Each user has access to two types of randomly generated asymmetric keys: temporary *identity keys* ($PubK_{ID}$, $PrivK_{ID}$) and *one-time-use keys* ($PubK_{OTU}$, $PrivK_{OTU}$). These keys are instrumental in the creation and management of a user's identity via pseudonyms and, at any given instant, a user may hold multiple identity keys and one-time-use keys. Each of the mentioned key pairs includes separate keys for encryption and signing; for simplifying our exposition, we refer to both the signing and encryption keys using the same notation, making explicit distinctions where appropriate. We use elliptic curve cryptography schemes to generate the above keys (see §5), as Anix operates in a bandwidth-constrained setting where it is desirable to transmit public keys with shorter lengths (e.g., vs. longer RSA keys for equivalent levels of security). Lastly, we refer to the cryptographic operations required by Anix—e.g., encryption (`Enc`), decryption (`Dec`), or signing (`Sign`)—as $\texttt{Operation}_{Key}(Message)$, and use $||$ to denote concatenation.

Pseudonym generation. Every time a user, e.g., Alice, wishes to send a message to the network, she must generate a new one-time-use pseudonym (PSU) which anonymously identifies the sender. Algorithm 1 describes the creation of a pseudonym. First, Alice randomly generates two new one-time-use (OTU) keypairs (lines 1 and 2) used for the encryption ($PubKE_{OTU}, PrivKE_{OTU}$) and signature ($PubKS_{OTU}, PrivKS_{OTU}$) scheme, respectively. Then, Alice generates a signature ($bsig$) of the concatenation of the public components of her one-time-use keypairs ($PubKE_{OTU}||PubKS_{OTU}$) using the (blinded) private component of one of her temporary identities' signing keypair ($PrivKS_{ID}$) (line 3). We describe the process of signing with key blinding (`BSign`) later in this section, as it is fundamental to ensure the unlinkability of pseudonyms. Finally, Alice generates a *pseudonym* (PSU) composed of the concatenation of $PubKE_{OTU}$, $PubKS_{OTU}$, and $bsig$ (line 4). Since each of the three components of a PSU has a fixed size, they can easily be split and interpreted separately.

Ensuring pseudonym unlinkability. Anix's pseudonyms (and the keys contained therein) must adhere to three important properties: a) signatures of one-time-use keys (included in different PSUs) under the same $PrivKS_{ID}$ cannot be

Algorithm 2 Signature with key blinding

Input: Message m, verification/signing keypair $PubKS_{ID}, PrivKS_{ID}$, algorithm [36] for generating a signature with public key blinding ($\texttt{BlSign}_{PrivKS_{ID},bk}(m)$), where bk is the blinding factor.
Output: Message m gets signed with public key blinding

1: $\texttt{func BSign}_{PubKS_{ID},PrivKS_{ID}}(m)$:
2: $bk \leftarrow \texttt{Hash}(m||PubKS_{ID})$
3: **return** $\texttt{BlSign}_{PrivKS_{ID},bk}(m)$

Algorithm 3 Verify a signature generated with key blinding

Input: Signature with key blinding s, message m, verification key $PubKS_{ID}$, algorithm [36] for verification of signatures with public key blinding ($\texttt{Verify}_{bpk}(m, s)$), where bpk is the blinded public verification key.
Output: The signature s of message m is verified

1: $\texttt{func BVer}_{PubKS_{ID}}(m, s)$:
2: $bk \leftarrow \texttt{Hash}(m||PubKS_{ID})$
3: $bpk \leftarrow \texttt{BlPubKey}_{bk}(PubKS_{ID})$
4: **return** $\texttt{Verify}_{bpk}(m, s)$

linked together and traced back to a given user without the knowledge of the corresponding $PubKS_{ID}$; b) messages encrypted with a given $PubKE_{OTU}$ can only be decrypted by the user possessing the corresponding $PrivKE_{OTU}$, and; c) PSUs are unforgeable.

As mentioned above, ensuring pseudonym unlinkability – i.e., that each message generated and signed by different PSUs of the same user cannot be associated together – is an important design consideration for Anix. Unfortunately, common signature schemes (e.g., based on ECDSA [34]) which would directly use $PrivKS_{ID}$ as a signing key – on both PSUs and votes (see §4.4) – are prone to public key recovery attacks [35] which enable an adversary to recover the public identity key associated with a signature, and link all of a user's activities performed under different PSUs.

To address this issue, we explored Denis et al.'s [36] signature scheme with key blinding (which has already been the focus of a formal security analysis by Eaton et al. [37]). Briefly put, this scheme allow us to *blind* (i.e., randomize) the key pair associated with a signature, such that this signature is not linkable to the corresponding public verification key without knowledge of a particular witness. It also ensures unlinkability, in that a verifier cannot distinguish between two signatures based on the same private signing key from two signatures based on distinct signing keys.

Leveraging signatures with public key blinding. Algorithm 2 defines the function `BSign`, used in Anix's pseudonym generation and vote signing procedures. Internally, this function can leverage any public key blinding scheme and, in our current implementation, `BSign` leverages Denis et al. [36] signature scheme with public key blinding (abstracted as `BlSign`). This function takes in a user's identity private signing key ($PrivKS_{ID}$) alongside a blinding factor (bk) and uses them to create a signature that cannot be linked back to the corresponding public verification key $PubKS_{ID}$ without the knowledge of bk. Under [36], a signer generates a different blinding factor bk (line 2) for each new public key blinded signature (line 3).

In turn, Algorithm 3 defines `BVer`, Anix's procedure for verifying a signature generated via `BSign`. Internally, this function leverages Denis et al. [36] signature verification procedure (abstracted as `Verify`), in which a user can verify a public key blinded signature if they have access to the corresponding blinded public verification key (bpk) – i.e., the public component of the key used to sign the message (Algorithm 2) blinded using a blinding factor (bk) under the public key blinding function (`BlPubKey`). While it is typically assumed that either bk or bpk must be communicated to verifiers out-of-band [37], this is unfeasible in our scenario. However, Anix users (both signers and verifiers) can locally generate bk (and, subsequently, bpk) for a message, e.g., by hashing the message to be signed/verified (M) concatenated with its authors's identity signing key public component ($PubKS_{ID}$), i.e., bk = $H(M||PubKS_{ID})$. This ensures Alice's signatures (included in her PSUs and votes) become linkable only to other Anix users who have access to Alice's $PubKS_{ID}$ (i.e., users who are trusted by Alice). In the case of PSU generation, bk = $H(PubKE_{OTU}||PubKS_{OTU}||PubKS_{ID})$.

Tracking message authorship and compartmentalization. The choice of identity keys provides a user with the ability to control the groups of nodes that can track the messages she sends to the network, without necessarily having her disclose her real identity. For instance, consider that Alice creates two different identities, ID_1 and ID_2, with public components ($PubKE_{ID_1}$, $PubKS_{ID_1}$) and ($PubKE_{ID_2}$, $PubKS_{ID_2}$). Alice could then choose to exchange $PubKS_{ID_1}$ in person with some trusted close contacts, e.g., to enable friends and family to track the set of messages whose authoring PSUs are based on ID_1. Further, upon sharing $PubKS_{ID_2}$ to place trust in other (anonymous) users in the network, Alice enables these users to track a different set of messages whose authoring PSUs are tied to ID_2. Thus, Alice can compartmentalize the groups of users she wishes to be able to link different messages, while still being able to build anonymous trust relationships within the network. In case Alice wishes to prevent a previously trusted contact from tracking her new messages, Anix provides mechanisms for revoking identities. We cover these in the next section, after discussing how Anix manages trust between anonymous users.

4.3. Trust Management

Like in previous work, we expect users to bootstrap their list of trusted contacts by exchanging identity keys in person. However, Anix introduces a new direct trust primitive – denoted as *one-way trust* – that enables users to remotely establish trust between each other across the mesh network. In addition, Anix introduces a novel direct trust mediation mechanism – in the form of *referrals* – with significant security enhancements when compared to the one available in previous apps. Lastly, Anix provides a set of identity revocation primitives that better protects users' anonymity.

The notion of one-way trust. Anix offers a way for Alice to place her trust in another anonymous user within the mesh, even if they have never met before. In contrast to popular social media platforms which typically enable Alice to decide to "follow" some other user whose posts seem interesting

Algorithm 4 Establish a one-way trust relationship (A→B)

Input: Public component of an identity key-pair of user A ($PubK_{ID_A}$) which contains both the public encryption and verification keys,
Encryption keys of user B's PSU ($PubKE_{OTU_B}, PrivKE_{OTU_B}$),
Output: User B securely receives user A's identity key

1: `func OWT(receiverOTU, senderID):`
2: `  OWTMessage` ← $\text{Enc}_{\text{receiverOTU}}$`(senderID)`
3: `  sendMessage(B, OWTMessage)`
4: `OWT(`$PubKE_{OTU_B}, PubK_{ID_A}$`)`

5: $PubK_{ID_A} \leftarrow \text{Dec}_{PrivKE_{OTU_B}}$`(OWTMessage)`

to her, Anix provides the concept of "allowed to follow". We call this construct a *one-way trust* (OWT), which can eventually result in a bi-directional trust relationship once two users are "allowed to follow" each other.

One-way trust establishment. OWT works as follows. Assuming Alice finds a message which is useful to her or that is endorsed by some of her trusted contacts, she can grant that message's author the right to track her messages and hope to be "allowed to follow" back. Alice grants this right by sending one of her own identity keys to the owner of the pseudonym that authored the message of interest. Apart from simply judging the content of a message before depositing her trust in the Anix user that authored it, Alice can also reason about whether (and whom amongst) her trusted contacts have upvoted the contents of this message.

To illustrate a unidirectional trust establishment leveraging OWT, consider the following example aided by the description of OWT's steps in Algorithm 4. Assume that Alice decides to trust a message's author via its pseudonym (that we somehow know is from Bob), after verifying that the content of the message is useful and that many of her trusted contacts have upvoted it. Alice proceeds to encrypt one of her identity (e.g., ID_A) keys' public encryption and signing components (jointly represented as $PubK_{ID_A}$) with the $PubKE_{OTU_B}$ key found in Bob's pseudonym (line 2) and sends this ciphertext over the mesh network (line 3), such that only Bob can open this message and access Alice's $PubK_{ID_A}$ (line 5). (In other words, OWT messages are encrypted using a receiver's one-time-use encryption key included in one of their PSUs, thus being indistinguishable from other O2O/S2S messages; see §4.5 – Alternative communication models.) As a result, Bob has now been "allowed to follow" the messages that Alice sends under any pseudonym generated based on her identity ID_A.

Bi-directional trust establishment. Eventually, Bob can decide to trust Alice back, e.g., after receiving useful messages authored by Alice's pseudonyms generated under (ID_A), or verify that his trusted contacts also tend to upvote messages authored by the same pseudonyms. To trust Alice, Bob simply follows an OWT establishment towards Alice. Once she receives this message, Alice can verify that Bob has allowed her to follow him back, thus establishing bi-directional trust.

Until Bob decides to allow Alice to follow him back, Alice's $PubK_{ID_A}$ identity key will sit on a *validation list*, i.e., a separate list for user identities who have one-way trusted Bob, but whom Bob has not yet decided to trust back. As we shall see in §4.4, identity keys in the validation list will not be used for the calculation of message trust scores.

Algorithm 5 Refer two users (A and B) to one another

Input: A PSU of user A (PSU_A) and the corresponding one-time-use public encryption key of user A ($PubKE_{OTU_A}$)
A PSU of user B (PSU_B) and the corresponding one-time-use public encryption key of user B ($PubKE_{OTU_B}$)
Output: Users A and B securely establish bi-directional trust

1: `referralMessageToA` ← $\text{Enc}_{PubKE_{OTU_A}}(PSU_B)$
2: `referralMessageToB` ← $\text{Enc}_{PubKE_{OTU_B}}(PSU_A)$
3: `sendMessage(A, referralMessageToA)`
4: `sendMessage(B, referralMessageToB)`

5: $PSU_B \leftarrow \text{Dec}_{PrivKE_{OTU_A}}$`(referralMessageToA)`
6: $PubKE_{OTU_B}||PubKS_{OTU_B}||bsig_B \leftarrow PSU_B$
7: `OWT(`$PubKE_{OTU_B}, PubK_{ID_A}$`)`

8: $PSU_A \leftarrow \text{Dec}_{PrivKE_{OTU_B}}$`(referralMessageToB)`
9: $PubKE_{OTU_A}||PubKS_{OTU_A}||bsig_A \leftarrow PSU_A$
10: `OWT(`$PubKE_{OTU_A}, PubK_{ID_B}$`)`

Should Bob choose not to follow Alice back, the presented scheme would allow Bob to track Alice's activities throughout the mesh network indefinitely. To limit her exposure, Alice can set a timer for a period she considers acceptable for having Bob establish bi-directional trust (see §4.5). If this period elapses and Alice has not received an OWT message from Bob, she can remove the trust she deposited in Bob by revoking her temporary identity. We elaborate on trust revocations later in this section.

Trust via referrals. Anix supports direct trust mediation (see §2.3) via *referrals*, allowing a user to refer two of their trusted contacts to one another. Anix's referrals are reminiscent of Briar and Perry et al.'s introduction scheme, albeit with a significant distinction. In these previous works, introductions would enable for the establishment of direct trust between two users that had not met before, without their explicit consent; the mediator node would simply reveal each user's long-term identities to one another. This solution could prove disruptive in cases where the mediator misjudges the trustfulness of one party (e.g., introducing a legitimate user to an adversary's agent), or in cases where one of the legitimate users would simply like to avoid other unvetted users accessing their identity.

Anix leverages PSUs to improve upon existing introduction schemes, requiring Alice and Bob's active consent to trust one another, should Charlie refer them. As outlined in Algorithm 5, a referral unfolds by first having Charlie send a PSU of Alice to Bob (and vice versa) to *refer* them to each other (lines 1–4). Then, Alice and Bob can each independently choose to leverage Anix's one-way trust mechanism to establish bi-directional trust (lines 5–10). As previously discussed in this section, if only one of the parties (e.g., Alice) feels comfortable issuing an OWT message, she may later revoke the one-way trust placed in Bob, aborting the unfinished bi-directional trust establishment.

Revocable identities. Anix allows its users to *revoke* their identities to help them preserve their anonymity and signal to other users in the mesh that their cryptographic material may have been compromised. To this effect, Anix supports the two identity revocation primitives described below.

Soft revocation. This form of identity revocation allows Alice to selectively remove the trust she placed on a particular

Algorithm 6 Soft identity revocation

Input: Old identity's encryption ($PubKE_{ID_{old}}, PrivKE_{ID_{old}}$) and signing ($PubKS_{ID_{old}}, PrivKS_{ID_{old}}$) keys, trusted users list (u), random r, r'
Output: New list of trusted users u'

$PubKE_{ID_{new}}, PrivKE_{ID_{new}} \leftarrow$ generateRandomKeyPair(r)
$PubKS_{ID_{new}}, PrivKS_{ID_{new}} \leftarrow$ generateRandomKeyPair(r')
u' $\leftarrow$ `new contact list`
`for` u_i `in` u:
  `if (`u_i `is still trusted):`
    $sig \leftarrow \texttt{Sign}_{PrivKS_{ID_{old}}}(PubKE_{ID_{new}}||PubKS_{ID_{new}})$
    $srm \leftarrow \texttt{Enc}_{PubKE_{ID_{u_i}}}(PubKE_{ID_{new}}||PubKS_{ID_{new}}||sig)$
    `sendMessage(`u_i`, `srm`)`
    u'`.append(`u_i`)`
`Del(`$PubKE_{ID_{old}}, PrivKE_{ID_{old}}, PrivKS_{ID_{old}}, PrivKS_{ID_{old}}$`)`

$PubKE_{ID_{new}}, PubKS_{ID_{new}} \leftarrow$ `DecryptAndVerify(`srm`)`
$h \leftarrow$ `Hash(`$PubKS_{ID_{old}}$`)`
`addEntryToRevokedKeys(`$h, PubKS_{ID_{old}}, PubKS_{ID_{new}}$`)`
`Del(`$PubKE_{ID_{old}}, PubKS_{ID_{old}}$`)`

user, e.g., Bob (or a set of users). Soft revocation relies on Alice's ability to update the compartments of users that can track her messages in the mesh network (see §4.2). Hence, to revoke the trust deposited in Bob, Alice must dispose of the identity key she previously forwarded to Bob via OWT, and rebuild a compartment that includes all users with access to this key, except for Bob. In practice, Alice generates a new identity key pair, encrypts both the identity key she wishes to revoke and her new public identity key with the public identity key of the contacts she is currently "allowed to follow", and sends this message to them privately on the mesh network. As a result, the soft revocation process allows trustworthy contacts who were tracking Alice's messages to continue doing so once she stops using the identity key also shared with Bob. The steps for Anix's identity revocation mechanism are detailed in Algorithm 6.

When Alice's contacts receive the soft revocation message, they hash her old public identity's signing key and store this hash before deleting the key itself. This hash is added to a linked list that contains the hashes of all previous identity keys that Alice may have used (and revoked) since she deposited her trust in that user. This list of expired identity keys can later allow a user to stop trusting messages authored by Alice's PSUs if her device is compromised.

Lastly, we note that a seized device may expose the identity keys of a user's trusted contacts, enabling an adversary to link all activities tied to these identity keys, before affected users may be warned and issue soft revocations for those identities. However, Anix users can generate different identity keys to engage with different parties (see §4.2) which helps mitigate this issue.

Hard revocation. In the case where Alice's device is seized or compromised by an adversary, one must assume the adversary has gained complete access to Alice's cryptographic material. This would allow the adversary to author messages on behalf of Alice and potentially leverage the trust that other users deposited in Alice to spread misinformation or manipulate their behavior. In such cases, Alice can use hard revocation as a last resort, compelling all users not to trust any activity from the revoked identity or any subsequent identities generated via "soft revocation". Hard revocation requires Alice to store her current identity key pairs on some storage medium considered secure and not under the control of the adversary (e.g., stored in a flash drive Alice keeps at home). It also requires Alice to be able to interact with the Anix network with some Anix-compatible device.

Algorithm 7 Hard identity revocation

Input: Old identity's signing keys ($PubKS_{ID_{old}}, PrivKS_{ID_{old}}$)
Output: Signal hard revocation of the old identity

$sig \leftarrow \texttt{Sign}_{PrivKS_{ID_{old}}}(PubKE_{ID_{old}}, PubKS_{ID_{old}})$
`revocationMessage` $\leftarrow PubKE_{ID_{old}}||PubKS_{ID_{old}}||sig$
`broadcastMessage(revocationMessage)`

Hard revocation works as follows. Alice signs one or more revocation certificates, where each certificate contains the public encryption and signing components of one of her current identity keys, signed by the identity's private signing key. Then, Alice can join the mesh network with some device and broadcast messages containing these revocation certificates. Similar to a PGP revocation certificate [38], each message signals any node that includes this key as part of their trusted contacts to neither trust any further communications coming from the user identified by that key, nor any other identity key that may be a result of a soft revocation of the key mentioned in the certificate (should the adversary perform soft revocation itself before Alice had the chance to broadcast her revocation certificates). Users can track this by hashing the revoked identity and checking it against their list of hashes of Alice's previous identities (as mentioned in "Soft revocation"). Algorithm 7 details the hard revocation steps. While hard revocations could also be preemptively issued by adversaries that seized a device (i.e., before Alice does it herself), this simply foils the adversary's (temporary) opportunity to impersonate Alice.

After having described how Anix users can manage trust relationships, we will now address how Anix's trust notions can help users rank and prioritize the messages they receive.

4.4. Message Endorsing and Ranking

We now describe votes and their role in the endorsement of Anix messages. Then, we detail how users can make use of votes, together with message authorship information, to reason about a message's trustability.

Votes and message endorsement. Anix allows for message endorsement in the form of votes, which we consider to be *upvotes* or *downvotes*, used for respectively approving and disapproving a message. For instance, if Alice is responsible for forwarding a message whose content she knows is misleading (regardless of whether she has established a trust relationship with the message's author), she can downvote that message to let her trusted contacts know about her opinion. As we will see later, messages can accrue votes over time, allowing Alice to issue votes on messages that she has already forwarded but had not yet voted on.

In practice, a vote is a signature of the hash of a message's full data (including its content, sender's PSU, and signature) denoted as M, concatenated with a single bit b, whose value is 1 for an upvote or 0 for a downvote. This signature incorporates the same public key blinding scheme used for PSU generation, as it uses the voting user's

identity key's private signing component ($PrivKS_{ID}$), alongside their identity key's public signing component ($PubKS_{ID}$) to sign the message in an unlikable fashion, i.e., $vote = \texttt{BSign}_{PubKS_{ID},PrivKS_{ID}}(H(M)||b)$. This ensures that only the users who know the identity key's public signing component $PubKS_{ID}$ can identify the vote's author and check the integrity of the received message. Recall that, in this case, bk is computed as $bk = H(M||b||PubKS_{ID})$.

Next, we showcase the utility of votes when reasoning about the trustability of messages exchanged through Anix.

Triaging incoming messages. Once Alice receives messages from another Anix node, she first attempts to pinpoint messages authored by her trusted contacts. To do so, she simply attempts to validate the signature included in the message's sender PSU using each one of her contacts' public identity keys. Then, Alice repeats a similar process to check whether some of her trusted contacts have cast any up/downvotes. The complexity of this process is of order $O(m \times c)$ where m is the number of all received messages that have to be verified, and c is the number of trusted contacts Alice possesses. While the delay to verify the signatures included in the PSUs scales linearly for both the number of contacts and messages, our experiments in §6.1 suggest that this process can be completed rather efficiently with each signature verification being completed within a few dozen milliseconds on commodity smartphones.

During message processing, nodes will also attempt to recognize OWT messages directed to them and either a) add the identity key included in a new OWT message targeted at them to their validation list, or; b) add the identity key obtained as a response to one of their previously sent OWT messages to their trusted contact list, thus completing the establishment of a bi-directional trust relationship.

Ranking trust on incoming messages. Once Alice verifies which of the received messages have been authored (and which votes have been cast) by her trusted contacts, she will compute a trust score (t) for each message, based on a formula ruled by the following factors: whether the message's author is a trusted contact (TCA), and how many trusted contacts upvoted (TCU) or downvoted (TCD) the message. Though we do not directly consider it in our design, other factors (e.g., for how long a user has been trusted) could also be considered when computing a message's trust score.

$$t = a.TCA + b.(TCU - TCD)$$

In this formula, a and b are coefficients used to tune the relevance given to each of the considered factors when computing a message's trust score. This formula will, first, largely increase the trust score of the message should it be authored by one of Alice's trusted contacts. The second most important component for attributing trust to a message is the number of up/downvotes generated by Alice's trusted contacts. Note that no importance is given to the trend of up/downvotes cast over the message coming from users who are unknown to Alice. Should a message not be authored or voted upon by any of Alice's trusted contacts, Anix displays these messages in a well-marked section of the app, outlining that they are not to be trusted since votes cast on them could have been artificially influenced by adversary-controlled nodes and do not necessarily reflect Alice's trusted circle sentiment about a message.

Also, as previously discussed when mentioning Anix's bi-directional trust relationships (see §4.3), the identities of users who have attempted to establish a bi-directional trust relationship with Alice but are still awaiting Alice to trust them back are kept in a validation list. Identities kept in the validation list are not accounted for in calculations depending on Alice's trusted contacts and, therefore, do not affect the messages' trust scores. This feature prevents adversaries from being able to artificially manipulate messages' trust scores by inflating a user's list of known public identities via the sending of OWT messages generated by Sybil identities.

Accruing message votes over time. Multiple copies of the same message can travel many different paths across the mesh network and eventually get delivered to the same user. Given that the votes that are tied to a given message will also be dependent on the path each of these copies takes, Anix allows votes to be accrued at a given device (should different copies of the message be received). This allows Anix to dynamically recompute the trust score that should be attributed to a message and, for instance, accordingly reorganize messages in Anix's microblogging feed.

Anix accrues votes on messages as follows. When receiving the first instance of a given message, Anix creates a string set [39] for holding the votes tied to that message (i.e., a *message vote set*). If a copy of the same message (and corresponding set of votes) is received again, Anix adds the new (non-duplicated) votes to that message's vote set.

Since the number of votes on a message can become large, we hypothesize that Anix could use probabilistic data structures such as Bloom filters [40] or bit-string caches [26] to optimize the vote-exchanging process. However, these data structures are prone to false positives, which could result in some votes being inaccurately recorded or lost (even without adversarial involvement). Exploring alternative data structures that would allow for a more efficient exchange of votes is a compelling direction for future work.

4.5. Point-to-Point Message Exchange

This section describes Anix messages' structure and how they are exchanged between peering devices. We start by focusing on the exchange of messages used in Anix's one-to-many communication model before addressing messages tailored to Anix's alternative communication models.

Message structure. The overall structure of a Anix message in the one-to-many setting consists of three sections:

1) *Message content:* For simplicity's sake, each message in Anix is considered a text-only message.
2) *Sender's PSU and signature:* A sender's PSU and a signature of the message, based on that PSU.
3) *Votes:* A compressed representation of the votes cast on the message itself.

In Appendix A, we detail the message formats used for Anix's one-to-many communication, alongside those used

for its one-to-one and some-to-some communication models. We discuss these communication models later in this section.

Endpoint discovery and selection. Similarly to existing work [7], [22], a user can discover other Anix users in her vicinity by using a mix of Wi-Fi Direct [41] and Bluetooth functionality. To make itself discoverable, a device sets the name of its Wi-Fi Direct beacon to some specially formatted name to broadcast the phone's Bluetooth MAC address (we use `ANIX-{MAC_ADDRESS}`). Once a device detects another endpoint following this naming convention, it attempts to connect to it via Bluetooth. If multiple endpoints are available, the device arbitrarily selects one and exchanges data with it, before moving on to the next. We use this hybrid approach since neither setting the name of a Wi-Fi Direct beacon nor connecting to another Bluetooth device whose MAC is known requires explicit user input.

Message exchange and storage. During a message exchange, both parties send the messages they hold in storage in a random order, along with their corresponding votes. We choose random ordering as opposed to prioritizing messages with higher trust scores since the latter could leak information about the trusted contacts of the forwarding node.

Considering that the devices where Anix is installed may also have limited space for storing messages, the app allows users to define a storage quota for saving messages and their associated votes. Anix then provides two mechanisms (a proactive one and a reactive one) for helping users manage the storage space that the app uses on their devices.

Firstly, to proactively help manage the growth of their message storage, Anix nodes keep a timestamp that records the time at which each message was received (rt_m). This timestamp is used in an aging process which is responsible for calculating the time at which a message should be deleted after being received (dt_m). Messages are deleted after being kept in storage for a persistence time (pt_m), such that $dt_m = rt_m + pt_m$. In addition, while we do not address this in our evaluation, the persistence time of different message types can be separately configurable. For instance, OWT messages can be configured to persist for a longer time than regular microblogging messages, allowing users additional time for reasoning about the intention to establish bi-directional trust relationships (see §4.3).

Secondly, Anix will reactively delete messages if an exchange causes the quota to be exceeded. To this end, the app will first randomly delete messages that have not been authored or voted upon by any trusted contact. In the event that all of these messages are deleted but the user is still above the storage quota, Anix further deletes the remaining messages according to their trust score (messages with lower trust get deleted first) in order to meet the quota.

Bounding message propagation. Epidemic routing often leverages a maximum hop count to determine the number of epidemic exchanges that a message is subject to [32], [42]. In Anix's case, however, deterministically setting a hop count (even if randomly within some bound [43]) could leak information to an adversary about their proximity to the author of a message [44]. To avoid disclosing information about the source of a given message while limiting the propagation of a message throughout the network, Anix keeps a table connecting a message's hash to the corresponding timestamp of the time the message was received (rt_m). The message hash includes all the components of the message except the potential votes, so as long as the message is not tampered with, the hash remains static. This table's entries persist even after a message has been deleted, so upon receiving a message, a node can refer to this table and verify whether the message has already been received, in which case the node acts according to the table entry for the received time, so if the local persistence time pt_m for this message has elapsed, the message will just get dropped. On the other hand, if the message was not previously received, the message gets added to the table and gets processed as normal. Table entries persist for a time duration $lt_m >> pt_m$ to ensure that proactively deleted messages are not erroneously accounted again as new messages.

Alternative communication models. Besides its focus on message broadcasting, Anix also natively supports one-to-one and some-to-some messaging as follows.

One-to-one communication. Anix facilitates one-to-one (O2O) communication by encrypting the message's payload with the receiver's public key before transmitting it over the network. In contrast to microblogging messages that are meant to be interpreted by all mesh users, private O2O messages do not expose identifiable public metadata; their payload is concatenated with the sender's key blinded signature over the message (see §4.2) while conforming to a fixed length (padded if needed), and then encrypted. The key used to encrypt a O2O message may be the public encryption key tied to a PSU (e.g., when generating an OWT), or, more commonly, the public encryption component of a temporary identity key (e.g., for private communication between two users who mutually trust each other).

Some-to-some communication. In Anix, group communication (S2S) is a direct extension of one-to-one messaging. To create a private messaging group, Alice generates a symmetric *group encryption key* and builds a group invitation message for each contact she wishes to invite by encrypting the group key with each user's identity key's public encryption component. Then, she broadcasts these messages. Once the intended users receive their group invitation, they can access and send messages encrypted with the group's key. As in Anix's O2O scheme, messages sent to a group can be signed using a key blinded signature. Should the group key be leaked, messages can be decrypted but cannot be trivially linked back to temporary user identities due to key blinding.

5. Evaluation Methodology

This section outlines our evaluation goals and the approach we used for measuring the quality of our solution.

5.1. Evaluation Goals and Approach

Our evaluation aims to assess the practicality of Anix over three main aspects:

Performance. We aim to assess the efficiency with which Anix can exchange messages between two devices. To this end, we conduct a set of micro-benchmarks based on a Anix software prototype, together with its deployment on real smartphone devices. We collect metrics such as the transmission time required for completing message exchanges, the computation time required to accomplish Anix's basic operations, and Anix's impact on battery consumption.

Resilience. We rely on simulations to gauge an adversary's ability to successfully manipulate a user's perception of misinformation transmitted through Anix and to build trust with legitimate Anix users. To assess the former, we compare the number of benign and misinformation messages that were up/downvoted by a majority of benign users. To assess the latter, we consider the number of adversarial nodes that received OWT messages authored by benign users.

Latency and coverage. We use simulations to gauge Anix's message propagation times across the mesh when assuming the presence of different ratios of adversarial nodes.

5.2. Experimental Testbed

This section describes the deployment settings of our Anix prototype and the parameters used in our simulation.

Anix prototype and hardware settings. We implemented a proof-of-concept of Anix [24] and installed it in two different smartphones to perform our hardware-based micro-benchmarks. We developed our prototype for Android v8.0 (or later) using 2 200 lines of Java code and implemented a simple epidemic routing protocol for disseminating messages. In our implementation, we assume the payload of each O2M message to have a maximum size of 256B, and we explicitly pad the message payload of OWT, O2O and S2S messages to a fixed size to ensure these messages exhibit the same size on the network. We use BouncyCastle's Java ECIES [45] implementation over secp256r1 for performing encryption/decryption of OWT, O2O, and S2S messages, and use the same library's implementation of EdDSA [46] over Ed25519 for signing/verifying Anix O2M messages. We also implemented Denis et al. [36] EdDSA-based [46] key blinding scheme in Java to create unlinkable signatures used in PSUs and votes.

We deployed our prototype in a Samsung Galaxy A04 (ARM Cortex-A53 CPU, 4GB RAM) and ZTE Blade V40 Smart (Unisoc ums9230 CPU, 3GB RAM). We rely on Bluetooth v5.0 (or later) for performing point-to-point communication between devices due to its widespread adoption and because it allows for automatic message routing without the need to root the device (unlike Wi-Fi Direct) [7].

Mesh network simulator and considered parameters. We implemented a simulator (1500 lines of Python code) to reproduce the activities of legitimate Anix users and adversarial nodes within a city during a blackout. Below, we briefly describe the parameters we consider (see a full rationale for these choices in Appendix B), and distinguish which of these are fixed throughout our simulations from those whose influence we directly evaluate in our experiments via different scenarios; we describe these scenarios later in this section, after introducing all parameters of interest.

Movement of Anix nodes. Based on ASMesh's [23] simulator, we represent the world as an $A \times B$ discrete grid, where a total of N users, including a ratio of Adv adversarial nodes, move around and interact throughout T steps. At each step, each user can move up to m cells in any direction, exchanging messages with other users when co-located in the same cell. The movement of adversarial nodes follows the same rules. We consider a blackout with a duration of 5 days and one hour-long steps, resulting in a total of $T = 120$ simulation steps. We fix other parameters as $A = B = 25, N = 600, m = 2$. In practice, each cell of the grid is equivalent to an area of 0.25 square kilometers. Lastly, we leave Adv as a configurable value.

Message generation and storage. We assume that users pick up their phones and interact with the Anix app at any given simulation step with probability P_{inter} and create and send out new messages with probability C_m. We also assume each user configures the Anix app with a storage space of S GBs. Finally, users set the persistence time of each message (tp_m), in steps, to steer Anix's proactive message aging and deletion. We fix all these parameters as $P_{inter} = 0.15, C_m = 0.05, S = 3$, $tp_m = 24$.

Users' social graph. To bootstrap the initial social graph between Anix users, we rely on the Watts-Strogatz model for small worlds [23], [47]. We assume that at the start of the simulation, some users already had the chance to become trusted contacts from one another, e.g., by manually exchanging identity keys or via Anix's bi-directional trust establishment. The model uses two parameters, K and β, for the mean degree and graph creation randomness, respectively. We set $K = 15$ to reflect that each user is initially connected to 2.5% of the overall population and set $\beta = 0.5$.

As mentioned in §3, we assume the adversary is less connected with legitimate users than an average benign user. We reflect this through R, where $R = 1$ means that the adversary's trusted contact circle is equally well connected to legitimate users as that of an average benign user, while $R = 0$ means that adversary nodes have no initial connections to benign users. In §6, we analyze the influence of R on an adversary's ability to earn benign users' trust.

Trusting messages and their authors. Our simulation also models the interactions between users (legitimate or otherwise) by simulating the creation of new trust relationships and interaction with messages from unknown sources. For simplicity during our simulations, we obviate specific values for the coefficients of our trust formula in §4.4 (a and b), and abstract them as the parameters mentioned below. We note however, that these coefficients can be experimentally calibrated as more realistic Anix usage data is gathered. OWT_{ud} and U_{ud} represent the minimum required ratios between the upvotes and downvotes on a message issued by a user's trusted contacts so that a user chooses to send an OWT to the message's author or, respectively, to upvote the message. We fix these as $U_{ud} = 0.55, OWT_{ud} = 0.66$.

TABLE 2. MESSAGES MAJORLY UP/DOWNVOTED AND OWTS ISSUED BY BENIGN USERS, FOR DIFFERENT AWARENESS SCENARIOS ($Adv = 0.02$).

Scenario ($Adv = 0.02$)	**Parameters**				**Benign**		**Misinformation**		**OWTs**	
	R	UV	UM	UN	Upvoted	Downvoted	Upvoted	Downvoted	Benign	Adversarial
Very naive	0.9	0.2	0.5	0.5	495	2522	204	1164	33581	106
Naive	0.7	0.1	0.4	0.55	1087	1874	40	1301	32278	43
Default	0.4	0.05	0.3	0.6	1510	1416	25	1320	31207	11
Aware	0.2	0.02	0.2	0.7	2111	704	15	1314	26115	5
Very Aware	0.1	0.01	0.1	0.8	2549	348	5	1297	15497	2

In addition, parameter UV denotes the probability that a user chooses to vote on a message for which they have neither authorship nor voting information provided by a trusted contact. In case a user chooses to vote on such a message, parameter UM denotes the probability that a user upvotes a message containing misinformation, and parameter UN denotes the probability that a user upvotes a legitimate message authored by a benign user. ($1 - UM$ and $1 - UN$ denote the probability that the user downvotes instead.) We vary these parameters throughout our experiments.

Anix user awareness-based scenarios. Adversaries targeting Anix will strive to spread misinformation while trying to gain the trust of benign users towards changing their perception of malicious messages. A fair part of the adversary's effectiveness thus depends on how aware benign users are of the adversary's tactics. Throughout our simulations, we rely on five different scenarios that aim to model the awareness of benign users and how this translates into the ability of Anix users to keep misinformation at bay. The more aware the benign user base is, the less they trust adversaries (R), vote on messages with unknown authors and voters (UV), and upvote misinformation messages (UM) because they can somehow be critical of them. They also identify benign users' messages more easily (UN).

The scenarios we consider, as well as the values for the above parameters considered in each scenario, can be seen in Table 2. While the description of our simulation results (see §6.2) focuses on an adversary whose nodes make up to 2% of the overall Anix network ($Adv = 0.02$), Appendix C addresses more stringent scenarios where the adversary makes up to 5%, 10%, and 25% of the overall user-base within the Anix network. Nevertheless, the trends observed below still hold for these alternative scenarios.

6. Evaluation Results

We now analyze Anix's performance and the impact of various attacks on Anix's coverage and resilience. Then, we compare Anix's resilience to that of Rangzen's.

6.1. Micro-benchmarks

Latency of node discovery. We measured the time required for a Anix device to be able to find and connect to another device in its vicinity (via the hybrid Wi-Fi Direct and Bluetooth pairing described in §4.5). Our experiments revealed that our devices can complete this operation in ≈1.8s.

Message exchange times. We measured the necessary time for our devices to conclude a message exchange. In this experiment, each device sends 100 messages to the other.

TABLE 3. COMPUTATION TIME (IN MS) FOR ANIX'S OPERATIONS.

Op./Device	**Gen. PSU**	**Create Msg.**	**Create Vote**	**Verify Sig.**	**`BVer` (Alg. 3)**
Samsung A40	175.06 ± 1.05	46.30 ± 0.01	84.61 ± 1.14	61.33 ± 0.21	67.68 ± 0.21
ZTE Blade V40	64.95 ± 0.29	19.75 ± 0.01	38.76 ± 0.32	43.29 ± 0.28	47.30 ± 0.48

Following Rangzen's [7] average message payload sizes, we set the length of each O2M message payload to 140B. This amounts to a total of 332B, accounting for each message's corresponding PSU (128B) and signature (64B). We spread 10 000 votes (64B each) amongst the messages held by each device, amounting to a total communication volume of ∼1300KB. Our results reveal that this exchange is completed in 11.58s, indicating an average Bluetooth transmission speed of ∼1.28Mbps (akin to recent work [26]).

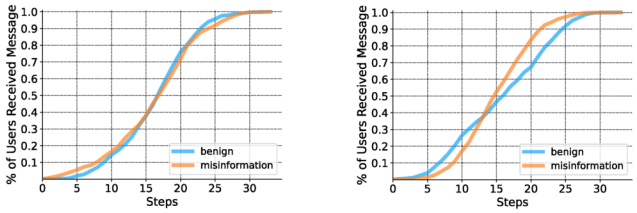
Computation time for basic operations. We measured the time required to complete the operations that support Anix's main functionalities, namely: generating PSUs; creating messages (excluding the generation of the PSUs that are appended to them); creating votes, and; verifying signatures (i.e., the backbone of Anix's operations when processing received messages). Table 3 shows the time (in milliseconds) spent on each operation, averaged after 10 000 trials. These numbers suggest that Anix's basic operations can be performed swiftly and that the necessary cryptographic operations to assess messages' authenticity and triage incoming messages/votes (see §4.5) do not add substantial overheads.

Battery consumption. We use Anix to continuously exchange messages between two devices for a full hour and measure the battery consumption imposed when running the app through Android's internal battery monitor. We find that Anix consumes 87.68mAh charge on our ZTE device per hour, which amounts to 1.5% of its 6000mAh battery.

6.2. Simulations

Anix users can weed out misinformation. Table 2 shows how many benign (and misinformation) messages were up/downvoted by a majority of benign users across different user awareness scenarios, suggesting that the more aware users are of the adversary, the less pronounced the effect of the adversary's actions; users tend to distrust misinformation and better assess the trustworthiness of benign messages. However, even for the *naive* user scenario, only 40 out of 1341 misinformation messages are majorly upvoted.

Anix users can avoid trusting adversarial nodes. Table 2 also shows how many benign/malicious users were able to gain the trust of other benign users (via OWT). We see that the vast majority of OWT messages are targeted at benign users instead of adversarial nodes. While users in the default or aware scenarios hardly ever gain enough confidence to trust adversary nodes (e.g., 11 OWT messages in the *default*



(a) No active DoS. (b) Active DoS.

Figure 1. Anix's coverage in function of message delivery latency.

scenario), Anix prevents even *Very naive* users from heavily trusting adversarial nodes (106 OWT messages). Users that mistakenly trust the adversary may still make use of identity revocation primitives (see §4.3) at some point in time.

Anix achieves a reasonable coverage and latency. We measured the time it takes for new messages to reach different fractions of the network in the *default* user awareness scenario (an analysis for other scenarios can be found in Appendix C). We assess this metric separately for benign and misinformation messages to understand whether the presence of adversarial nodes can favor the spread of misinformation. As depicted in Figure 1(a), our results show that, on average, a benign message takes 23 steps to reach 90% of all Anix users. As each step correlates to one real-world hour in our simulation, an average message takes roughly a day to reach more than 90% of the total users of the network. The figure also shows that, in this scenario, misinformation messages move at a similar speed across the mesh.

DoS attacks do not significantly degrade coverage or latency. In addition to the above, adversary nodes can choose not to forward any kind of benign messages, and spread only misinformation messages authored by themselves. Figure 1(b) reveals that, while misinformation messages can now travel faster through the mesh (reaching 90% of all users in about 21 steps), these attacks do little to impede legitimate communications since the average time to deliver benign messages to 90% of all Anix users only increases 2 steps (up to 25) when compared to the absence of DoS.

6.3. Comparing Rangzen's and Anix's Resilience

In this section, we compare the resilience of Anix to that of Rangzen's in regards to the spreading of misinformation messages. To perform this comparison, we reproduced the operation of Rangzen in our mesh simulator (with the same world and adversary configurations used for Anix in §5.2).

As described in §2.3, the messages that Rangzen shows to users are prioritized based on their trust scores. Further, messages below a certain trust threshold T_{thr} can also be hidden from users to limit the spreading of misinformation [7]. Given this, we configured an experiment aimed at assessing how many benign and misinformation messages are effectively shown to (i.e., deemed trusted by) more than 50% of Rangzen's users. (This is intentionally reminiscent of the concept of majorly upvoted benign/misinformation messages in Anix to facilitate comparisons.)

Table 4 depicts the results of our experiments for different trust thresholds T_{thr} (we omit results for $T_{thr} > 1e{-}07$ as they remain unchanged). Our results suggest that, without defining a threshold ($T_{thr} = 0$), there is a substantial risk that Rangzen may allow adversaries to disseminate misinformation – indeed, 1184 of the misinformation messages exchanged throughout the simulation are shown to at least 50% of users. While implementing a threshold helps reducing the number of misinformation messages shown to users, this also limits the number of benign messages shown to them. For instance, while 171 misinformation messages are seen by at least 50% of users when $T_{thr} = 1e^{-10}$, only 369 benign messages get to be seen ($\approx 2\times$ vs. misinformation).

TABLE 4. # OF MAJORLY SHOWN/NOT SHOWN MESSAGES IN RANGZEN.

Msg. Status	T_{thr}	0	$1e^{-13}$	$1e^{-12}$	$1e^{-11}$	$1e^{-10}$	$1e^{-8}$	$1e^{-7}$
Shown to >50% users	Benign	2901	1917	1558	769	369	3	0
	Misinf.	1184	868	703	305	171	0	0
Hidden from >50% users	Benign	600	1584	1943	2732	3132	3498	3501
	Misinf.	218	534	699	1097	1231	1402	1402

In contrast, Anix achieves a more effective balance between the benign messages shown to users and the misinformation messages it can weed out. For instance, even when users are *naive* about the activities of the adversary (see Table 2), Anix users only majorly upvote (i.e., trust) 40 misinformation messages while majorly upvoting (i.e., trusting) 1087 benign messages ($\approx 27\times$ vs. misinformation).

7. Related Work

Blackout-resistant data exchange. In addition to the use of mobile devices to build ad-hoc mesh networks, other research efforts have analyzed the deployment of scalable data exchange mechanisms that can resist blackouts.

Community-based mesh networks. These networks rely on users who coordinate their efforts to build resilient communication infrastructure making use of wireless and/or wired mediums [48]. Besides extending coverage to remote regions, these meshes have been touted to help resist internet censorship campaigns and increase net neutrality [49], [50]. Prominent examples include the `guifi.net` [51] network in Spain and AlterMundi project [52] in Argentina. In contrast to Anix, these networks demand more specialized communication equipment with higher maintenance efforts [53].

Satellite-based communication. Other solutions for tackling shutdowns have explored satellite-based data transmissions. Projects such as Toosheh [54] and Outernet [55], embed digital content within video streams transmitted over satellite TV channels. Yet, these can only be used to consume content that is curated by the projects' operators. Recently, Starlink has enabled Internet connectivity to disconnected areas during warfare [56]. However, satellite internet alternatives require users to possess specialized equipment which oftentimes must be smuggled into affected regions [57].

Cellular-based data exchange. Dolphin [58] allows for the exchange of data by modulating it over cellular voice connections. In contrast to Anix, Dolphin assumes that cellular voice infrastructure is operational during shutdowns.

Practical anonymous communication systems. There are several practical and usable anonymous communication systems, such as Freenet/Hyphanet [59], Tor [60], I2P [61],

or Nym [62]. Anix draws inspiration from these systems but aims to provide a secure platform for anonymously exchanging information during internet shutdowns.

Privacy-preserving reputation systems. Solutions like reputation transfer systems, or coin- and ticket-based reputation systems can provide reputation tracking without hindering users' privacy [63]. However, some of these solutions rely on trusted entities for handling the management of reputation [64], while others require internet access [65]. In turn, Anix allows a mesh network's users to progressively build trust relationships through the analysis of messages' content and message endorsements, while retaining anonymity.

8. Conclusion

We introduced Anix, a novel blackout-resistant messaging app that addresses the need for secure and anonymous microblogging-alike communication during internet shutdowns. Through new trust establishment and anonymous message endorsing features, Anix allows users to maintain privacy while managing trust relationships and assessing the trustworthiness of messages. Our evaluation suggests that Anix is resilient to attackers that pose as legitimate users.

Limitations and future work. Anix's hard revocations offer post-compromise anonymity but require swift action from the affected users so that they can issue revocation certificates and resist impersonation attempts. This may be difficult in practice, e.g., if the user whose device was seized is temporarily detained for interrogation. An interesting direction for future work would include the exploration of anti-forensic mechanisms for automating revocations without user involvement, e.g., by automatically issuing revocation certificates from a secondary device if the user does not periodically cancel a "self-destruct" countdown timer on it.

China is the first country to crack down on mobile ad-hoc file-sharing apps [66], [67], targeting commercial solutions like Apple's AirDrop, and Bluetooth-based apps. Possible upcoming restrictions envision banning the sale of commercial apps, not affecting apps like Bridgefy, Briar, or Anix. Currently, ad-hoc Bluetooth communications are not restricted, but future regulations might prompt the development of enhanced mesh protocols which hide the fact that users are using blackout-resistant messaging apps altogether.

Other research directions aligned with our work may also include the study of alternative schemes for the selective linking of pseudonyms and pseudonym revocation, e.g., by adapting recent solutions for other contexts such as signing code commits [68] and VANET scenarios [69], respectively.

Acknowledgments

We express our gratitude to our shepherd and the anonymous reviewers for their insightful comments. We also thank Vecna and Ian Goldberg for their helpful feedback about this work. This work was supported in part by NSERC under grant RGPIN-2023-03304, and benefited from the use of the CrySP RIPPLE facility at the University of Waterloo.

References

[1] A. Shahbaz, A. Funk, and K. Vesteinsson, "Countering an authoritarian overhaul of the internet (freedom on the net)," https://freedomhouse.org/report/freedom-net/2022/countering-authoritarian-overhaul-internet, 2022.

[2] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé, "Analysis of country-wide internet outages caused by censorship," in *Proceedings of the Internet Measurement Conference*, 2011, pp. 1–18.

[3] M. Chulov, "Syria shuts off internet access across the country," https://www.theguardian.com/world/2012/nov/29/syria-blocks-internet, 2012.

[4] R. Padmanabhan, , A. Filastò, M. Xynou, R. S. Raman, K. Middleton, M. Zhang, D. Madory, M. Roberts, and A. Dainotti, "A multi-perspective view of Internet censorship in Myanmar," in *Free and Open Communications on the Internet*, 2021.

[5] Z. S. Bischof, K. Pitcher, E. Carisimo, A. Meng, R. B. Nunes, R. Padmanabhan, M. E. Roberts, A. C. Snoeren, and A. Dainotti, "Destination unreachable: Characterizing Internet outages and shutdowns," in *Proceedings of the Special Interest Group on Data Communications Conference*, 2023.

[6] S. Hasan, Y. Ben-David, G. Fanti, E. Brewer, and S. Shenker, "Building dissent networks: Towards effective countermeasures against large-scale communications blackouts," in *Free and Open Communications on the Internet*, 2013.

[7] A. Lerner, G. Fanti, Y. Ben-David, J. Garcia, P. Schmitt, and B. Raghavan, "Rangzen: Anonymously Getting the Word Out in a Blackout," 2016, arXiv:1612.03371 [cs].

[8] M. R. Albrecht, J. Blasco, R. B. Jensen, and L. Mareková, "Collective information security in Large-Scale urban protests: the case of hong kong," in *Proceedings of the 30th USENIX Security Symposium*, 2021, pp. 3363–3380.

[9] P. Shadbolt, "Firechat in hong kong: How an app tapped its way into the protests," https://www.cnn.com/2014/10/16/tech/mobile/tomorrow-transformed-firechat/index.html, 2014.

[10] "Briar," https://briarproject.org/, 2024.

[11] "Bridgefy," https://bridgefy.me/, 2024.

[12] M. R. Albrecht, J. Blasco, R. B. Jensen, and L. Mareková, "Mesh messaging in large-scale protests: Breaking bridgefy," in *Topics in Cryptology – CT-RSA 2021*. Springer, 2021, pp. 375–398.

[13] M. R. Albrecht, R. Eikenberg, and K. G. Paterson, "Breaking bridgefy, again: Adopting libsignal is not enough," in *Proceedings of the 31st USENIX Security Symposium*, 2022, pp. 269–286.

[14] J. Koetsier, "Ukrainians prepping for internet loss by getting apps for offline, private, mesh communications," https://www.forbes.com/sites/johnkoetsier/2022/02/25/ukrainians-prepping-for-internet-loss-by-getting-apps-for-offline-private-mesh-communications/?sh=710a40b7791d, 2022.

[15] F. Potkin and J. Pang, "Offline message app downloaded over million times after myanmar coup," https://www.reuters.com/article/idUSKBN2A22H0/, 2021.

[16] D. Windelspecht, "What journalists need to know to report on protests in iran," https://ijnet.org/en/story/what-journalists-need-know-report-protests-iran, 2022.

[17] J. H. Parmelee and S. L. Bichard, *Politics and the Twitter revolution : how tweets influence the relationship between political leaders and the public*, ser. Lexington studies in political communication. Lexington Books, 2013.

[18] A. Ronzhyn, "The use of facebook and twitter during the 2013-2014 protests in ukraine," in *Proceedings of the European Conference on Social Media*, 2014, pp. 442–449.

[19] D. Zirugo and A. Mare, "Assessing twitter's revolutionary potential in an authoritarian regime," *Digital Technologies, Elections and Campaigns in Africa*, 2023.

[20] S. L. Myers and P. Mozur, "China is waging a disinformation war against hong kong protesters," https://www.nytimes.com/2019/08/13/world/asia/hong-kong-protests-china.html, 2019.

[21] H. Murphy, "Ominous Text Message Sent to Protesters in Kiev Sends Chills Around the Internet," https://archive.nytimes.com/thelede.blogs.nytimes.com/2014/01/22/ominous-text-message-sent-to-protesters-in-kiev-sends-chills-around-the-internet/, 2014.

[22] A. Pradeep, H. Javaid, R. Williams, A. Rault, D. Choffnes, S. Le Blond, and B. Ford, "Moby: A Blackout-Resistant Anonymity Network for Mobile Devices," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, no. 3, pp. 247–267, 2022.

[23] A. Bienstock, P. Rösler, and Y. Tang, "ASMesh: Anonymous and secure messaging in mesh networks using stronger, anonymous double ratchet," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 1–15.

[24] "Anix source code and simulator." https://www.github.com/kamalisina/anix.

[25] Y. Liu, D. R. Bild, D. Adrian, G. Singh, R. P. Dick, D. S. Wallach, and Z. M. Mao, "Performance and energy consumption analysis of a delay-tolerant network for censorship-resistant communication," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2015, pp. 257–266.

[26] N. Perry, B. Spang, S. Eskandarian, and D. Boneh, "Strong Anonymity for Mesh Messaging," 2022, arXiv:2207.04145 [cs].

[27] "X. happening now." https://x.com/, 2024.

[28] "Reddit - dive into anything," https://www.reddit.com/, 2024.

[29] "Bramble transport protocol, version 4," https://code.briarproject.org/briar/briar-spec/-/blob/master/protocols/BTP.md, 2024.

[30] "Signal," https://signal.org, 2024.

[31] E. De Cristofaro, P. Gasti, and G. Tsudik, "Fast and private computation of cardinality of set intersection and union," in *Proceedings of the International Conference on Cryptology and Network Security*, 2012, pp. 218–231.

[32] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," *Tech. Report CS-200006, Duke University*, 2000.

[33] D. Xue, A. Ablove, R. Ramesh, G. K. Danciu, and R. Ensafi, "Bridging barriers: A survey of challenges and priorities in the censorship circumvention landscape," in *Proceedigns of the 33rd USENIX Security Symposium*, 2024.

[34] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, pp. 36–63, 2001.

[35] S. Schmieg, "Reconstructing public keys from signatures ," https://keymaterial.net/2024/06/15/reconstructing-public-keys-from-signatures/, 2024.

[36] F. Denis, E. Eaton, T. Lepoint, and C. A. Wood, "Key Blinding for Signature Schemes ," https://www.ietf.org/archive/id/draft-irtf-cfrg-signature-key-blinding-06.html, 2024.

[37] E. Eaton, T. Lepoint, and C. A. Wood, "Security analysis of signature schemes with key blinding," Cryptology ePrint Archive, Paper 2023/380, 2023. [Online]. Available: https://eprint.iacr.org/2023/380

[38] E. Gerck *et al.*, "Overview of Certification Systems: x. 509, CA, PGP and SKIP," *The Black Hat Briefings*, vol. 99, 1997.

[39] D. Fenz, D. Lange, A. Rheinländer, F. Naumann, and U. Leser, "Efficient similarity search in very large string sets," in *Proceedings of the 24th Scientific and Statistical Database Management*, 2012, pp. 262–279.

[40] S. Geravand and M. Ahmadi, "Bloom filter applications in network security: A state-of-the-art survey," *Computer Networks*, vol. 57, no. 18, pp. 4047–4064, 2013.

[41] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE wireless communications*, vol. 20, no. 3, pp. 96–104, 2013.

[42] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Transactions on networking*, vol. 14, no. 3, pp. 479–491, 2006.

[43] S. Seys and B. Preneel, "Arm: Anonymous routing protocol for mobile ad hoc networks," *International Journal of Wireless and Mobile Computing*, vol. 3, no. 3, pp. 145–155, 2009.

[44] R. Guerraoui, A.-M. Kermarrec, A. Kucherenko, R. Pinot, and S. Voitovych, "On the inherent anonymity of gossiping," 2023, arXiv:2308.02477 [cs.DC].

[45] V. G. Martínez, L. H. Encinas *et al.*, "A comparison of the standardized versions of ecies," in *2010 Sixth International Conference on Information Assurance and Security*. IEEE, 2010, pp. 1–4.

[46] S. Josefsson and I. Liusvaara, "Edwards-curve digital signature algorithm (eddsa)," Tech. Rep., 2017.

[47] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[48] "Community Connectivity: Building the Internet from Scratch - Annual Report of the UN IGF Dynamic Coalition on Community Connectivity," 2016.

[49] P. De Filippi and F. Tréguer, "Expanding the internet commons: The subversive potential of wireless community networks," *Journal of Peer Production, Issue*, no. 6, 2015.

[50] D. Keller, *Towards an Internet Free of Censorship II: Perspectives in Latin America*. Facultad de Derecho, Centro de Estudios en Libertad de Expresión y Acceso a la Información, 2017.

[51] R. Baig, R. Roca, F. Freitag, and L. Navarro, "guifi.net, a crowdsourced network infrastructure held in common," *Computer Networks*, vol. 90, pp. 150–165, 2015.

[52] N. J Bidwell, "Wireless in the weather-world and community networks made to last," in *Proceedings of the 16th Participatory Design Conference 2020*, 2020, p. 126–136.

[53] P. Garrison, E. H. B. Jang, M. A. Lithgow, and N. A. Pace, "The Network Is an Excuse: Hardware Maintenance Supporting Community," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW2, 2021.

[54] M. Yahyanejad, "Methods and systems for digital data transmission through satellite TV channels," U.S. Patent 10 448 116, 2019.

[55] S. Karim, A. Rogers, and E. Birrane, "Bridging the Information Divide: Offering Global Access to Digital Content with a Disruptive CubeSat Constellation," in *Proceedings of the 28th Annual AIAA/USU Conference on Small Satellites*, 2014.

[56] I. Aviv and U. Ferri, "Russian-Ukraine armed conflict: Lessons learned on the digital ecosystem," *International Journal of Critical Infrastructure Protection*, vol. 43, p. 100637, 2023.

[57] K. Vick, "Inside the Clandestine Efforts to Smuggle Starlink Internet Into Iran," https://time.com/6249365/iran-elon-musk-starlink-protests/, 2023.

[58] P. K. Sharma, R. Sharma, K. Singh, M. Maity, and S. Chakravarty, "Dolphin: a cellular voice based internet shutdown resistance system," *Proceedings on Privacy Enhancing Technologies*, pp. 589–607, 2023.

[59] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Designing privacy enhancing technologies: international workshop on design issues in anonymity and unobservability*. Springer, 2001, pp. 46–66.

[60] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router." in *Proceedings of the USENIX security symposium*, 2004, pp. 303–320.

[61] B. Zantout, R. Haraty *et al.*, "I2P data communication system," in *Proceedings of the 1st ACM SIGCOMM Workshop on Information-Centric Networking*, 2011, pp. 401–409.

[62] C. Diaz, H. Halpin, and A. Kiayias, "The Nym Network," https://nymtech.net/nym-whitepaper.pdf, 2021.

[63] S. Gurtler and I. Goldberg, "SoK: Privacy-Preserving Reputation Systems," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 1, pp. 107–127, 2021.

[64] N. Busom, R. Petrlic, F. Sebé, C. Sorge, and M. Valls, "A privacy-preserving reputation system with user rewards," *Journal of Network and Computer Applications*, vol. 80, pp. 58–66, 2017.

[65] E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin, "Reputation Systems for Anonymous Networks," in *Privacy Enhancing Technologies*, 2008, vol. 5134, pp. 202–218.

[66] K. Ng, "Chinese censors take aim at AirDrop and Bluetooth," https://www.bbc.com/news/world-asia-china-65830185, 2023.

[67] W. Yun, "Is Apple's AirDrop safe to use in China?" https://www.rfa.org/english/news/china/china-apple-airdrop-01152024112633.html.

[68] K. Merrill, Z. Newman, S. Torres-Arias, and K. R. Sollins, "Speranza: Usable, privacy-friendly software signing," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, p. 3388–3402.

[69] C. Correia, M. Correia, and L. Rodrigues, "Using range-revocable pseudonyms to provide backward unlinkability in the edge," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, p. 3018–3032.

[70] R. Mitchell, "Tracking Internet Shutdowns in 2023," https://pulse.internetsociety.org/blog/tracking-internet-shutdowns-in-2023.

[71] E. Marchant and N. Stremlau, "A spectrum of shutdowns: Reframing internet shutdowns from africa," *International Journal of Communication*, vol. 14, 2020.

[72] Statista, "Average time spent per day on select social media platforms in the United States in 2023," https://www.statista.com/statistics/1301075/us-daily-time-spent-social-media-platforms/.

[73] S. Wojcik and A. Hughes, "Sizing up twitter users," *PEW research center*, vol. 24, pp. 1–23, 2019.

Appendix A. Anix Message Formats

PSU structure.

$PubKE_{OTU}$ (32 Bytes)	$PubKS_{OTU}$ (32 Bytes)
$\text{BSign}_{PubK_{ID},PrivK_{ID}}(PubKE_{OTU}\|\|PubKS_{OTU})$ (64 Bytes)	

One-to-many (O2M) message structure.

PSU_{sender} (128 Bytes)	$\text{Sign}_{PrivKS_{OTU_{sender}}}(m)$ (64 Bytes)
Message m (max. 256 Bytes)	

One-way-trust (OWT) message structure.

$\text{Enc}_{PubKE_{OTU_{receiver}}}(PubKE_{ID_{sender}}\|\|PubKS_{ID_{sender}})$ (405 Bytes under ECIES, where the payload is the concatenation of public identity keys padded to 320B)

One-to-one (O2O) message structure.

$\text{Enc}_{PubKE_{ID_{receiver}}}(\text{BSign}_{PrivKS_{ID_{sender}}}(m)\|\|m)$ (405 Bytes under ECIES, where m is 256 Bytes – padded if needed – and the signature is 64 Bytes)

Some-to-some (S2S) group invitation message structure.

$\text{Enc}_{PubKE_{ID_{receiver}}}(SimKEgroup)$ (405 Bytes under ECIES, where the payload is a symmetric group encryption key padded to 320 Bytes)

Appendix B. Rationale on Simulation Parameters

Table 5 summarizes the parameters of our simulations. Below, we detail the rationale for each parameter's value.

Network scale. We drew the configurations for parameters (A,B,m,N,β) – related to the scale of the network – directly from ASMesh's model [23]. We set K=15, meaning that, at the start of the simulation, each user is deemed as a trusted contact of an average 2.5% of all other users within a city (where the total number of users is given by N=600).

Simulation duration. We set T=120 simulation steps of 1 hour each (5 days) as this interval sits within a common duration for reported Internet shutdowns, e.g., during civil unrest and high-profile political events [70], also being longer than a sizable fraction of shutdowns consistently reported to last less than a day (e.g., as in India [71]).

Adversarial power. Our evaluation throughout § 6 assumed a ratio of 2% adversarial nodes (*Adv*=0.02) following the reasoning provided in Rangzen (see [7] – Sec.4.1). In Appendix C, we provide results for the effects of larger *Adv* ratios on the degradation of Anix's latency and coverage.

Anix device configurations. We set S to 3GB, amounting to 2% of the storage available in each of our smartphones.

User interactions with the Anix app. We fixed P_{inter} at 0.15, based on an average Anix usage of 4 hours per day (4/24= 0.167). This is consistent with recent surveys on social media app usage [72], from where we estimate an average total usage of social media apps (extrapolated from the combined usage time for the top seven trending apps) of $\approx$ 4h/day. We fixed C_m at 0.05. While being conservative, we believe this value to be consistent with existing studies that revealed that most Twitter users rarely issue posts [73].

Anix-specific parameters. We fixed OWT_{ud} and U_{ud} at 0.66 and 0.55, respectively, based on preliminary experiments that suggested reasonable trade-offs in Anix's performance and resilience. Parameters R and (UV, UM, UN) were tuned in § 6.2. We fixed t_{pm} at 24 hours after experimenting with values ranging from 1 hour to multiple days (up to 5 days, equaling to no deletion in our T=120 hours shutdown scenario), before converging on a day-long persistence for operational reasons: longer t_{pm} would make phone storage run out faster, and, for larger values of T, force the storage allocated to Anix to be consistently maxed out (consequently prompting for forced message deletion).

Appendix C. More Stringent Adversarial Settings

A. Weeding out misinformation with additional adversarial nodes. Tables 6, 7, and 8 showcase the number of majorly up/downvoted benign and misinformation messages, for scenarios where the proportion of adversarial nodes in the network is increased to 5%, 10%, and 25%, respectively.

We can observe that, in general, the numbers shown in the tables follow the trend observed in our running example

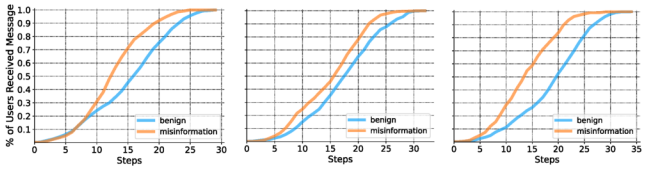
TABLE 5. DESCRIPTION OF EACH PARAMETER (AND RESPECTIVE CONFIGURATION) USED IN ANIX'S MESH NETWORK SIMULATOR.

Parameter / Category	Description	Value	Rationale
A and B	Dimensions of the simulation world ($A \times B$ grid)	25×25	ASMesh's [23] simulation (default)
m	Maximum distance that a user can move in a simulation step	2	ASMesh's [23] simulation (default)
N	Total number of users	600	ASMesh's [23] simulation (default)
β	Connectivity of the network given by the Watts-Strogatz model	0.5	ASMesh's [23] simulation (default)
K	Average social graph degree in the Watts-Strogatz model	15	ASMesh's [23] simulation (adjusted)
T	Total steps of the simulation (1 step = 1 hour)	120	5-day blackout period
Adv	Fraction of adversarial nodes amongst all users	2% – 25%	Increasingly stringent adversarial settings
S	Maximum device storage space allotted to the Anix app	3 GB	Reasonable storage use (128 GB/device total)
P_{inter}	Probability of a given user interacting with the Anix app at any step	0.15	Average daily social media apps usage
C_m	Probability for a user to send out a message in a given step	0.05	Twitter users' posting behaviour
OWT_{ud}	Required ratio of a message's known upvotes/downvotes to OWT the author	0.66	Performance/resilience trade-offs
U_{ud}	Required ratio of a message's known upvotes/downvotes to upvote it	0.55	Performance/resilience trade-offs
R	Ratio of an adversary's friends to benign user's friends	0.1 – 0.9	Tuned during experiments
UV	Probability of a user who has no information about a message to vote on it	0.01 – 0.2	Tuned during experiments
UM	Probability that a user upvotes a message containing misinformation	0.1 – 0.5	Tuned during experiments
UN	Probability that a user upvotes a benign message	0.5 – 0.8	Tuned during experiments
tp_m	Persistence time of a message on a user's device	24h	Performance & operational concerns

throughout §6.2 when the proportion of adversarial nodes is set to 2%. Briefly, a user base that is more aware of an adversary's activities can better distinguish benign from misinformation messages. While our results suggest this task is increasingly difficult for a larger proportion of adversary nodes in the network (where misinformation messages tend to be more leniently upvoted), Anix can still ensure that misinformation messages are usually perceived negatively. In addition, our results suggest that the advantage of an adversary in gaining the trust of benign users remains limited, despite the adversary's increased presence.

B. Coverage and latency for an increased adversary presence (*default* scenario) Figure 2 depicts the coverage and latency of Anix in the default user awareness scenario, while increasing the presence of adversarial nodes in the network to 5%, 10%, and 25% and assuming adversaries choose not to forward benign users' messages. We can observe that, while the presence of additional nodes tends to favor the faster spread of misinformation messages, Anix can tolerate this attack and still ensure the delivery of benign messages to 90% of the mesh in about 25 simulation steps. However, the ratio of adversarial nodes brings a more pronounced effect on latency in the initial stages of a message's propagation. For instance, for benign messages to reach 30% of users under a 2%, 5%, 10%, and 25% adversarial node ratio, they take on average 11, 12, 14 and 16 steps, respectively.

C. Coverage/latency on different scenarios. Figure 3 depicts the coverage and latency of Anix in alternative user awareness scenarios. We see that users who are more aware of the adversary's activities can positively influence the propagation speed of benign messages, when compared to naive counterparts (where we observe the opposite effect).



(a) ($Adv = 0.05$). (b) ($Adv = 0.10$). (c) ($Adv = 0.25$).

Figure 2. Anix's coverage (under active DoS) in function of message delivery latency with increasing proportions of adversarial nodes.

For instance, for the *very aware* scenario, benign messages reach 90% of all users in the mesh after 19 steps, whereas this number grows to 26 steps in the *very naive* scenario.

TABLE 6. ANIX'S RESILIENCE TO MISINFORMATION ($Adv = 0.05$).

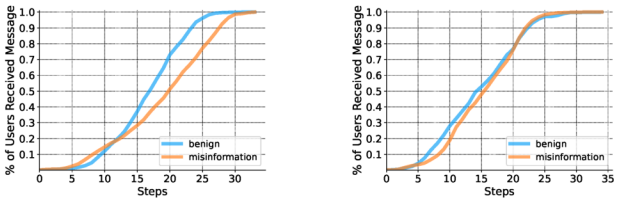
Scenario ($Adv = 0.05$)	Benign		Misinformation		OWTs	
	Upvoted	Downvoted	Upvoted	Downvoted	Benign	Adversarial
Very naive	444	2496	517	2834	45218	353
Naive	1002	1921	121	3227	59541	269
Default	1462	1423	44	3182	59861	185
Aware	2096	761	34	3236	44947	125
Very Aware	2614	285	11	3212	37155	76

TABLE 7. ANIX'S RESILIENCE TO MISINFORMATION ($Adv = 0.10$).

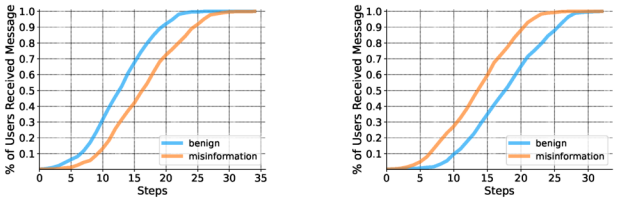
Scenario ($Adv = 0.10$)	Benign		Misinformation		OWTs	
	Upvoted	Downvoted	Upvoted	Downvoted	Benign	Adversarial
Very naive	519	2328	1238	5512	44430	261
Naive	947	1829	192	6522	40319	159
Default	1392	1437	109	6547	32883	73
Aware	2016	726	57	6481	22193	64
Very Aware	2308	347	35	6333	12961	41

TABLE 8. ANIX'S RESILIENCE TO MISINFORMATION ($Adv = 0.25$).

Scenario ($Adv = 0.25$)	Benign		Misinformation		OWTs	
	Upvoted	Downvoted	Upvoted	Downvoted	Benign	Adversarial
Very naive	528	1736	3954	12749	36193	11
Naive	1002	1146	584	16157	15554	2
Default	1269	919	284	16238	6444	1
Aware	1545	527	149	16067	3350	0
Very Aware	1837	285	79	15953	1961	0



(a) Coverage in the aware setting. (b) Naive.



(c) Very aware. (d) Very naive.

Figure 3. Anix's coverage and latency in alternative awareness scenarios.