Toward Delay Tolerant Network Anonymity: Threshold Pivot Scheme

Rob Jansen*
jansen@cs.umn.edu

Robert Beverly*
rbeverly@nps.edu

Abstract

Research on Delay and Disruption Tolerant Networks (DTNs) challenges the traditional assumption of end-to-end connectivity, extending networked communication to e.g. intermittently connected devices, ad-hoc mobile environments, first-responder disaster scenarios, etc. In such environments, ensuring the security and privacy of both content, networks, and participants is often vital. In this work, we consider DTN anonymity and privacy. The disconnected nature of DTNs presents a unique difficulty for traditional anonymity approaches, namely limited knowledge of other nodes and paths in the dynamic, mobile network. We develop a particular solution, the Threshold Pivot Scheme (TPS), to provide source anonymity and sender-receiver unlinkability in DTNs. Our scheme, based on secret sharing primitives, permits a user-selectable level of anonymity, an important feature for DTN environments that must balance security and usability. Through simulation and analytical analysis, we evaluate the performance and overhead of TPS and find that it addresses the constraints of DTNs while providing a suitably high-level of anonymity.

1 Introduction

Modern network communication often assumes immediate and reliable end-to-end connectivity. This assumption is true in environments such as the Internet, however, there is a large class of networks, so-called *challenged networks*, where it does not hold. Examples of such networks include media, sensor, and satellite networks, and mobile, vehicular, first-responder, and military ad-hoc networks. These networks are characterized by limited resources, intermittent connectivity, and potentially long delays and low data rates. Interest in these challenged environments has fueled research in Delay and Disruption Tolerant Network (DTN) architectures [14]. DTNs facilitate intelligent store-and-forward behavior to provide *eventual* data delivery when a contemporaneous end-to-end path does not exist in the network. Design and development of DTN protocols is currently in progress ([5, 44] and references therein). While developing operational systems and demonstrating feasibility has been a primary goal of DTN research, the security of DTNs is beginning to receive attention. The foundations for basic security have been outlined, [16, 52] including data origin authentication, integrity, and confidentiality primitives. However, DTNs present new and unique security challenges as a result of limited resources and lack of end-to-end connectivity. Farrell *et al.* argue [15] that key management, traffic analysis, policy enforcement, and node introduction are the core challenges in DTNs; we argue for adding anonymity to this list.

We logically divide DTN anonymity into two categories, *identity* and *location*. Identity anonymity implies that the identity of a traffic source is hidden to all other nodes, including the traffic recipient. Location anonymity concerns the discovery of, or advancement in, knowledge of geographic location from information leaked in messages.

Many examples have been advanced in literature and practice [55] where anonymity is valuable, and often invaluable. Some examples include: online journaling and blogging for people visiting, working, or living in a country that censors or blocks traffic, law enforcement operations involving monitoring criminal activity in which success depends on stealth, and human rights workers or "whistleblowers" who might otherwise resist speaking out in fear of physical attacks or threats from those with opposing viewpoints or motives. Anonymity prevents these types of actions from being linked to an identity on a network. Existing DTN applications which fall into the above usage scenarios and requirement for anonymity include blogging [38], web serving and surfing [39], electronic mail [23], and interactive voice messaging [24, 25].

DTN-specific environments yield additional uses of anonymity, e.g. social networking applications on intermittently connected mobile devices, military intelligence gathering, voting, etc. in under-developed regions, and collecting aggregate medical data in first-responder disaster scenarios. The continued development and maturity of DTNs and its transfer agent, the Bundle Protocol [44], will likely lead to an increasing number of applications and opportunities for anonymity.

This work therefore focuses on DTN anonymity and

[*] Work performed while part of BBN Technologies' Internetworking Research department.

privacy. The disconnected nature of DTNs presents a unique difficulty for traditional anonymity approaches, namely limited knowledge of other nodes and paths in potentially dynamic, mobile, and resource constrained networks. To address these challenges, we develop a particular solution, the Threshold Pivot Scheme (TPS), to providing source anonymity and sender-receiver unlinkability in DTNs. Our primary contributions include:

1. Development of TPS, using standard and well-accepted security primitives, to provide source anonymity and sender-receiver unlinkability in DTNs.

2. Qualitative comparison of TPS to DTN-naïve anonymity schemes.

3. A basic analytic framework for understanding ideal anonymity guarantees for a mobile and dynamic DTN.

4. Simulation results from a mobile DTN network including anonymity performance and overhead for various network sizes of 25 to 250 nodes.

The remainder of this paper is outlined as follows. §2 reviews related work upon which we draw, while §3 details our security model, assumptions, and design goals. We present the design of TPS and comparisons to other anonymizing approaches in §4. Our analytic and simulation results are in §5 and we conclude with a discussion of major findings and future work in §6.

2 Related Work

The first method for anonymous electronic communication, based on *mixing*, was introduced by Chaum *et al.* [7, 8]. A significant amount of subsequent research has explored enhancements to Chaum's mixnets. For instance, a more efficient mixing system is Crowds [43], which relies on the notion of "blending into a crowd". Requests are forwarded at random so that no member of the crowd can determine with absolute certainty the origin of a request.

Existing and deployed IP-based systems similarly rely on the principles of mixing. One of the most popular, Tor [11], is an *Onion Routing* network where clients encrypt traffic in several overlapping layers to produce an "onion." The i'th layer of the onion is encrypted with public key of the i'th node through which the message must pass. Layer i encapsulates layer $i-1$, etc. The onion is sent through a *circuit*, or collection of nodes acting as message routers. Each router decrypts its layer of the onion and the last router sends the traffic to the destination specified by the client.

Unfortunately, mixnets in general, and Tor in particular, rely on source-based routing. Nodes must have knowledge of both the network membership and topology to create circuits. Thus, mixnets and onion routing are not directly applicable in a DTN context. We discuss how traditional onion routing may be modified to work in a DTN environment in §4.

There is extensive research on anonymity for wireless ad-hoc networks. Many proposed schemes [59, 51, 47, 32, 49] are based on dynamic source routing [27, 28, 29], which is not practical in DTNs since in many cases nodes have limited topological information and no immediate knowledge of the path to the destination. Moreover, even non-source routing-based schemes e.g. [4, 9] assume a level of connectivity not guaranteed in a DTN environment where nodes may be disconnected for long durations and never obtain a contemporaneous end-to-end path to the final message recipient.

Proposed methods for location privacy in ad-hoc networks [10, 34, 57, 26, 12] assume the use of anonymous routing schemes [4, 58] to prevent neighbors from learning each others identities. However, these schemes similarly rely on the network being fully connected, available, and reliable. In contrast, we are interested in a system that is fully functional even in a delay and disruption prone environment.

Key management is a well-known known open problem in DTNs. Traditional approaches for public key infrastructure (PKI) verify communication partners via public key certificates and negotiate a shared key for communication. However, PKI assumes an *online* entity is available for immediate verification of the public key certificate and certificate revocation lists. In lieu of a PKI, Seth *et al.* [46] propose the use of identity based cryptography (IBC) for DTN key management using a hierarchical based IBC scheme [18]. However, IBC still requires an online trusted private key generator (PKG) and contact with the PKG is needed for public parameter updates and private key rotations. Moreover, Farrell and Cahill explain that IBC DTNs are not scalable as public parameters must be kept for all PKGs [15]. Asokan *et al.* [1] discuss the trade-offs between PKI and IBC in DTNs.

A complete anonymous rural area DTN system using IBC was created by Kate *et al.* [30] which reduced the number of public parameters required for each node. Their resulting scheme increases efficiency and reduces the role of the PKG. Anonymity is achieved using existing pseudonymous techniques that replace real identities with dynamically generated pseudonyms [59, 22, 9]. While their technique is an improvement over previous schemes, it still requires periodic updates from the PKG and is specific to rural environments. In particular, the scheme relies on the existence of trusted gateways and kiosks for the system to function. The anonymity provided is strictly dependent on these trusted entities, each of which represents a single point of failure.

Our approach most closely resembles the Cashmere system, the result of recent research in leveraging Distributed Hash Table (DHT) overlays [60]. Cashmere similarly espouses the notion of relay groups, however their objective is to provide more resilient anonymous routing. Cashmere specifies a set of groups for relaying mixing and assumes reachable nodes in the overlay capable of responding for each group. In contrast, DHT nodes have no a priori information on which nodes are reachable, or when particular nodes will become reachable. TSP 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.

3 Security Model

This section details our security and anonymity model as well as assumptions over the classes of adversaries we defend against. The following is inspired by the discussion of anonymity in [40]. A general overview of the DTN architecture [5, 44] and available DTN security mechanisms, namely the Bundle Security Protocol (BSP) [16, 52], can be found in appendix A. However, anonymity is not currently provided in these DTN design standards.

3.1 Anonymity

Identity Anonymity: The *anonymity* of a subject is the property that the subject's identity, or other personally identifiable information, is not known. A subject's anonymity in an anonymous system depends on the existence of several other indistinguishable subjects performing similar actions in the same system. All subjects in the anonymous system form the anonymity set. The level of anonymity provided by the system is stronger with a larger anonymity set. The type of anonymity provided is largely dependent on system design.

Sender anonymity is provided through a disjoint anonymity set only composed of those entities who send data in the system. A sender is then anonymous if it can not be distinguished from other senders. Similarly, *receiver anonymity* is the property that receivers of data in a system remain anonymous in the set of other receivers. Although it is conceivable for a system to provide both sender and receiver anonymity, the quality of anonymity achieved depends on the quantity provided (size of anonymity sets and probability of de-anonymizing a subject) and the robustness to stronger adversaries.

Perfect anonymity, the absolute highest level of anonymity, permits no probabilistic advantage over a random guess to identify a user. Perfect anonymity is thought to be impossible on the Internet because IP addresses identify a subject to a first-order and data is sent over managed network links, i.e. data is forwarded by other network entities, at least one of which is aware of its origin. However, lower levels of anonymity, which still provide a highly usable system in practice, are achieved through unlinkability, unobservability, and pseudonymity.

Location Anonymity: In addition to typical concerns of identity anonymity, we must be particularly concerned with physical location anonymity. The very nature of DTNs implies that a node may be connected only to a small set of other nodes at any given instant in time or that the total size and geographic reach of the DTN is small. In such environments, traffic analysis attacks are particularly effective as the ability to identify the relative location of a source is equivalent to revealing that source's identity. Tor, a production onion routing overlay on the Internet, handles location anonymity by ensuring each path contains no more than a single Tor relay in any given IPv4 subnet of size 2^{16} [54]. Although it is possible for tier-1 ISPs and Internet exchanges to perform traffic analysis [36, 17], this selection criteria, based on the hierarchical nature of IP address assignment, suffices to provide sufficient geographic node distribution and path disparity.

In contrast, location anonymity is much less trivial in ad-hoc or DTN deployments, but still of extreme importance. A significant application of DTN location anonymity applies to soldiers in a military environment, e.g. who have been issued DTN devices for communication. Field agents and soldiers may require anonymity to prevent adversaries from discovering their location or that of their military base camps, e.g. by linking them to the military servers to which they are connecting. Without anonymity, it is much easier to discover geographic location, e.g. by promiscuously listening on wireless links and watching patterns of traffic flow to and from a battlefield commander. This puts military objectives at risk to compromise and allows enemies to single out commanders or base camps as targets for attacks.

Unlinkability: When data sent in a system can not be linked to data received, it is said to be *unlinkable*. Unlinkability provides *relationship anonymity*; it is unknown who communicates with whom. Relationship anonymity does not hide information about a source sending a message or a sink receiving one, rather, that a sent message can not be linked to any message received and vice versa. The probability of an adversary linking two messages does not increase by continually observing the system. Relationship anonymity is weaker than both sender and receiver anonymity since messages can be traced to specific subjects. However, unlinkability can also provide sender (receiver) anonymity if any sent (received) message can not be linked to its sender (receiver). The exact properties achieved by a system is dependent on its design.

Unobservability: System communication is *unobservable* if it is indistinguishable from no communication. *Sender unobservability* is achieved when senders sending messages are indistinguishable from those that do not; *receiver unobservability* is parallel. *Relationship unobservability* refers to the indistinguishability of the transfer of a message between any possible sender-receiver pair. In general, unobservability implies anonymity and is stronger than unlinkability. However, it is most often not employed in production anonymous systems because it requires constant cover traffic and message broadcasting which can drastically decrease communication efficiency.

3.2 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. It has been shown that anonymous systems are vulnerable to traffic analysis [42, 36, 17, 2, 45, 20], circuit clogging [35, 13, 33, 21], and relay selection and circuit extension attacks [56]. While we assume the adversary is capable of launching such attacks against our system, we note that traffic analysis remains an open problem for current Internet anonymity [36, 17].

To minimize potential attacks, we leverage existing and well-understood cryptographic primitives – encryption, decryption, and digital signatures – with exponential hardness guarantees, i.e. secure operations that adversaries cannot realistically subvert. Additionally, we can employ standard techniques such as node authentication, periodic cover traffic, and redundant transfers. These techniques are discussed further in §5.4.

We assume that Denial-of-Service (DoS) attacks are out-of-scope for this work. In particular, an adversary may drop packets at the point of attack, and custody transfers (see appendix A) do not particularly aid in minimizing this effect since the adversary could drop packets after receiving custody. Further, it would be difficult to locate the attacker in a DoS attack since we expect all nodes to be authenticated.

3.3 Key Management

Key management is a well known open problem in DTNs, however, our anonymity systems will require a key management scheme to establish efficient means for anonymous communication. In connected networks, an online trusted Certificate Authority is contacted to verify the authenticity of public keys and validity of the corresponding certificates. DTNs require a unique approach to key management because there is no online entity to which every node is continuously connected. Sensor network proposals such as [6] seem promising, however, they do not work particularly well in DTNs since they too assume a connected graph. We note that any *offline* or *DTN* key management scheme can be used to establish the required keys among group members, e.g. the HIBC scheme from [30] could be adapted to our systems. Additionally, any new schemes that result from research advances could also replace our construction. For simplicity and clarity of presentation, we assume a PKI where public key certificates are trusted and therefore on-line verification is not required.

Henceforth, we assume a DTN deployment of $n > 0$ nodes, $N = \{N_1, N_2, \ldots, N_i, \ldots, N_n\}$. Each node N_i will maintain a public/private key-pair (PK_i, SK_i). We further assume that each node N_i has immediate access to all other node's public keys, i.e. $PK_j \forall j \neq i$.

3.4 Design Goals

We first define our design goals for an anonymous DTN system as these directly influence the security model. We note that these goals are ambitious, even for connected networks such as the Internet given the possible attacks discussed in §3.2.

1. ***Anonymous:*** source anonymity and sender-receiver unlinkablility
2. ***Secure:*** resilient to powerful adversarial attacks
3. ***Practical:*** does not add significant overhead relative to existing DTN routing protocols
4. ***Usable:*** empowers users to choose per-message performance vs. anonymity parameters

4 System Design

This section provides an overview and basic concepts associated with our anonymous systems. We then describe three approaches to an anonymous bundle transfer system for DTNs. Each of the three approaches has different merits and drawbacks that explore tradeoffs in security and efficiency. The first two approaches are perhaps most interesting in illustrating that seemingly simple techniques and adaptations on existing anonymity systems are impractical for DTNs.

Common to the schemes we introduce is onion routing. As in other onion routing or mixnet systems, the bundle traverses multiple network hops in encrypted form before it is possible to decode the true destination. The last node on the encrypted path we designate the *pivot*[1] node. This

[1] The pivot node "pivots" the bundle from an encrypted state floating through the network to a decrypted state routed towards the destination.

node is capable of removing any remaining encryption layers and forwards the bundle to the destination.

Groups: A distinguishing feature of our approaches is the use of anonymity "groups." Let $G = \{G_1, G_2, \ldots, G_j, \ldots, G_g\}$ be a set of $1 \leq g \leq n$ groups. Each group G_j, for $1 \leq j \leq g$, potentially contains a set of nodes and each node N_i, for $1 \leq i \leq n$, is in at least one group, i.e. $\forall i \; \exists j \; N_i \in G_j$. Based on the aforementioned key management assumptions, a group public/private key-pair $(GPK_j,\ GSK_j)$ is generated for each group $G_j \in G$.

Keychain: Each node N_i maintains a "keychain" denoted $keychain_i$ composed of the following keys:

1. Every node has its own public/private key-pair and a copy of every other node's public key:

$$\forall i \; \exists (PK_i,\ SK_i) \in keychain_i$$

$$\forall i, \forall j \neq i \; \exists PK_j \in keychain_i$$

2. Every node is given a copy of the public key of every group to allow nodes to *encrypt* messages to any other group.

$$\forall i \; \forall j,\ GPK_j \in keychain_i$$

3. Every node is given a copy of the private key of each group to which it belongs to allow nodes to *decrypt* messages specified for its group.

$$\forall j \; \forall i \; s.t. \; N_i \in G_j,\ GSK_j \in keychain_i$$

Symmetric One-Time Keys: As will become evident, public keys are not used to secure the bundle payload. Instead, a fresh, one-time, per-bundle symmetric key is generated and used to encrypt the bundle. This symmetric key is protected using various public keys, a technique we employ for performance, efficiency, and security reasons. The fresh symmetric key prevents any pair of encrypted messages from appearing identical, even if the same message is sent multiple times.

Secret Sharing: Secret sharing, originally developed independently by Shamir [48] and Blakley [3], is a well-established method for distributing trust, i.e. the ability to decrypt, among a configurable number of participants. Shamir defines a τ of 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 $0 \leq \tau \leq s$. In Shamir's scheme, knowledge of at least any τ shares allows the secret κ to be reconstructed while knowledge of at most any $\tau - 1$ shares reveals absolutely no information about the secret κ. Shamir's scheme is also *ideal* in the sense that the size of each share does not exceed the size of the secret κ.

Secret sharing has been extending in many ways, including catching cheaters [53], verifying shares [41], verifying shares non-interactively [37], and proactively updating compromised or corrupt shares [19]. In contrast, our primary use of secret sharing is as a means to enable anonymity in DTNs which can operate without knowledge of the network topology or bundle communication path.

4.1 Epidemic

This first scheme we describe, and then reject for efficiency reasons, takes advantage of existing DTN "epidemic" routing mechanisms. Epidemic routing are essentially flooding protocols, but ensure that only a single copy of each bundle is exchanged between any pair of DTN nodes.

In the spirit of epidemic routing, we simply remove the source and destination address information[2] and epidemically disseminate the bundle to all network nodes. Assume node a wishes to communicate anonymously with node b by sending $data$. The bundle payload contains:

$$payload_{a \rightsquigarrow b} = PK_b(b, data, RK)$$

Each node receiving the bundle will use its private key to decrypt the message and determine whether it is the intended recipient. Return traffic uses the return key RK so that b can reply without knowing the true identity of a:

$$payload_{b \rightsquigarrow a} = RK(a, data)$$

As shown in Figure 1, only the intended recipient is capable of decoding the bundle.

This epidemic scheme leverages existing DTN routing protocols, does not require groups or group information, and is simple to implement. Other nodes' public keys are necessary only so that the source can confidentially send a message to the destination (and for cover traffic, described later). Additionally, the true source of the message remains hidden as its address is never contained in the message and it is difficult to differentiate a source from a node acting as an intermediate forwarder.

An obvious drawback of this approach is inefficiency. Traditional epidemic routing can employ message "pruning" to partially mitigate the inefficiency of flooding – when the destination receives a message, it notifies other nodes to stop unnecessary forwarding. However, pruning by definition reveals the destination of a message, breaking anonymity. Finally, because pruning improves the overhead of flooding only in select DTN scenarios, epidemic dissemination is impractical for any sufficiently large and practical DTN. Also note that this approach does not meet our usability goal as the user has no control over the performance/security setting.

[2] Equivalently, a broadcast destination address.

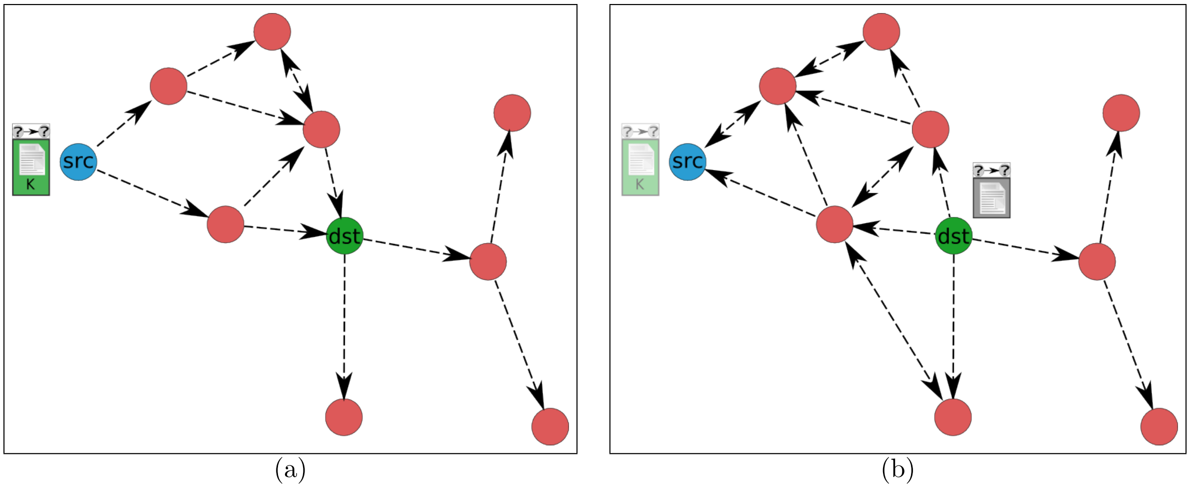


Figure 1: Epidemic scheme. Only the source knows it sent a message. (a) source broadcasts a message (b) destination broadcasts a response

4.2 Random Pivot Scheme

This scheme introduces a single indirection point introduced above as a *pivot*. This scheme provides source anonymity and sender-receiver unlinkablility, although as in the epidemic scheme, it does not require group information to function. The source node chooses a node, called the pivot node, randomly from the available nodes in the network. The source node then constructs a bundle B_κ that does not contain source information. This method can be considered a one-of-one secret sharing scheme, but we note that the same functionality can be obtained using only the configured PKI. The source node constructs a bundle that encapsulates B_κ as described above and sends it out into the network. Once the bundle reaches a node from the pivot group, B_κ is decrypted and B is pivoted towards the destination. Figure 2 shows the simple version of this scheme, which only provides one-way communication.

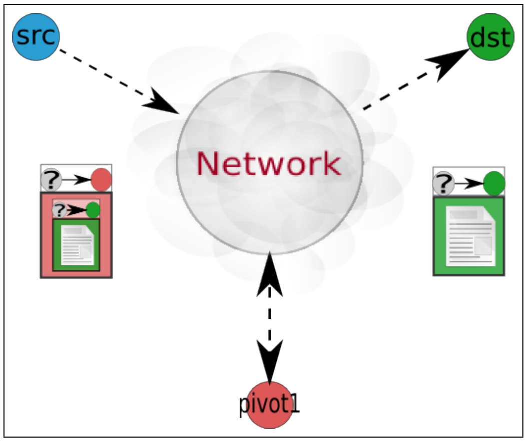


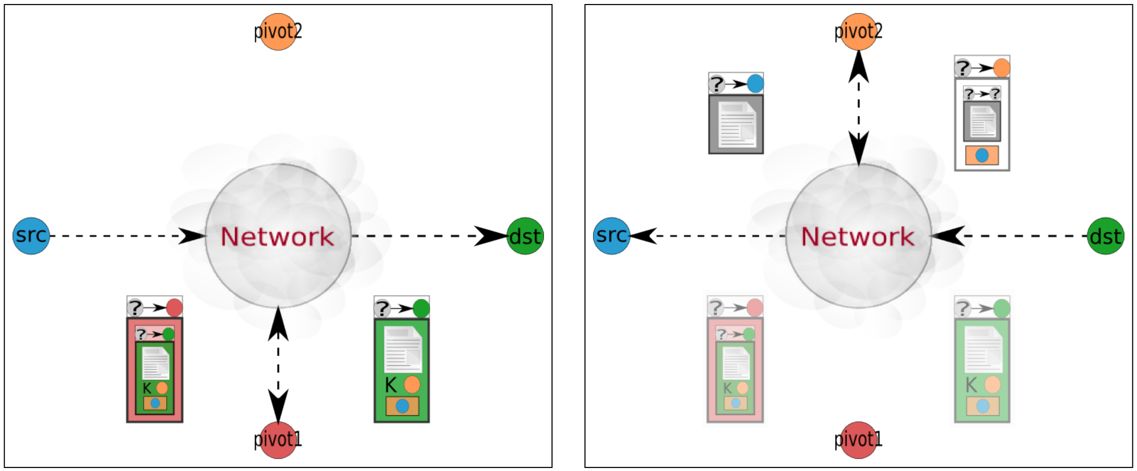
Figure 2: Random pivot scheme. A random node is chosen as the pivot to the destination.

In order to provide two-way communication, a similar approach is used with additional information included in the bundle (see figure 3). The source node must generate an additional pivot node to which the destination will send the response (pivot2). This second pivot will also receive the sources identity (from the source node, through the destination node) so it can return the response to the correct node. A fresh symmetric key is also included so that the response can be encrypted for the source without revealing its identity to the destination. This way, unlinkability is maintained; the destination will not be able to discern exactly who sourced the bundle.

This scheme does not meet our goal of usability – the user has no control over the level of security it obtains. It also has the practical drawback of being less efficient since the pivot node would be located at the mean distance from the source. These delays associated with finding and routing to a random node are forced on the user. On the other hand, this scheme reduces location anonymity problems since it is just as likely that *any* node in the network is chosen as the pivot node regardless of its geographical location. This means that the pivot node will not learn any information about the location of the source.

4.3 Adapting Tor

Consider adapting Tor [11] and other onion routing protocols directly to DTNs. While feasible, Tor-like schemes require a source-specified path, or circuit; information that the source cannot be assumed to know. If a source



(a)

(b)

Figure 3: Random pivot scheme – two-way communication. The source includes cryptographic material to enable the destination to send a response while maintaining anonymous properties. (a) The source sends a message through random pivot1. (b) The destination returns the encrypted response through random pivot2. The encryption key and pivot2 was supplied by the source.

chooses random indirection nodes for the circuit, the message could take a *long* time to reach each node in the specified order.

Instead, we make use of groups. Rather than using source routing, the source creates a group onion. In this way, any node belonging to group can decrypt that group's layer of encryption. As we will see, this scheme reduces the the number of nodes through which a message must travel and increases efficiency. Instead of sending the message through the g specific nodes that form the circuit (g is configurable), it now must only travel through one node from each of g groups.

The order in which these groups will be encountered is still unknown, and assuming a particular group encounter order is impractical. Therefore, the message is replicated to create an "onion" for every possible permutation of the g groups desired for the "circuit."

Once a node meets a potential next-hop node of a different group through its natural mobility, it forwards only the onions in which the next-hop node is able to decrypt (i.e. those with an outer encryption layer that the next-hop node can remove, based on keys in its keychain). Figure 4 shows an example of the group onion scheme.

This naïve method requires $g!$ message replications of the bundle payload for a g group circuit, causing increased computation and storage requirements, increased communication, and an increase in "wasted" traffic – the permutations that are not forwarded to the next-hop are discarded. Without requiring each node to keep state information about each transferred message and its previous and next hop, two way communication is not possible (our practicality goal). This state information is necessary so that the same path can be used for the response as was used for the request, preventing the source from giving out its identity to the destination and losing its anonymity.

A critical consideration is that physical location information is leaked to the pivot node – the point where the onion becomes decrypted. Assuming nodes in the net-

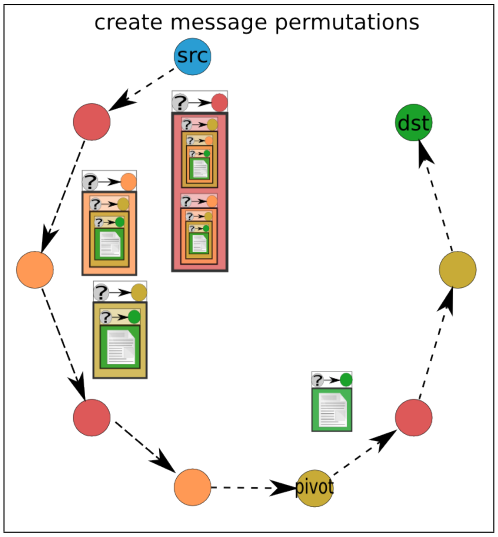


Figure 4: Group onion scheme. Only those permutations of onions which are forwarded are shown. The number of message replications is $g!$ for g groups.

work are uniformly distributed to groups and since the message only needs to travel through g groups, in many environments the pivot node will be located close to the true source. This breaks our security goal as an adversary could use this information to discover the source's relative location.

4.4 Threshold Pivot Scheme

To address the efficiency and robustness issues of the previous two anonymity schemes, and to provide configurable anonymity levels, we introduce the Threshold Pivot Scheme (TPS). TPS leverages Shamir's secret sharing [48], splitting a secret κ into s shares of which τ are required for the reconstruction of κ. $\tau - 1$ shares reveals no information about κ and the size of each share does not exceed the size of κ. Shamir's scheme gracefully allows us to use the existing key infrastructure to require input from multiple entities without forcing a source-specified order.

Initializing Share Lists: Let κ be a unique, per-bundle symmetric key. The source node generates κ and specifies a threshold parameter τ and total number of shares s where $1 \leq \tau \leq s$. Let M be a well-known "magic cookie." Let $a|b$ denote the concatenation of a and b. The resulting s shares, denoted $S_1, \ldots, S_s$, encode $\kappa|M$.

Each share is encrypted with the public key of a particular group. For ease of exposition, we assume $s = g$ and each group is allowed one share. Each share S_j is encrypted thusly:

$$\sigma_j = GPK_j(S_j)$$

In addition, we maintain a per-share "flag" list f_j. Let $d_j \in \{0, 1\}$ be an indicator variable that denotes whether σ_j has been processed by G_j and decoded. Let R_j be a string of random bits. Then:

$$f_j = GPK_j(R_j|d_j)|j$$

The flag list permits a node belonging to a particular group to determine which of the σ_j it can possibly decrypt and further whether σ_j has been decrypted previously by another node of the same group.

The flag list does not reveal the total number of decrypted shares carried by the bundle, but rather only provides information on that node's group share. The R_j bits seal the flag so that an adversary cannot guess the indicator bit in order to infer the number of decrypted shares. This relies on $GPK_j(S_j)$ and S_j being indistinguishable. We leave this indistinguishability problem to future work.

Forming the Anonymous Bundle: To create an anonymous bundle, we alter the original DTN bundle B as follows. The original bundle is encrypted for the destination node dst using key PK_{dst} and BSP [52]. It is then formatted according to the DTN Bundle Protocol (BP) specification [44] except we remove the source information from the BP source fields. The key κ serves as our *bundle encryption key.* The entire contents of the original DTN bundle, including payload and all headers, are encrypted with κ using AES symmetric key encryption.

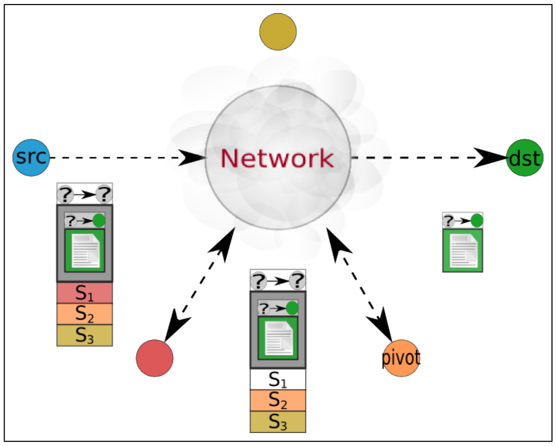


Figure 5: Threshold scheme ($\tau = 2$). The message is forwarded until 2 shares can reconstruct κ and decrypt and route the destination message.

The resulting encrypted data B_κ is combined with the share information to form a new bundle:

$$\beta = \{f_1|\sigma_1|f_2|\sigma_2|\cdots|f_s|\sigma_s|B_\kappa\}$$

and called the *anonymous bundle.* Note that anonymizing a bundle in this manner may be done by any user of the system. β obfuscates the original bundle contents and the bundle source can repudiate creation of the bundle.

Network Traversal: As β travels through the network, each hop will check the list f_j and decrypt a share σ_j if it can, replacing σ_j with the decrypted contents. To find which share to decrypt, the node will check each f_j for its group id j and decrypt using GPK_j to find if the flag d_j has been set. If d_j has been set, it will additionally decrypt σ_j, otherwise it will forward the message.

Bundle Delivery: The pivot node in this scheme is the node whose decrypted σ_j allows for the complete reconstruction of κ, i.e. the pivot node is a member of the τth of g groups required to reconstruct κ. The pivot node combines τ shares to reconstruct κ. Recall from [48] that the shares are constructed in such a way that no information is gained when $\tau - 1$ shares are known, i.e. it is no easier for any adversary to compute κ given $\tau - 1$ shares as it is with one share. Then, κ is used to decrypt B_κ, and obtaining the original bundle B which contains destination information required for bundle delivery. The pivot node then routes B to the final destination as a normal DTN message (see figure 5). Note that B could have additional security mechanisms applied by the BSP [52] (see appendix A.4).

The threshold pivot scheme has similar drawbacks as the Tor group adaption scheme in regards to two-way communication and the source being geographically distinguishable to the location of the pivot node. We can improve geographic indistinguishability by reducing each group to a single node and selecting required groups, hence nodes, that can reconstruct κ at random. However, this involves a trade-off between efficiency and security; the more nodes you force the bundle through, the longer it will take to be delivered but the less likely it will leak information about the location of the source. In this scheme, however, the novel idea is that the sender of the bundle can choose the number of groups the bundle must pass through by adjusting the threshold parameter τ. Thus, usability is natural. For highest security $\tau = g$, and for maximum efficiency $\tau = 1$. Each separate DTN deployment can decide on the threshold parameter based on the characteristics of the DTN and the desired security and performance of the bundle transmission. Other clear benefits are that message replication and path ordering are not required.

5 Results

This section describes metrics, experiments, and results from simulations of TPS in various realistic DTN scenarios. Where possible, we compare analytic and simulation results to understand the practical impact of these different DTN mobility and connectivity models.

5.1 Analysis

In §1 we divide anonymity into two main categories: identity and location. Identity anonymity is achieved through cryptographic security mechanisms such as encryption. Location anonymity additionally depends on the physical location of the pivot node in relation to the source node. We want to ensure that no additional information about the source is leaked to the pivot node that would enable it to deduce the source's identity. For example, if the pivot node was always within 2 hops of the source, it would be able to reliably deduce the approximate location of the source and break anonymity.

The importance of the relative location of the source and pivot has lead us to consider an optimal distance metric. In a best case scenario, we would want the pivot node to be a random node in the network. In other words, we want it to be equally likely that the pivot node is any node in the network to guarantee that it does not learn anything about the location of the source. In our experiments, we measure the distance from the pivot node to the source node.

To model an optimal distance measurement, we use Square Line Picking [50]. This model provides a distribution of distance between any two randomly chosen points in a unit square. The distribution function is given in table 1. This distribution provides us with an optimal measurement of the distance between two nodes which we can use as a comparison to the distance achieved by our schemes.

5.2 The ONE Simulator

All simulations and measurements were captured with The ONE Simulator [31]. Our experiments used a varying number of nodes each assigned to groups randomly and at random locations. Our world size was the default 4500x3400 meters. Each node has a transmit range of 200 meters and a transfer rate of 0.5-1.5 meters per second. Each node was given an infinite storage buffer to prevent messages from getting dropped, disrupting our measurements. All experiments used first contact routing except for one epidemic experiment which we ran for a crude comparison. All experiments used either random waypoint or map-based movement models. The results for map-based movement models were similar to random waypoint and are not presented. In each experiment, messages are randomly generated and sent every 25-35 seconds. Each simulation was run for 12 simulated hours with 10 random seeds (95 percent confidence intervals shown).

5.3 Discussion

Our experiments test message transfer times and how our anonymity scheme affects those times. The ONE was enhanced to allow for the inclusion of anonymous messages as per our system design. Anonymous message transfer was carried out as described in §4, however, we did not perform any cryptographic operations or key generation. In non-anonymous mode, messages are transferred to the destination with no changes to the implementation of the routing protocol in use. In non-anonymous mode, message creation and message delivery timestamps were recorded. Anonymous mode included the addition of the decryption timestamp. These timestamps were used to determine the message delivery ratio and overhead. We also record the distance between the pivot node and the source node for those messages that were decrypted during the experiment.

Delivery Ratio: An important metric we wish to measure is the bundle delivery ratio, the rate at which the system is capable of delivering messages. The importance of this metric is that is allows us to determine how anonymity affects overall message delivery capabilities by comparing it with non-anonymous delivery rates.

$$D(l) = \begin{cases} \frac{1}{2}t^4 - \frac{8}{3}t^3 + \pi t^2 & \text{for } 0 \leq l \leq 1 \\ -\frac{1}{2}t^4 - 4t^2 tan^{-1}(\sqrt{t^2-1}) + \frac{4}{3}(2t^2+1)\sqrt{t^2-1} + (\pi-2)t^2 + \frac{1}{3} & \text{for } 1 \leq l \leq \sqrt{2} \end{cases} \quad (1)$$

Table 1: Square line picking distance distribution function [50].

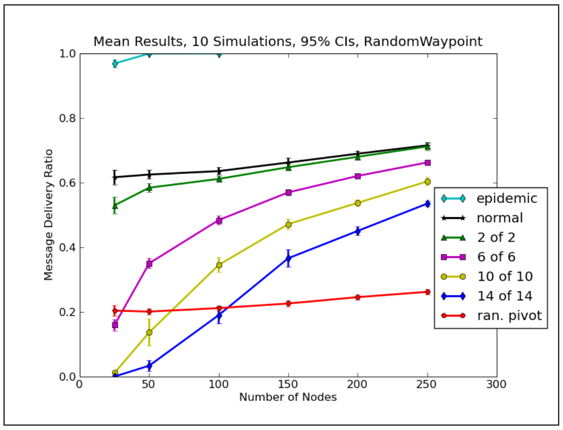


Figure 6: Message delivery ratios achieved by our various schemes. Message delivery ratio is the number of messages that got delivered over the number sent. "Normal" represents non-anonymous delivery ratio and is the best case for our anonymous schemes.

As we expect, the delivery ratio reduces as the number of groups increases. This is because 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. This is because the number of messages created in each experiment was held constant. This results more nodes per group and so it is easier to find nodes for the required groups. The random pivot scheme stays fairly level because the group size is not changing in that case (there is always 1 node per group). This graph shows tolerable losses, especially if the number and distribution of nodes to groups is appropriately configured.

Overhead: We are also interested in message overhead, the additional time it takes, beyond normal delivery time, to deliver a message from source to destination using the anonymous scheme. We measure this by running our experiments in anonymous mode and non-anonymous mode with the same seed and recording the differences in delivery times.

As in the last graph, as the number of nodes increases it gets easier to find a member of each group. This results in a lower overhead. Again, an appropriately configured network would result in only slight overhead times. Given that DTNs are expecting delay anyway, these results are encouraging.

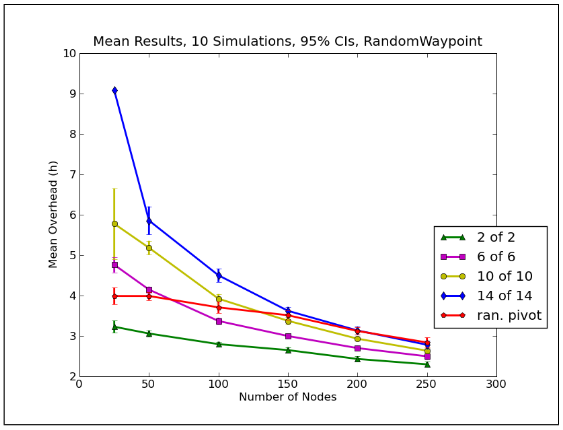


Figure 7: Mean overhead introduced by anonymity in our various schemes. Overhead is the additional time it takes to anonymously deliver a message.

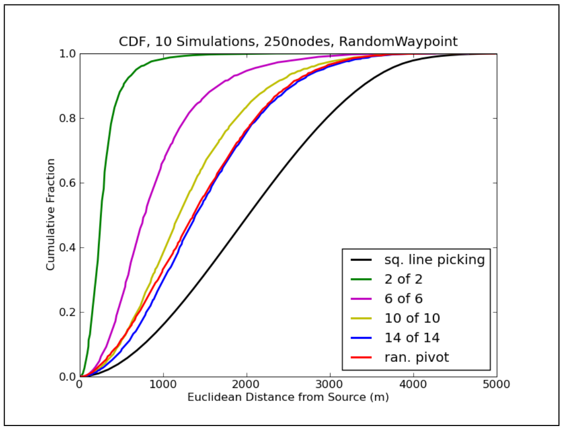


Figure 8: Distances from the pivot node to the source node measured in our various schemes. "Sq. line picking" represents the optimal distance that leaks no information about the location of the source.

Distance: Distance between source node and pivot node is important for location anonymity, as described in §5.1. We compared our recorded distance measurements to the optimal square line picking distance distribution.

We expect our random pivot scheme to match the square line picking distribution. We attribute the difference to the discrete vs. continuous difference in the square line picking distribution and our experiments. Essentially our experiments are not picking points completely randomly because it is constrained to points where nodes are located. This graph provides rather encouraging results – several moderate group sizes results in almost no leakage of information beyond the random pivot scheme. Performance gains could be achieved by using a group scheme instead of the random pivot scheme.

5.4 Enhancing the System

Here we briefly discuss several possible enhancements to our schemes.

Node Authentication: Pre-distributed keys will be used to ensure that each member of our system is authenticated. Communication can then occur without the need to contact a trusted third party to verify public key certificates.

Redundant Transfers: DoS attacks are a problem in DTNs since any node can potentially accept custody of a bundle and then drop the bundle. This action would cause a permanent loss of data if the original source does not retain a copy of the bundle. To reduce the probability that an adversary can effectively delete a message and prevent its delivery, each node can send a message to multiple other nodes. As long as at least one of these messages does not travel through an adversary, the message will be properly delivered. Note that this will increase the amount of traffic in the network and the storage space required for each node.

Cover Traffic: *Pairwise* cover traffic is achieved as follows. Whenever a node connects to another node, it periodically sends either real encrypted bundle traffic padded to a globally defined size (and MTU) to that node, or else randomly generated data in a bundle of the same size. We assume the other node is able to discern real traffic from cover traffic, e.g. by success of the decryption operation – nodes can drop bundles which decrypt to data indistinguishable from random bits. When a node is not in contact with another node, it does not attempt to send any traffic. Pairwise cover traffic is only sent on individual links between two nodes and is not forwarded into the network. When encrypted and padded properly, cover traffic provides unobservability as described in §3.1.

Although pairwise cover traffic provides unobservability, we note that cover traffic alone is not enough to completely protect against a strong adversary due to the possibility of an intersection attack [35], where a node may be able to attribute incoming and outgoing messages to the same source after watching traffic over time. To avoid this attack, we use path indirection by choosing one or more random indirection points (i.e. proxies, or pivots) through which we initially transport our bundle.

6 Conclusions and Future Work

We would like to combine the beneficial properties from each of out schemes into one scheme that will provide the best usable anonymity. Our experiments all use first contact routing to avoid changing too many parameters during experimentation. Smarter routing strategies should be investigated further. Another area for future work is the distribution of nodes to groups. This distribution will be vital for the security of the system.

We introduced anonymity and related concepts, described some desirable properties for a DTN anonymity system, and presented various schemes while discussing strengths and weaknesses of each. We experimented and measured delivery ratio, overhead, and distance and compared our anonymous scheme to non-anonymous routing. We hope this work has shed some light on anonymity and problems it presents in DTNs.

References

[1] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo. Towards securing disruption-tolerant networking. *Nokia Research Center, Tech. Rep. NRC-TR-2007-007*, 2007.

[2] A. Back, U. Moller, and A. Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. *Lecture Notes in Computer Science*, 2137:245–257, 2001.

[3] G. Blakley. Safeguarding cryptographic keys. In *Proceedings of the National Computer Conference*, volume 48, pages 313–317. Montvale, NJ: AFIPS Press, 1979.

[4] 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 *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 618–624, 2004.

[5] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), Apr. 2007. http://www.ietf.org/rfc/rfc4838.txt.

[6] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pages 197–215. IEEE Computer Society, 2003.

[7] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[8] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[9] H. Choi, P. McDaniel, and T. La Porta. Privacy Preserving Communication in MANETs. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*, pages 233–242, 2007.

[10] J. Deng, R. Han, and S. Mishra. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. *Pervasive and Mobile Computing*, 2(2):159–186, 2006.

[11] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium-Volume 13*, pages 21–21. USENIX Association Berkeley, CA, USA, 2004.

[12] R. Doomun, T. Hayajneh, P. Krishnamurthy, and D. Tipper. Secloud: Source and destination seclusion using clouds for wireless ad hoc networks. In *IEEE ISCC 2009. IEEE Symposium on Computers and Communications*, 2009.

[13] N. Evans, R. Dingledine, and C. Grothoff. A practical congestion attack on Tor using long paths. In *18th USENIX Security Symposium*, pages 33–50, 2009.

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

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

[16] S. Farrell, S. Symington, H. Weiss, and P. Lovell. Delay-tolerant networking security overview. Internet draft, version 06, March 2009. draft-irtf-dtnrg-sec-overview-06.

[17] N. Feamster and R. Dingledine. Location diversity in anonymity networks. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004)*, Washington, DC, USA, October 2004.

[18] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. *Lecture Notes in Computer Science*, 2501:548–566, 2002.

[19] A. Herzberg, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. *IBM TJ Watson Research Center*, 1995.

[20] A. Hintz. Fingerprinting websites using traffic analysis. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies workshop (PET 2002)*, pages 171–178, San Francisco, USA, April 2002. Springer.

[21] N. Hopper, E. Y. Vasserman, and E. Chan-Tin. How much anonymity does network latency leak? *ACM Transactions on Information and System Security*, forthcoming 2009.

[22] D. Huang. Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks. *Int. J. Security and Networks*, 2(3/4), 2007.

[23] T. Hyyrylinen, T. Krkkinen, C. Luo, V. Jaspertas, J. Karvo, and J. Ott. Opportunistic email distribution and access in challenged heterogeneous environments. Demo at ACM CHANTS, September 2007. http://www.netlab.tkk.fi/~jo/dtn/2007-chants-dtn-mail.pdf.

[24] M. T. Islam. Dt-talkie: Push-to-talk in challenged networks. Demo at ACM MobiCom, September 2008. http://www.netlab.tkk.fi/tutkimus/dtn/dttalkie/dttalkie_paper_acm_mobicom_2008.pdf.

[25] M. T. Islam. Dt-talkie: Interactive voice messaging for heterogeneous groups in delay-tolerant networks. Demo at IEEE CCNC, January 2009. http://www.netlab.tkk.fi/tutkimus/dtn/dttalkie/dttalkie_paper_ieee_ccnc_2009.pdf.

[26] Y. Jian, S. Chen, Z. Zhang, and L. Zhang. Protecting receiver-location privacy in wireless sensor networks. In *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, pages 1955–1963, 2007.

[27] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), Feb. 2007. http://www.ietf.org/rfc/rfc4728.txt.

[28] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, pages 153–179, 1996.

[29] D. Johnson, D. Maltz, J. Broch, et al. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, 5:139–172, 2001.

[30] A. Kate, G. Zaverucha, and U. Hengartner. Anonymity and security in delay tolerant networks. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 504–513, 2007.

[31] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.

[32] J. Kong and X. Hong. ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 291–302. ACM New York, NY, USA, 2003.

[33] J. McLachlan and N. Hopper. Don't Clog the Queue! Circuit Clogging and Mitigation in P2P Anonymity Schemes. *Lecture Notes in Computer Science*, 5143:31–46, 2008.

[34] K. Mehta, D. Liu, and M. Wright. Location privacy in sensor networks against a global eavesdropper. In *IEEE International Conference on Network Protocols, 2007. ICNP 2007*, pages 314–323, 2007.

[35] S. Murdoch and G. Danezis. Low-cost traffic analysis of tor. In *2005 IEEE Symposium on Security and Privacy*, pages 183–195, 2005.

[36] S. Murdoch and P. Zielinski. Sampled traffic analysis by internet-exchange-level adversaries. *Lecture Notes in Computer Science*, 4776:167, 2007.

[37] T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Crypto*, volume 91, pages 129–140. Springer, 1991.

[38] L. Peltola. Dtn-based blogging. Special Assignment, TKK Networking Laboratory,

2007. http://www.netlab.tkk.fi/~jo/dtn/2007-Lauri-Peltola-DTN-Blogging.pdf.

[39] L. Peltola. Enabling dtn-based web access: the server side. Master's thesis, Helsinki University of Technology, Department of Communications and Networking, April 2008. http://www.netlab.hut.fi/~jo/dtn/2008-Lauri-Peltola.pdf.

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

[41] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85. ACM New York, NY, USA, 1989.

[42] J. Raymond. Traffic analysis: Protocols, attacks, design issues, and open problems. *Lecture Notes in Computer Science*, pages 10–29, 2001.

[43] M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[44] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), Nov. 2007. http://www.ietf.org/rfc/rfc5050.txt.

[45] A. Serjantov and P. Sewell. Passive attack analysis for connection-based anonymity systems. In D. Gollmann and E. Snekkenes, editors, *Proceedings of Privacy Enhancing Technologies workshop (PET 2002)*, pages 116–131, Gjvik, Norway, December 2003. Springer.

[46] A. Seth and S. Keshav. Practical security for disconnected nodes. In *1st IEEE ICNP Workshop on Secure Network Protocols, 2005.(NPSec)*, pages 31–36, 2005.

[47] S. Seys and B. Preneel. ARM: Anonymous routing protocol for mobile ad hoc networks. In *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, volume 2, 2006.

[48] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[49] R. Song, L. Korba, and G. Yee. AnonDSR: efficient anonymous dynamic source routing for mobile ad-hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 33–42. ACM New York, NY, USA, 2005.

[50] Square Line Picking. http://mathworld.wolfram.com/SquareLinePicking.html.

[51] D. Sy, R. Chen, and L. Bao. Odar: On-demand anonymous routing in ad hoc networks. In *Proc. of IEEE MASS*, pages 267–276, 2006.

[52] S. Symington, S. Farrell, H. Weiss, and P. Lovell. Bundle security protocol specification. Internet draft, version 08, March 2009. draft-irtf-dtnrg-bundle-security-08.

[53] M. Tompa and H. Woll. How to share a secret with cheaters. *Journal of Cryptology*, 1(3):133–138, 1989.

[54] TOR Path Specification. https://git.torproject.org/checkout/tor/master/doc/spec/path-spec.txt. Accessed August 12, 2009.

[55] The TOR Project. http://www.torproject.org/.

[56] A. Tran, N. Hopper, and Y. Kim. Hashing it out in public.

[57] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao. Towards event source unobservability with minimum network traffic in sensor networks. In *Proceedings of the first ACM conference on Wireless network security*, pages 77–88. ACM New York, NY, USA, 2008.

[58] Y. Zhang, W. Liu, and W. Lou. Anonymous communications in mobile ad hoc networks. In *Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, 2005.

[59] 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.

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

A Delay and Disruption Tolerant Network (DTN) Overview

This appendix provides an overview of DTNs and discusses how security can be achieved in a DTN environment. We also discuss the current design strategies (the Bundle Security Protocol [52, 16]) that provide fundamental cryptographic services for data traveling over DTNs.

A.1 Characteristics

DTNs have several characteristics that differentiate them from more conventional, structured networks. These characteristics have introduced challenges in the communication of DTN devices, and also the security of communication. The standard DTN architecture [5] and data transfer protocol [44] include the following concepts.

Resources: DTNs are generally composed of any number of challenged networks, each of which consist of resource-limited mobile devices that can be found in vehicles, phones, actuators, and deep-space satellites. These types of devices experience both high latency and low data rates. The resource limitation also affects connectivity.

Connectivity: DTNs differ from regular networks in that there is no guarantee of end-to-end connectivity between any two nodes in the network. Communication between DTN nodes happens in an *opportunistic* manner, e.g. as a car passes another. Even this opportunistic transfer of data can not be considered reliable, and the time period in which the devices are within communication distance can be quite short. This lack of connectivity has resulted in the creation of new methods of data transfer. Techniques that attempt to maximize efficiency of communication include fragmentation, store-and-forward, and custody transfers.

Fragmentation: Ssince the connection between two DTN devices might be short lived, a fragmentation method of data transfer is used. A *bundle* is created from the data that requires transmission to another node in the DTN. As this bundle travels through the network, nodes may wish to fragment it for more efficient transfers, or because of resource constraints. These fragments themselves become smaller bundles which can be reconstructed at any other point in the network (e.g. at the destination node.) Fragmenting bundles might increase reliability of data transfer and efficiency in some networks because it reduces wasted communication opportunities that occur when a large bundle transfer does not complete.

Store-and-Forward: Each node is only expected to have limited exposure to other nodes, and is also required to pass along data during this exposure period. Therefore, nodes must store data between contact with other nodes, which represents a store-and-forward strategy to data transfer. Significant implications of memory limited devices reduce the reliability of data transfer.

Custody Transfers: Custody transfer is an attempt to increase reliability of data and reduce storage requirements of nodes. This concept specifies a single node that is responsible for, or has custody of, a bundle. As a bundle travels through the network, its custody changes to each node in the path through which it travels. After custody transfers to the next node in the path, previous nodes are not required to store the bundle. If a bundle transfer does not complete successfully, the custody is not transferred to the next node. This results in a high level of assurance since there will always be one node in the network who will do its best to ensure that the bundle continues on its path towards the destination.

A.2 Security Threats

Although communication mechanisms in DTNs are vastly different than in the traditional Internet, the threats to security remain somewhat similar in both environments. The major threats include interception, modification, and injection of bundles. These threats represent basic security vulnerabilities in DTNs, and although these issues have been handled in an Internet environment, they must be reconsidered and also managed properly in DTNs. These threats are discussed below and shown in figure 9.

Message Interception: Interception of messages is a relatively easy task. In IP, this can be achieved by controlling a link between source and destination, or by directly tapping the wire of interest. In DTNs, similar strategies can be employed for networks that are connected with wires. Wireless networks send data through the air, which is also relatively easy to intercept with appropriate signal receivers. These methods allow for passive traffic analysis of data traveling through DTNs. If the message is not properly secured, the messages' content and meta-data can be read, including its source, destination, length, time-stamp, etc. This has implications in both confidentiality and anonymity.

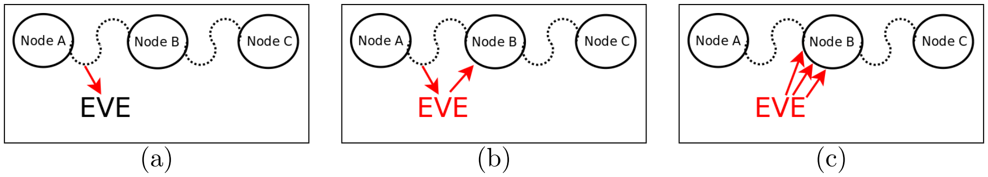


Figure 9: (a) Eve passively monitors a link and captures a message (b) Eve modifies a captured message and replays it back to the network (c) Eve injects several random messages to consume resources

Message Modification: Once a message has been intercepted and analyzed, it can then be modified given that it has not been properly secured. The modified message might contain malicious information that the message initiator did not intend to transmit. However, this message can still appear to have originated at the original source node. This attack might be launched by a military opponent to change battle commands sent to soldiers. Appropriate methods should be used to ensure proper data integrity, confidentiality, and source authentication.

Message Injection: Another plausible threat is message injection. Malicious DTN nodes might find it useful to their goal to inject messages into the network. This message injection might contain messages that have been intercepted and modified, or messages that look like real messages but in fact only contains random or uninteresting data. Replay of old messages could allow enemies to cause attacks to be unnecessarily relaunched, and injecting modified messages could cause an attack to change its target. These have significant consequences in a military environment. Injecting junk data could result in a DoS attack since it would use up precious network resources that might otherwise be used for legitimate traffic.

A.3 Security Requirements

The requirements for security in DTNs have been extracted from the threats described in appendix A.2, represent basic security properties found in many environments, and are not specific to DTNs.

Authentication: Data origin authentication provides corroboration of the source of a message. It ensures that the source of the message is given accurately. Data authentication can be achieved using keyed cryptographic hash functions – also called hash message authentication codes (HMACs) – and cryptographic signatures of the data being transmitted. Data authentication will prevent attacks where the adversary poses as an authentic user, and also can prevent users from performing actions that require authentication. This can allow a system to track malicious behaviors. Authentication and its uses in DTNs is further discussed in appendix A.4.

Integrity: Data integrity allows for the detection of the unauthorized modification of data. Data integrity can also be provided by MACs and signatures. Data integrity enables the detection of the modification of data on the path from source to destination. This will prevent attacks that attempt to modify the message, because this modification will be detected and handled appropriately. Integrity mechanisms, which are further discussed in appendix A.4, are being developed into DTNs.

Confidentiality: Data confidentiality or privacy involves keeping information secret from those who are not authorized or intended to see it. The cryptographic operations providing confidentiality include encryption and decryption. Since messages can be intercepted so easily in DTNs, it is important to ensure that the messages can not be read by others. Confidentiality is being built into the DTN security model, which is discussed in appendix A.4.

A.4 Bundle Security Protocol

The current Internet draft of the BSP [52], and related overview [16], discuss the inclusion of several security blocks that can be added to a bundle to provide the security requirements discussed in appendix A.3. A ciphersuite that contains the desired cryptographic techniques is defined for each of these blocks. An overview of these security blocks is given below.

Overview: A DTN might be composed of several different types of networks, each with their own underlying characteristics. Some of these networks might require security at the bundle protocol layer, while others may not. Because of this, the BSP defines a notion of a *security zone*. Security zones consist of *security-sources* and *security-destinations* at which *security services* are to be applied and removed, respectively. The application/removal of security services involves adding or removing *security blocks* that contain *security results* needed to provide a given service (e.g. the result of a signature or some encrypted data). The BSP defines a notion of both *hop-by-hop* security and *end-to-end* security. A hop-by-hop security service is one that is applied and removed at each hop on the transmission path, whereas end-to-end security services are only added at security-sources and removed at security-destinations (although the benefits are realized along the entire communication path).

Bundle Authentication Block (BAB): The BAB is used to provide data authentication and integrity on a hop-by-hop basis. For example, if A and B are two adjacent communication nodes, and A is sending a message (that includes a BAB) to B, B can be sure that she is indeed receiving data from A and that the data was not modified between the time when A sent it and B received it. The BAB uses the BAB-HMAC ciphersuite, which provides authenticity and integrity services through the use of a keyed hash message authentication code (based on SHA1 and using a shared secret). Although SHA1 is considered broken, the problem with collisions is not affected when combined with a key in the HMAC. Therefore this technique provides 160-bit security. Since a BAB is a hop-by-hop service, the next node will verify that the HMAC included in the BAB is consistent with an HMAC it produces (using a shared key) upon receiving the message. If verification fails, the bundle is dropped.

Payload Integrity Block (PIB): The PIB is used to provide end-to-end integrity. A PIB applies security services at the security-source intended for removal at the security-destination. The PIB uses the PIB-RSA-SHA256 ciphersuite, which provides integrity through the use of an RSA signature on a SHA256 hash of the message. Because of the birthday attack, the probability of finding a collision reduces SHA256 to 128-bit security. This security strength is maintained when using an appropriately long RSA key (e.g. 3072 bits). Once the bundle reaches the security-destination, the PIB is removed and the signature is verified by using the public key of the signer. If this verification fails, the bundle is dropped.

Payload Confidentiality Block (PCB): The PCB provides end-to-end data confidentiality. Similar to a PIB, the PCB applies security services at the security-source intended for removal at the security-destination. The PCB uses the PCB-RSA-AES128-PAYLOAD-PIB-PCB ciphersuite, which provides confidentiality through AES encryption of the payload and other security blocks using a 128-bit symmetric key (providing 128-bit security). The symmetric key is encrypted with the RSA public key of the security-destination and is placed inside the security result field in the PCB. The RSA key used for encryption must be a different key than the one used for signatures when creating PIBs, otherwise an adversary could recover a message encrypted for the security-destination by asking it for a signature on the same message (this is because a RSA signature is like RSA decryption with the private key). When PCB protects the payload, it encrypts the payload in-place. When a PCB is used to protect another security block, an encrypted form of the block is encapsulated and placed in the security result field.

Extension Security Block (ESB): The ESB also provides end-to-end data confidentiality. The ESB is very similar to a PCB, except that it is only allowed to protect extension blocks (extension blocks are special non-standard blocks that specific DTN applications might have added). The ESB uses the ESB-RSA-AES128-EXT ciphersuite, which provides the same functionality as the PCB. As with PCB, the RSA key used to encrypt the symmetric key must be different than the one used for RSA signatures.