2010

Toward Anonymity in Delay Tolerant Networks: Threshold Pivot Scheme

Jansen, Rob

Toward Anonymity in Delay Tolerant Networks: Threshold Pivot Scheme

Rob Jansen *
jansen@cs.umn.edu

Robert Beverly *
rbeverly@nps.edu

***Abstract*—Delay Tolerant Networks (DTNs) remove traditional assumptions of end-to-end connectivity, extending network communication to intermittently connected mobile, ad-hoc, and vehicular environments. This work considers anonymity as a vital security primitive for viable military and civilian DTNs. DTNs present new and unique anonymity challenges since we must protect physical location information as mobile nodes with limited topology knowledge naturally mix. We develop a novel Threshold Pivot Scheme (TPS) for DTNs to address these challenges and provide resistance to traffic analysis, source anonymity, and sender-receiver unlinkability. Reply techniques adapted from mix-nets allow for anonymous DTN communication, while secret sharing provides a configurable level of anonymity that enables a balance between security and efficiency. We evaluate TPS via simulation on real-world DTN scenarios to understand its feasibility, performance, and overhead while comparing the provided anonymity against an analytically optimal model.**

I. Introduction

Modern network communication paradigms often assume end-to-end connectivity. While this assumption holds for environments such as the Internet, military and civilian examples of *challenged networks* abound. Disconnected rural communications, "pocket switched networks" [20], mobile, first-responder, vehicular, and ad-hoc networks are all instances of non-traditional challenged networking environments characterized by unreliable resources and intermittent connectivity. Delay Tolerant Networks (DTNs) [3] attempt to provide networking in these challenged environments. DTNs store-and-forward data to provide *eventual* delivery if a contemporaneous end-to-end network path does not exist. While current research addresses DTN routing and naming problems [3], [9], less understood are the unique security issues inherent to DTNs [10]. This work explores adding anonymity and privacy to DTNs.

DTNs are well-suited to an ad-hoc deployment of nodes, such as a military distribution of troops in occupied regions. In military contexts, security and anonymity are mandatory. Battlefield warfare depends on a resistance to traffic analysis to prevent communication from leaking information, and covert military operations and intelligence gathering require the ability to remain anonymous. Anonymity also plays an increasingly important role within non-military engagements including e-governance, citizen journalism, social networking, law enforcement operations, "whistleblowing," telemedicine, vehicular networks, and disaster response scenarios.

* Work performed while authors were part of BBN Technologies' Internetworking Research.

DTNs present unique challenges to designing a secure and efficient anonymous DTN system. Since nodes are often disconnected and opportunistically use links as they become available, traditional approaches to anonymity [4], [7] are not directly applicable to DTNs. The disconnected and dynamic nature of DTN links due to node mobility both constrains and increases the complexity of system design.

Design challenges are accompanied by a new potential for adversarial attacks. An adversary has an advantage in inferring the source of a message when the network topology reduces the size of the anonymity set or prevents traffic mixing. Therefore, DTN anonymity must not only protect the *identity* of communicating nodes, but also their physical *location* in the network. It is non-trivial to develop a system that prevents an adversary from linking communicating nodes or deducing their network location.

We overcome these challenges and develop the Threshold Pivot Scheme (TPS) as a step toward identity and location anonymity in DTNs. Our contributions include:

1) TPS, a system to balance user-specified anonymity requirements versus efficiency in DTNs
2) An analytic framework to help understand ideal anonymity guarantees in a mobile and dynamic DTN
3) Qualitative overhead and anonymity comparison of TPS to DTN-naïve schemes and analytic models

While TPS is a work in progress and makes assumptions about the ability to pre-distribute public keys and identities[1], our hope is to advance the state of *practical* DTN anonymity.

II. Related Work

Traditional approaches to anonymity rely on the concept of mixing [4]. In mix networks, messages are sent along a chain of proxy nodes, called mixes, each of which accumulate and forward source-encrypted messages in batches. Batching messages and layered decryption prevents the correlation of mix inputs with mix outputs while sending a message along a chain of mixes requires only a single honest mix to ensure the message source cannot be linked with its destination. Provably secure mix message formats have been developed to facilitate an anonymous response to an unknown source using *reply blocks* [5], [6], [24], but the disconnected nature of DTNs precludes the pseudonym management required by these schemes.

[1] Such distribution is assumed in many military contexts.

In Tor [7], the most popular deployed mix network, mixing is achieved by layer-encrypting a message at the source and decrypting once at each hop of a source-selected circuit of proxy relays. The last relay sends the unencrypted message to the destination specified by the client. Unfortunately, this source-based routing approach becomes extremely inefficient in DTNs since an opportunistic path to the destination cannot be pre-computed. Anonymity and privacy schemes for ad-hoc networks [2], [17], [22], [25], [31] are similarly incompatible with DTNs since they either employ source-routing or assume a fully connected, available, and reliable network.

Specific to rural DTNs, Kate *et al.* [14] created a complete anonymous rural area system using Identity Based Encryption (IBE). Their resulting scheme increases efficiency and reduces the role of the IBE-required Private Key Generator (PKG). Anonymity is achieved using existing pseudonymous techniques that replace real identities with dynamically generated pseudonyms [13]. However, their system still requires periodic updates from the PKG and relies on the existence of trusted gateways and kiosks to function. Anonymity is thus strictly dependent on these trusted entities, each of which represents a point of failure.

Our TPS approach most closely resembles the Cashmere system, the result of recent research in leveraging Distributed Hash Table (DHT) overlays [32]. Cashmere specifies a set of groups for relaying messages and assumes reachable nodes in the overlay capable of responding for each group. In contrast, DTN nodes have no *a priori* information on which nodes are reachable, or when particular nodes will become reachable. TPS therefore augments many of the notions introduced in Cashmere by permitting any encountered groups to serve as a sequence of relays while providing a per-message configurable level of anonymity.

III. Security Model

This section details our security model as well as assumptions over the adversaries against whom we defend.

Identity Anonymity: A subject is *anonymous* if its identity, or other personally identifiable information, is unknown with absolute certainty [19]. In practice, a subject's anonymity is relative to other indistinguishable subjects in the same system. All such subjects form the *anonymity set*. With only two members in the anonymity set, an adversary should have no greater than a 0.5 chance of assigning an identity. The *degree* of anonymity therefore depends on anonymity set size and probability of de-anonymizing a subject. A message source has *sender anonymity* if it cannot be distinguished from other senders in the system: *receiver anonymity* is analogous. Deployed systems often achieve anonymity through *unlinkability*—sender-receiver pairs cannot be linked as communication partners.

Location Anonymity: Physical location anonymity is a particular concern in DTNs. Their very nature implies that a node may be connected only to a small set of other nodes at any given instant and that the total size and geographic reach of the DTN is small. In such environments, traffic analysis attacks are especially effective as the ability to identify the relative location of a source equates to revealing that source's identity. Whereas Internet-wide anonymity systems can attempt to enforce spatial diversity [28], location anonymity is far more difficult in DTNs.

Adversary: We define our adversary as a local-global adversary in the sense that not only can it control a single DTN node, it also has the ability to passively monitor all communication in the network. Anonymity systems are vulnerable to numerous types of attacks [8], [11], [12], [29]. While we assume the adversary is capable of launching such attacks against our system, we note that traffic analysis remains an open problem for currently deployed anonymity networks [11], [18]. Note that Denial-of-Service attacks are out-of-scope for this work since a DTN adversary may prevent communication by trivially dropping packets at the point of attack.

Key Management: Although key management is a well-known open problem in DTNs, our system will require that a node has access to other nodes' keys. In connected networks, an online trusted Certificate Authority is contacted to verify the authenticity of public keys and validate certificates. DTNs require a unique approach to key management because there is no online entity to which every node is continuously connected.[2]

Henceforth, we assume a DTN deployment of $n > 0$ nodes, $N = \{N_1, N_2, \ldots, N_i, \ldots, N_n\}$. For simplicity and clarity of presentation, we assume each node N_i maintains a public/private key-pair $(PK_i,\ SK_i)$ and has immediate access to all other node's public keys, i.e. $PK_j \forall j \neq i$. However, we note that any *offline* or *DTN* key management scheme, such as [14], may replace this approach to establish the required keys between nodes.

IV. System Design

Not only do common approaches to anonymity fail in DTNs, many adaptations of existing systems are inefficient or impractical. Therefore, a central theme of our system is to provide a practical and configurable level of anonymity to balance security and efficiency in challenged environments. We briefly describe some naïve DTN anonymity schemes and why we reject them in order to better understand the Threshold Pivot Scheme (TPS).

A. Naïve Strategies

Epidemic: The canonical DTN forwarding mechanism, "epidemic" routing [30], provides a natural form of anonymity. DTN messages, called *bundles* [21], are single copy flooded to connected DTN components. Simple anonymity is seemingly provided by epidemically disseminating an encrypted message with a spoofed source address and broadcast destination address. Assume Alice ($\mathcal{A}$) wishes to send the cleartext[3] msg anonymously to Bob ($\mathcal{B}$). The bundle is:

[2] In many military and commercial contexts, pre-placing keys and identities is feasible.

[3] Random values may be added to seal and prevent replay: we omit such details in our discussion for sake of exposition.

$$\beta = (msg|RK)_{PK_{\mathcal{B}}}$$

where $(\cdot)_k$ is encryption with key k.

Each node receiving the bundle will attempt to decrypt the message using its private key to determine whether it is the intended recipient. Any node able to decrypt is implicitly the destination. Return traffic uses the return key RK so that $\mathcal{B}$ can reply without knowing the true identity of $\mathcal{A}$:

$$\beta_{reply} = (msg')_{RK}$$

Epidemic anonymity leverages existing DTN routing protocols and is simple to implement. The true source of the message is difficult to differentiate from intermediary forwarding nodes. However, despite its simplicity, anonymity based on epidemic flooding is inefficient and impractical for all but the smallest networks.

Random Pivot: A second simple scheme introduces a single indirection *pivot* to provide source anonymity and sender-receiver unlinkability. The source $\mathcal{A}$ chooses a pivot node p at random and constructs an encrypted bundle:

$$\beta = ((msg)_{PK_{\mathcal{B}}}|\mathcal{B})_{PK_p}$$

Once β reaches p, it is decrypted and is pivoted towards the destination $\mathcal{B}$. To provide two-way communication, a similar approach is used with additional information included in $\mathcal{A}$'s message. The source message will additionally contain another pivot node, p_2, and $\mathcal{A}$'s address encrypted for p_2. A fresh symmetric return key is included so that $\mathcal{B}$ can encrypt a response for $\mathcal{A}$ without revealing $\mathcal{A}$'s identity to $\mathcal{B}$:

$$\beta' = \left(\left(msg|RK|\left(\mathcal{A}\right)_{PK_{p_2}}|p_2\right)_{PK_{\mathcal{B}}}|\mathcal{B}\right)_{PK_p}$$

$\mathcal{B}$ forwards a reply to p_2, passing along $\mathcal{A}$'s encrypted address in the response. Finally, p_2 will decrypt $\mathcal{A}$'s address and relay the reply to $\mathcal{A}$. $\mathcal{B}$'s reply becomes:

$$\beta_{reply} = (msg')_{RK}\,|\,(\mathcal{A})_{PK_{p_2}}$$

Unlinkability is maintained: the destination $\mathcal{B}$ is unable to discern the bundle source $\mathcal{A}$.

Random pivot is attractive in that the pivot selection is independent of its physical location. The pivot learns no information about the location of the source, and the pivot location leaks no information to the destination. However, pivoting is inefficient in practice. Delays associated with finding and routing to the random pivot node causes unacceptably low message delivery ratios (see §V).

Adapting Tor: Consider applying Tor [7] and other onion routing protocols directly to DTNs. While feasible, this approach is analogous to selecting three random pivots as above since nodes do not know the current state of the dynamic DTN topology. A direct adaptation of Tor is at least as inefficent as the random pivot scheme and therefore similarly impractical.

Group Onions: Consider assigning each DTN node to a group and routing on a "group onion." Instead of requiring a bundle to pass through a pre-specified in-order sequence of nodes, the message passes through nodes belonging to a sequence of groups. However, the order in which groups are encountered is still unknown *a priori*, and presupposing a particular group encounter order is impractical. Allowing for any encounter order is similarly impractical since it requires a number of permutations that is factorial in the number of groups.

B. Threshold Pivot Scheme

TPS improves upon the aforementioned naïve strategies with the intuition of group onions while improving efficiency.

Let $G = \{G_1, G_2, \ldots, G_g\}$ be a set of g anonymity groups, where $1 \leq g \leq n$. Each group G_j contains a set of nodes. Each node N_i belongs to at least one group: $\forall i\ \exists j\ N_i \in G_j$. Each group $G_j \in G$ is assigned a public/private keypair (GPK_j, GSK_j). The assignment of nodes to groups is an important but separable problem which our future research considers. In describing and evaluating TPS, we reasonably assume a random distribution of nodes to groups. In military environments, this process would be akin to pre-placing keys, for instance by assigning various troops or regiments to groups prior to deployment.

Public/Private Keys: Each node N_i maintains a keychain of the following keys:

- A public/private keypair (PK_i, SK_i) and a copy of every other node's public key $PK_j \forall j \neq i$
- ($GPK_j\ \forall G_j \in G$): the public key of every group
- ($GSK_j\ \forall j\ N_i \in G_j$): the private key of groups to which N_i belongs

A node has the necessary group public keys to *encrypt* messages for other groups, and the necessary group private keys to *decrypt* messages for groups of which it is a part.

Secret Sharing: Secret sharing is an established method for distributing trust, i.e. the ability to decrypt among a configurable number of participants. We use secret sharing as a means to accommodate the common case of unknown DTN topology or communication paths. Shamir [23] defines a (τ, s) *threshold* secret sharing scheme in which a secret κ is divided into s shares $S_1, S_2, \ldots, S_\tau, \ldots, S_s$ for $1 \leq \tau \leq s$. Knowledge of any τ shares allows the secret κ to be reconstructed, while knowledge of at most $\tau - 1$ shares reveals no information on κ. Shamir's scheme is *ideal* in that the size of each share does not exceed the size of κ. TPS leverages secret sharing to allow a dynamic routing path without incurring the overhead of group permutations.

Anonymous Messages: Alice wishes to send msg to Bob (herein abbreviated $\mathcal{A}$ and $\mathcal{B}$). TPS uses secret sharing to divide up a key such that the message must go through a configurable number of groups. For each transmission to Bob, Alice generates a one-time (τ, s) threshold secret:

$$\kappa_x = S_1, \ldots, S_s$$

For ease of exposition, assume $s = g$. Let $\pi(i)$ be a bijection to an element in the random permutation of $1, \ldots, s$. From the shared secret, Alice creates a group shared secret that encrypts each share on a group-wise basis:

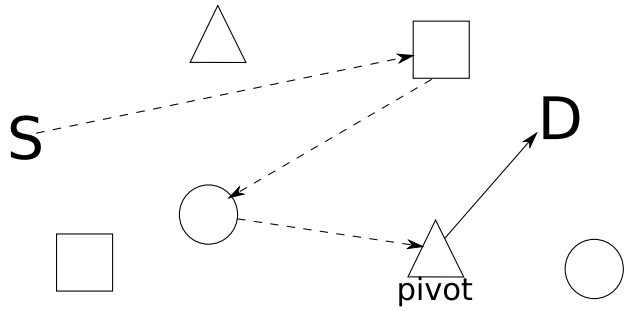

Fig. 1. Threshold Pivot Scheme. Nodes and group membership are represented by shapes. Lines represent logical hops through the DTN. The message source is null, and the destination is only revealed at the pivot node, after being routed through a user-specified τ threshold of s unique groups.

$$GSS_x = (S_1)_{GPK_{\pi(1)}} |f_1| \ldots | (S_s)_{GPK_{\pi(s)}} |f_s$$

Alice encrypts msg with Bob's public key and seals both the message and Bob's address with κ_x. The final standards-based [21], [27] bundle appends the group shared key:

$$\beta = \left((msg)_{PK_{\mathcal{B}}} |\mathcal{B}\right)_{\kappa_x} |GSS_x$$

Decoding and Routing: As shown above, group shared secrets include auxiliary information f for each group-wise encrypted share. This permits a node belonging to a particular group to determine which of S_j it can decrypt and whether those shares have been previously decrypted by another node of the same group. The auxiliary information:

$$f_j = (R_j|d_j)_{GPK_j} |j$$

includes a fixed-length string of random bits R_j, and an indicator bit $d_j \in \{0, 1\}$ denoting whether S_j has been processed and decrypted.

The auxiliary information does not reveal the total number of decrypted shares carried by the bundle, but only provides information on that node's group share. The R_j bits randomize f_j so that an adversary cannot guess the indicator bit and infer the number of decrypted shares (where path length approximates distance from source). Note that we do not prevent an active adversary from inserting bogus shares and flags[4].

As bundle β travels through the network, each node N_i will check f_j for all j such that $N_i \in G_j$. If N_i can decrypt a share (S_j), on the basis of d_j, it decrypts the contents in-place. Otherwise the node forwards β opportunistically, or takes advantage of more intelligent routing protocols, e.g. [16], to increase group coverage as a function of time. For small values of s and τ, nodes may attempt all $\binom{s}{\tau}$ permutations of S_j and either correctly reconstruct κ_x, [5] or know that not enough decoded shares exist (but not the number of decrypted shares). For large values of s and τ, we can limit overhead by using the error-correcting algorithm from Ar *et al.* [1] to reconstruct the Shamir secret polynomial. Once β has passed through τ unique groups, the pivot node can reconstruct κ_x from GSS_x, recover $\mathcal{B}$'s address, and route the msg sealed for $\mathcal{B}$ (see Figure 1).

Anonymous Replies: For forward message anonymity, we require only κ_x. However to permit anonymous replies from $\mathcal{B}$ to $\mathcal{A}$, Alice must generate an additional shared secret κ_y and a one-time symmetric key, K_z. As demonstrated in [5], Alice facilitates replies by including a *reply block* in her message:

$$rb_{\mathcal{A}} = (\mathcal{A})_{\kappa_y} |GSS_y|K_z$$

Alice's bundle then includes the reply block and msg sealed with Bob's public key:

$$\beta' = \left((msg|rb_{\mathcal{A}})_{PK_{\mathcal{B}}} |\mathcal{B}\right)_{\kappa_x} |GSS_x$$

Upon receiving the sealed message and reply block, Bob decrypts both with his private key. He forms his reply bundle by encrypting his reply data with K_z. Bob then appends the remaining parts of $rb_{\mathcal{A}}$ to form and route the reply bundle:

$$\beta_{reply} = (reply)_{K_z} | (\mathcal{A})_{\kappa_y} |GSS_y$$

Thus, Bob can reply to Alice without learning her identity.

We note that, although not presented, TPS requires that each node in the path uses El Gamal encryption to randomize the message and auxiliary information, and a polynomial without a zero coefficient to randomize each share. These randomization techniques are required to prevent tagging and correlation attacks. In other words, every hop needs to re-encrypt the message so the GSS_x value changes at every hop, not only when a share is decrypted. Otherwise the adversary can tag a message, and can gain information if he sees the same message with some bytes modified.

V. Evaluation

As our highest-level goal in TPS is to ensure *practical* DTN anonymity, this section seeks to understand the real-world feasibility of TPS in comparison to what is theoretically optimal. We describe metrics, experiments, and results from TPS simulations in various DTN scenarios.

Analysis: §III divides anonymity along identity and location. TPS conceals identity information by encrypting messages. The pivot should neither learn nor reveal information about the physical location of the source, even if the identity information itself is protected. For example, if the pivot node can reliably determine it is located within two network hops of the source, it can deduce the approximate location of the source and break anonymity.

We can improve geographic indistinguishability by reducing each group to a single node and selecting required groups that can reconstruct κ at random. However, this involves a trade-off between efficiency and security: the more nodes a bundle is forced through, the longer it requires to be delivered but the less likely it will leak information about the location of the source. Notably, the sender of the bundle can choose the number of groups the bundle must pass through by adjusting

[4]Active malicious behavior can be limited by employing group-level integrity signatures and by providing byzantine tolerance via multiple group decodings: such protection is beyond our immediate scope.

[5]A correct reconstruction can be recognized by including a fixed bit-string alongside the destination address.

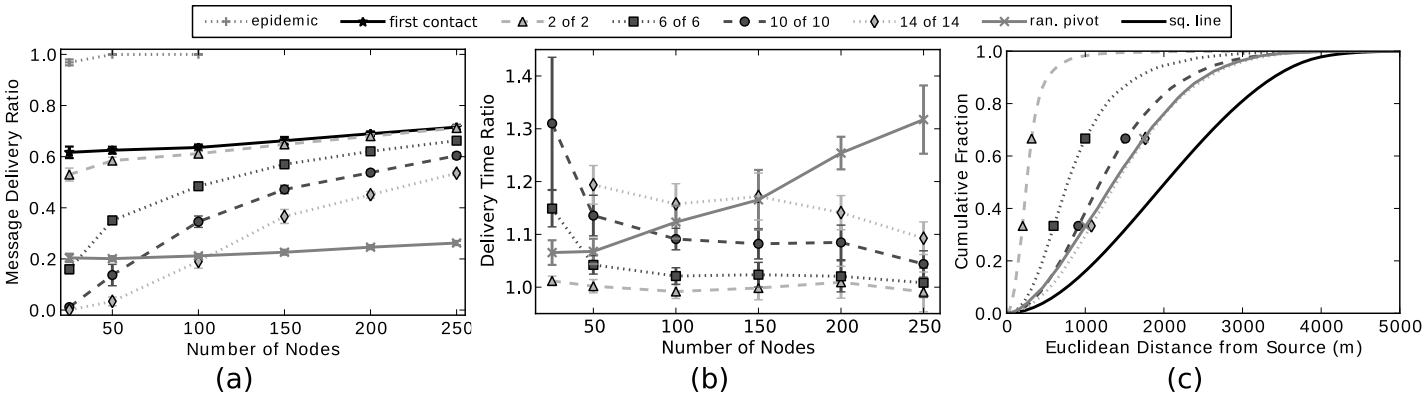

Fig. 2. Comparison of DTN routing strategies. Our TPS scheme uses a threshold equal to the group size of 2, 6, 10, and 14 nodes. All experiments use first-contact routing unless otherwise noted. (a) Message delivery ratio is the number of messages delivered over the number sent. In "first-contact" routing in the ONE simulator and an upper bound for TPS configurations. The inefficient "Epidemic" routing has high delivery rate and is shown for comparison. (b) TPS anonymous to normal message delivery time ratio. (c) Distances between pivot and source: "sq. line" optimally allows no probabilistic advantage in guessing the source.

the threshold parameter τ. For maximum efficiency, $\tau = 1$: the threshold for maximum security is dependent on network configuration and group assignment. We note that since maximum security is achieved when a message has been uniformly mixed among nodes, there is a threshold beyond which increasing τ will have no effect on security and only increase overhead. Individual nodes and networks can select τ values to suit their desired security and performance requirements.

The importance of the relative location of the source and pivot can be captured via a "square line picking" [26] optimal distance metric which models a distribution of distances between any two randomly chosen points on a unit square. Ideally, the node that reconstructs the shared secret (the pivot node) should be equivalent to a uniformly sampled node in the network. Uniformly sampled nodes have no probabilistic advantage in guessing the location of the source. We compare the distances between nodes achieved by methods in §IV to this optimal distance metric.

We use the ONE discrete event simulator [15] to understand TPS. Our experiments vary the assignment of nodes to groups randomly and place nodes at random locations. We used the default world size of 4500 by 3400 meters. Each node has a transmit range of 200 meters and a movement rate of 0.5-1.5 m/s. Our prototype does not perform any cryptography and each node is given an infinite storage buffer so we can focus on protocol performance. Routing in all experiments selects the first node contacted as the next hop – except for epidemic which broadcasts bundles and provides a naïve baseline. We use both random waypoint and map-based movement models (using the default map from the ONE simulator), but present only random waypoint as the results are not quantitatively different. In each experiment, a single message with random (and valid) source and destination addresses is generated every 25-35 seconds. Each experiment lasts 12 simulated hours and is run 10 times with different random seeds. We report the mean results with 95 percent confidence intervals.

Discussion: We show our experimental measurements of message transfer times, delivery ratios, and overhead to quantify the effect of anonymity on these metrics. The ONE simulator is enhanced to allow for the inclusion of anonymous messages and transfer as described in §IV.

The message delivery ratio — the ratio of messages delivered to those sent — allows us to determine the effect of anonymity on delivery capabilities. As we expect, Figure 2a shows a decreasing delivery ratio as the number of groups increases, since it takes longer to find all nodes required to decrypt and route the message to the destination. As the number of nodes in the system increases, the delivery ratio also increases. Since the number of messages created in each experiment is held constant, increasing the number of nodes per group improves message flow and the ability to find nodes for the required groups. However, the random pivot scheme does not realize the same improvement since its group size is constant (one node per group). Figure 2a indicates that larger networks and smaller threshold parameters allow TPS nodes to approach the upper bound for delivery ratio — our $(2, 2)$ threshold anonymity scheme performs nearly identically to non-anonymous routing. Note that messages that have not been delivered when the simulation ends are counted as failed, reducing the delivery ratio.

Figure 2b shows message delivery time overhead induced by anonymity mechanisms in TPS. We measure overhead by running TPS and normal (non-anonymous) routing experiments with identical configuration and recording the differences in delivery times. We show the results as the ratio of TPS to normal message delivery time, noting that we only compute delivery ratios for those messages that are actually delivered in both cases. Anonymous TPS messages take slightly longer to be delivered and we again see an improvement in the ability to find a member of each group with a denser network, producing lower overhead measurements since it is easier to find a member of each group. However, random pivot does not follow the pattern since it is more difficult to find a

particular node at random as the network size increases.

Figure 2c compares the distance between the pivot node and source to the optimal square line picking distance distribution for the 250 node experiment. We find that higher threshold experiments achieve closer to our analytic optimal – pivot nodes' locations reveal less information about the source's relative location. Note that "square line picking" marks the practically optimal distances, due to random node placement, allowed by the simulation.

To summarize our results, we find that the highest threshold value we simulated, the (14,14) configuration, yields the strongest location anonymity but the highest overhead and lowest delivery ratio. Performance, in terms of delivery ratio and overhead, improve as the threshold decreases, but this also leads to weaker location anonymity: the (2,2) configuration provides the weakest degree of anonymity but performs the best. We do not suggest a TPS configuration that works in all networks since the correct configuration depends on the deployed network's conditions and goals, and the user's desired level of security and performance for each message.

VI. Conclusions and Future Work

We described some desirable properties and military motivations for DTN anonymity systems and presented a novel threshold-based approach to anonymous communication for DTNs. We measured delivery ratio, overhead, and distance and compared our anonymous scheme to non-anonymous routing.

One area for future work involves the assignment nodes to groups. This assignment will be vital for preventing information leakage since it determines how far a message must travel until it becomes fully decrypted. We also wish to further explore the use of smarter routing strategies since our experiments currently use first contact routing to avoid disrupting analysis by introducing too many variables during experimentation.

Finally, there are several potential enhancements that we wish to include in our system. In particular, several anonymous packet formats for Mixnets [6], [24] may be non-trivially adaptable to a DTN context, but would strengthen our security guarantees. Other enhancements include node authentication without the need for a trusted third party, redundant transfers to reduce the probability that an adversary can destroy a message of which it has accepted custody, and pairwise cover traffic between a pair of connected nodes to frustrate an attacker's ability to discover real messages sent through the network by simply observing the system.

References

[1] S. Ar, R. Lipton, R. Rubinfeld, and M. Sudan. Reconstructing algebraic functions from mixed data. *Annual IEEE Symposium on Foundations of Computer Science*, 0:503–512, 1992.

[2] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. SDAR: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *IEEE LCN*, pages 618–624, 2004.

[3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838, 2007.

[4] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *CACM*, 4(2), February 1981.

[5] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *the Symposium on Security and Privacy*, 2003.

[6] G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In *the Symposium on Security and Privacy*, pages 269–282, 2009.

[7] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *the USENIX Security Symposium*, 2004.

[8] N. Evans, R. Dingledine, and C. Grothoff. A practical congestion attack on Tor using long paths. In *the USENIX Security Symposium*, pages 33–50, 2009.

[9] K. Fall. A delay-tolerant network architecture for challenged internets. In *the conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34, 2003.

[10] S. Farrell and V. Cahill. Delay-and Disruption-Tolerant Networking, Artech House. *Inc., Norwood, MA*, 2006.

[11] N. Feamster and R. Dingledine. Location diversity in anonymity networks. In *WPES*, 2004.

[12] N. Hopper, E. Y. Vasserman, and E. Chan-Tin. How much anonymity does network latency leak? *ACM Transactions on Information and System Security*, 13(2):1–28, 2010.

[13] D. Huang. Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks. *the International Journal of Security and Networks*, 2(3/4), 2007.

[14] A. Kate, G. Zaverucha, and U. Hengartner. Anonymity and security in delay tolerant networks. In *SecureComm*, 2007.

[15] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools*, 2009.

[16] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[17] K. Mehta, D. Liu, and M. Wright. Location privacy in sensor networks against a global eavesdropper. In *IEEE ICNP*, 2007.

[18] S. Murdoch and P. Zielinski. Sampled traffic analysis by internet-exchange-level adversaries. In *PET*, 2007.

[19] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology. *Version v0*, 27:20, 2006.

[20] N. Sastry, K. Sollins, and J. Crowcroft. Delivery properties of human social networks. In *INFOCOM*, 2009.

[21] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050, 2007.

[22] S. Seys and B. Preneel. ARM: Anonymous routing protocol for mobile ad hoc networks. In *AINA*, volume 2, 2006.

[23] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[24] E. Shimshock, M. Staats, and N. Hopper. Breaking and provably fixing Minx. In *PET*, 2008.

[25] R. Song, L. Korba, and G. Yee. AnonDSR: efficient anonymous dynamic source routing for mobile ad-hoc networks. In *SASN*, 2005.

[26] Square Line Picking. http://mathworld.wolfram.com/SquareLinePicking.html. Accessed August, 2009.

[27] S. Symington, S. Farrell, H. Weiss, and P. Lovell. Bundle security protocol specification. Internet draft, version 15, 2010.

[28] Tor Path Specification. https://git.torproject.org/checkout/tor/master/doc/spec/path-spec.txt. Accessed August, 2009.

[29] A. Tran, N. Hopper, and Y. Kim. Hashing it out in public. In *WPES*, 2009.

[30] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report 200006, Duke University, 2000.

[31] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Mask: Anonymous on-demand routing in mobile ad hoc networks. *IEEE transactions on wireless communications*, 5(9):2376, 2006.

[32] L. Zhuang, F. Zhou, B. Y. Zhao, and A. I. T. Rowstron. Cashmere: Resilient anonymous routing. In *NSDI*, 2005.