Leveraging Social Contacts for Message Confidentiality in Delay Tolerant Networks

Karim El Defrawy, John Solis, Gene Tsudik
{keldefra,jsolis,gts}@ics.uci.edu

University of California, Irvine

ABSTRACT

Delay- and disruption-tolerant networks (DTNs) have received much attention from the research community in recent years and are likely to play an important role in future networking. DTNs can bring much-needed connectivity to rural areas and other settings with limited or non-existing infrastructures. High node mobility and infrequent connectivity inherent to DTNs make it challenging to implement simple and traditional security services, e.g., message integrity and confidentiality. In particular, it is hard to retrieve credentials of peer users/nodes. Also, multi-round security protocols (typically found in handshakes at network and session layers) are greatly handicapped due to long and uneven delays.

In this paper, we focus on the problem of initial secure context establishment in DTNs. We construct a scheme that allows DTN users to leverage social contact information in order to exchange confidential and authentic messages. We show how the proposed scheme applies for both intra- and inter-region communication scenarios.

1. INTRODUCTION

Delay- and disruption-tolerant networks (DTNs) are characterized by highly mobile nodes, intermittent connectivity and frequent disruptions. Disruptions may occur because of limited wireless communication range, sparsity of nodes, energy resources, attacks and general noise. At the same time, DTNs represent an attractive solution for low-cost networking in rural areas and developing countries with no or poor communication infrastructure. Several prototype DTNs are already in use and have been reported on in the literature [3, 19, 26].

A sample DTN usage scenario, as shown in Figure 1, would be a person (DTN user) located in a rural area who wants to send content to someone in a different area. The originating area might not have any communication infrastructure, or using it might be too expensive (e.g., GSM). However, suppose that there is a bus running through the villages on a regular basis. The bus is equipped with a wireless access point. Each time a passenger enters and/or whenever the bus stops, data can be uploaded or downloaded. The bus travels along its route, and, during its course of travel, transfers content to other passengers, other buses or fixed infrastructure nodes. The exact method of transfer depends on the underlying routing protocol implemented by the particular DTN. When the bus arrives at the central station or depot (which has access to some infrastructure) the remaining content is off-loaded to some fixed node(s) connected to the Internet. Once the data reaches the Internet, it travels to the destination if the latter is directly connected to the Internet. Otherwise, it travels to the fixed point closest to the destination and then uses other DTN nodes to reach the destination. For example the destination could also be in a rural area served by its own bus that delivers DTN content.

As illustrated by this scenario, DTNs provide a means of inexpensive communication in the absence of communication infrastructure. A DTN can also be appropriate for cases when the network operates on a temporary basis and investment in costly fixed infrastructure is unjustified. Furthermore, DTNs are well-suited for very long-haul communication settings, such as space and inter-planetary networks [5, 8, 20], where distances (and delays) are very long, error rates are potentially high, and rotation of celestial bodies (and nodes themselves) can periodically inhibit communication. However, in this paper, we focus on terrestrial DTNs characterized by the sample usage scenario above.

Inhibited communication and low connectivity results in a network that is frequently partitioned. Complete source-to-destination paths rarely exist and, even when they do, tend to be highly unstable. This raises a number of security-related challenges.

One security challenge is the problem of initial secure context establishment. We can no longer assume that a traditional public key infrastructure (PKI) is universal and available. Even if a common PKI is available, we cannot assume that each user/node can retrieve the public key of a peer or that queries (e.g., to obtain a peer's certificate) are returned in a timely fashion. If users can not retrieve the necessary parameters then they can not establish a secure context.

This paper presents a simple technique for achieving message confidentiality by using casual information – knowledge of current and previous affiliations as well as social contacts of peers. We use this knowledge to link a user to a more prominent entity (e.g., an institution or a group of users) that is likely to have a public key already known to the originating user. With this general approach, a DTN user need only store a small number of certificates or they may be cached at some nearby infrastructure node with much better connectivity to the source than the destination.

This paper is structured as follows: sections 2 and 3 motivate our work and formulate the problem at hand, respectively. Section 4 describes our DTN model and assumptions.

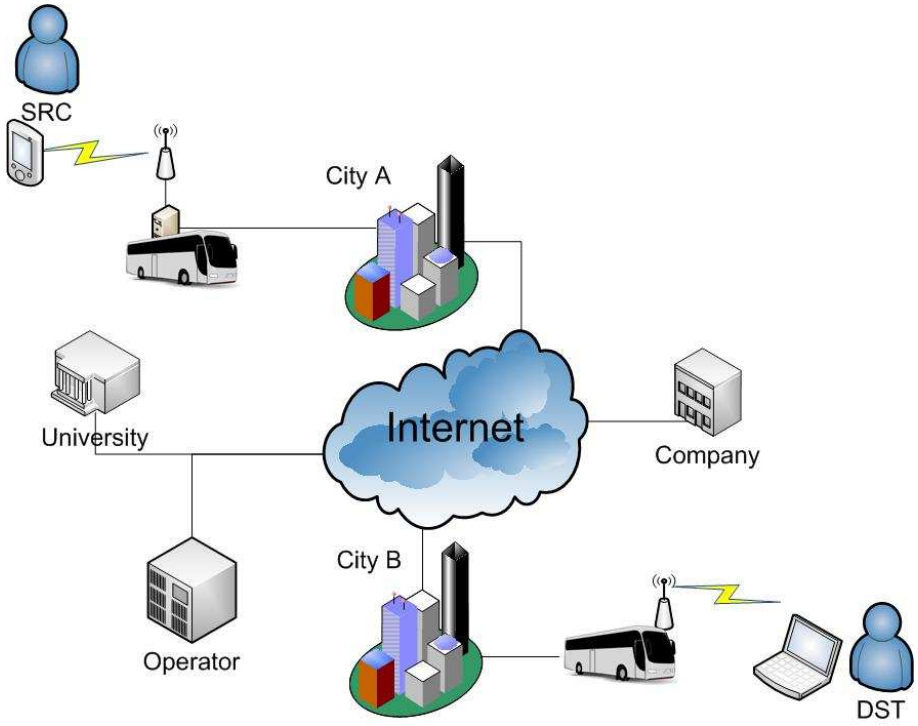

Figure 1: Sample DTN usage scenario

Sections 5 and 6 present the proposed scheme. Section 7 discusses analytical and simulation results. Next, section 8 presents the security analysis. Related work is summarized in section 9 and section 10 overviews future research directions.

2. MOTIVATION

Several factors motivate distinct security techniques for DTNs.

Traditional protocols for establishing secure connections (e.g., SSL/TLS) require multiple rounds to exchange credentials and then agree on a set of cryptographic algorithms and security parameters. In a DTN, the round-trip delay may be excessively long (e.g., minutes or hours). The path may be unpredictable, as it depends on mobility patterns at the time of message transmission, i.e., which nodes cross paths. Thus, messages can be lost or delivered out-of-order. However, security protocols typically require strict message ordering. Also, requiring several protocol rounds before being able to encrypt a message is not a viable option, since DTNs are characterized by opportunistic, sporadic and isolated message transmission. Establishing secure context for a single message is highly awkward and expensive.

Some prior solutions suggested employing Identity-Based Encryption (IBE) [10, 22] and let the message source derive the destination public key from some identity string associated with the destination, e.g., an e-mail address. However, IBE is quite expensive, even compared to other more traditional public key encryption primitives, especially, considering a limited-capacity mobile device envisaged as a typical DTN end-node. Another issue is that domain-specific public IBE parameters must be distributed to set up the scheme and allow users to derive the public keys from identity strings. Distributing such parameters may be problematic since not all users belong to the same group and different groups could have different parameters.

A solution based on Hierarchical IBE (HIBE) has been proposed [22]. It assumes that the security management of the entire DTN is organized hierarchically, which is unrealistic, at least for the time being. Also, HIBE offers no forward secrecy, meaning that compromised keys can be used to compromise secrecy of earlier messages. [22] proposes an extension to the basic HIBE scheme that provides forward secrecy by using time-based keys. Such keys are issued from a secure location, such as the user's personal desktop. However, requiring all DTN nodes be paired with a separate secure device is impractical. Not all users may have a separate device to pair with or the node may be an unattended device (drop boxes/kiosks). Who becomes responsible for issuing new keys? How are these keys issued?

Another issue with HIBC is that a compromise of a public key generator (PKG) means that all keys issued by lower-level PKGs are compromised as well. Since HIBE, by definition, involves multiple PKGs, assuming that all of them are immune to compromise is impractical.

Finally, pre-sharing secret keys with all users one might ever communicate with is impossible since a DTN user cannot be expected to know in advance the entire set of other users it might communicate with. If a DTN is relatively small and fixed (or bounded) in size, various key pre-distribution schemes [2, 4, 7, 18] might be applicable. Clearly, this is a very limited approach, at best.

3. PROBLEM FORMULATION

Given the above issues, the problem at hand can be formulated as follows:

> How can a DTN user ($\mathcal{SRC}$) send a confidential message to another DTN user ($\mathcal{DST}$) without any prior security context, i.e., $\mathcal{SRC}$ neither preshares a secret key with $\mathcal{DST}$ nor does it know $\mathcal{DST}$'s public key (if one exists).

We propose a scheme whereby $\mathcal{SRC}$ leverages social information, such as workplace affiliation, or common social contacts to send a secret message to $\mathcal{DST}$. $\mathcal{SRC}$ routes the message using keys of the affiliated intermediate nodes to ultimately deliver it securely to the destination.

4. NETWORK MODEL AND ASSUMPTIONS

Following the language and definitions of the IETF DTNRG Delay-Tolerant Networking Architecture [5], we model our DTN as shown in Figure 2. The model assumes a network with multiple operating regions. Regions are defined by geographic boundaries (e.g., city or county) with gateways interconnecting regions. A gateway is a fixed node that is part of the (limited) infrastructure. If needed, a gateway performs protocol translation between different routing protocols.

Buses travel between regions and whenever they encounter gateways, messages are up- and down-loaded. We assume that most nodes within a specific region travel predominately within that region. We make the following further assumptions about the DTN:

- Each DTN nodes is uniquely identified by a variable-length Endpoint Identifiers (EID). This can take the form of (1) a single EID unique across all regions, if we assume a global addressing scheme, or (2) a tuple of the form $\{Region-ID, Entity-ID\}$, where the former is globally unique (region-specific).[1]
- The DTN is composed of heterogeneous nodes. However, we assume that all nodes have enough processing power and energy to perform certain symmetric cryptographic operations. Larger and more powerful nodes may have the ability to perform more expensive public-key operations
- Different DTN regions might be running different routing protocols. Whereas, all nodes within a region use the same routing protocol. Nodes entering a new region are notified of the local routing protocol.
- One or more gateway(s) may interconnect two regions.
- The set of all gateways can be viewed as a DTN-wide overlay network that route much of the inter-region traffic.
- Whenever a DTN message encounters a node connected to the Internet that node attempts message delivery directly over the Internet. If the destination is not reachable directly, the node routes (over the Internet) the message to the gateway of the region closest to the destination. Thereafter, DTN routing resumes.

[1] See [9] for details.

We consider the problem of routing messages both within and across DTN regions. We address both cases in the following sections.

5. INTRA-REGION MESSAGING

Table 1 summarizes the notation used hereafter.

5.1 Basic Idea

As discussed earlier, it is highly unlikely that a DTN user can retrieve a key on-demand (or store a key) for every other DTN user it may ever wish to communicate with. Instead, we observe that users can take advantage of social information to send confidential messages. Suppose that a user can link the destination to some larger (perhaps well-known) entity. The assumption is that this entity might know the destination's public key or already share a secret key with it. The entity in question could be as large as a company, a government agency, a university or a city hall. It could also be as small as a few mutual friends. With as few as two entities in common, a source can send a confidential message to a destination within the same region. At the same time, the destination can easily identify that the received message came through friends or affiliated entities, making it less likely to be spam.

5.2 Details

Even if the message is sent through some entity affiliated with the destination we still need to ensure its secrecy. A DTN user has no control over who routes its message since the basic mechanism in many DTN routing schemes [3, 16, 24, 25] is flooding.

Assuming that $\mathcal{SRC}$ wants to send a message *m* to $\mathcal{DST}$, the intra-region scheme operates as follows:

Step 1:.

$\mathcal{SRC}$ determines (or already knows) that $\mathcal{DST}$ is affiliated with $t > 1$ entities $AE_1, ..., AE_t$ present in the same region. These entities might be selected among (this is not an exhaustive list):

- $\mathcal{DST}$'s employer
- $\mathcal{DST}$'s DTN operator (service provider)
- $\mathcal{DST}$'s alma mater (e.g., high school, college or university)
- $\mathcal{DST}$'s civic/political/religious organization
- $\mathcal{DST}$'s hobby or sports club/association
- $\mathcal{DST}$'s (and $\mathcal{SRC}$'s) common friends

Intuitively, at least $t = 2$ AE-s are needed in order to offer a minimum level of confidentiality (as discussed below).

Step 2:.

For each AE_i ($0 < i \leq t$), $\mathcal{SRC}$ already has either a public key PK_{AE_i} or a shared secret key $K_{AE_i}^{\mathcal{SRC}}$. The latter is likely if AE_i is a common friend, and the former – in most

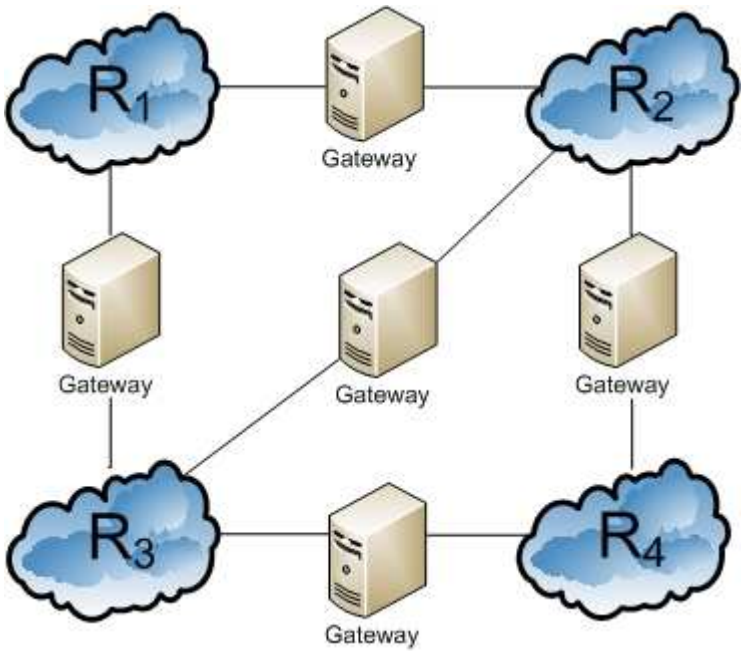

Figure 2: DTN Network Model

$\mathcal{SRC}$	source node
$\mathcal{DST}$	destination node
m	message $\mathcal{SRC}$ wants to send to $\mathcal{DST}$
PK_X	public Key of entity X
K_X^Y	symmetric key shared by entities X and Y
$AE_1, ..., AE_k$	$k > 1$ entities that $\mathcal{DST}$ is affiliated with
$E_X^Y(m)$	encryption (public key or symmetric) by entity Y of message m for entity X using either (1) a key K_X^Y that encryptor Y shares with X, or (2) X's public key PK_X
$PRF(K)$	a cipher-stream generated by a pseudo-random function keyed with a seed K. (We assume that $PRF()$ has the same bit-size as m.)

Table 1: Notation

other cases. Either way, $\mathcal{SRC}$ constructs the DTN message as $M_1 = [HDR_1, BODY_1]$, where:

$$HDR_1 = < 1, [AE_1, E_{AE_1}^{\mathcal{SRC}}(K_1)], ..., [AE_t, E_{AE_t}^{\mathcal{SRC}}(K_t)] >$$

and

$$BODY_1 = [m \bigoplus PRF(K_1) \bigoplus ... \bigoplus PRF(K_t)]$$

The initial header (HDR_1) represents a loose source route (similar to IP LSRR option [14]). The initial message body ($BODY_1$) is an onion-like structure composed by repeatedly XOR-ing the message with cipher-stream generated by a seed K_i $(0 < i \leq t)$ individually encrypted for each AE_i in the header.

Step 3:.

At some point, M_1 reaches the first LSRR hop AE_1. Recall that AE_1, like other AE-s in the route, is assumed to "know" $\mathcal{DST}$ and either shares with it a secret key $K_{\mathcal{DST}}^{AE_1}$ or has $\mathcal{DST}$'s public key. AE_1 decrypts $E_{AE_1}^{\mathcal{SRC}}(K_1)$ found in HDR_1, re-encrypts K_1 as $E_{\mathcal{DST}}^{AE_1}(K_1)$, and constructs $M_2 = [HDR_2, BODY_2]$, where: $BODY_2 = BODY_1$ and:

$$HDR_2 =< 2, \quad [\mathbf{AE_1}, \mathbf{E}_{\mathcal{DST}}^{\mathbf{AE_1}}(\mathbf{K_1})], [AE_2, E_{AE_2}^{\mathcal{SRC}}(K_2)], ..., [AE_t, E_{AE_t}^{\mathcal{SRC}}(K_t)] >$$

This process repeats itself until the message M_t reaches AE_t. At that time, $BODY_t = BODY_1$ and:

$$HDR_t =< t, \quad [\mathbf{AE_1}, \mathbf{E}_{\mathcal{DST}}^{\mathbf{AE_1}}(\mathbf{K_1})], ..., [\mathbf{AE_{t-1}}, \mathbf{E}_{\mathbf{AE_{t-1}}}^{\mathcal{SRC}}(\mathbf{K_{t-1}})], [AE_t, E_{AE_t}^{\mathcal{SRC}}(K_t)] >$$

AE_t decrypts $E_{AE_t}^{\mathcal{SRC}}(K_t)$ found in HDR_t, re-encrypts K_t as $E_{\mathcal{DST}}^{AE_t}(K_t)$, constructs $M_{t+1} = [HDR_{t+1}, BODY_{t+1}]$ and sends it to $\mathcal{DST}$.

Step 4:.

Upon receiving M_{t+1}, for each AE_i, $\mathcal{DST}$ decrypts the corresponding $E_{\mathcal{DST}}^{AE_i}(K_i)$ (using either $PK_{\mathcal{DST}}$ or $K_{\mathcal{DST}}^{AE_i}$) to obtain K_i. Finally, it computes:

$$m = BODY_{t+1} \bigoplus_{i=1}^{t} PRF(K_i)$$

We note that one benefit of using stream ciphers in constructing $BODY$ is that there need not be any fixed order of decryption and, hence, of route traversal. The same process as above would hold (with minor header modifications) regardless of the order that $AE_1, ...AE_t$ are traversed.

5.3 The Poor Man's Approach

The worst-case scenario for DTN security (at least for message confidentiality) occurs if $\mathcal{SRC}$ cannot identify any affiliated entities for $\mathcal{DST}$. The following scheme attempts to make the best of the situation.

1. $\mathcal{SRC}$ generates a key $K = H(K_1, ..., K_t)$ where $t > 1$, $H()$ is a suitable cryptographic hash function (e.g., SHA-2 [1] and $K_1, .., K_t$ are random values of appropriate size.[2]
2. $\mathcal{SRC}$ composes $t+1$ messages: $M_0 = m \bigoplus PRF(K)$ and $M_i = K_i$ (for $0 < i \leq t$).
3. $\mathcal{SRC}$ sends these messages in different directions and with a certain delay in between, so as to force them to travel along different DTN routes (see Section 7 below).
4. When $M_0, ..., M_t$ arrive at $\mathcal{DST}$, the latter recomputes K and decrypts M_0 to obtain m.

The obvious problem with this approach is that an intermediate node that receives all $t+1$ messages can simply recover m. Increasing t decreases the probability that a single node will capture all messages; but, it also increases latency. We investigate this issue further in Section 7.

Of course, to increase reliability, forward error correcting (FEC) codes (e.g., erasure codes [17]) can be used to ensure that receiving $t' < t$ components is enough to reconstruct the entire message. Alternatively, secret sharing [23] can be used for the same purpose. However, neither approach results in better security: if $\mathcal{DST}$ can recover the message after receiving $t' < t$ components, so can any intermediate node which receives as many.

A more promising, though opportunistic, measure is to modify the above scheme such that any intermediate node that shares a key with $\mathcal{DST}$ or knows $\mathcal{DST}$'s public key uses it to encrypt the message. To see how this results in better security, consider the following example. Suppose that, inadvertently, all $t + 1$ message components M_0,...,M_t traverse the same malicious intermediate node X. However, suppose that *at least one* of the components (M_i) passed through another DTN node Y, before reaching X, and Y knows $PK_{\mathcal{DST}}$ or has a shared key $K_{\mathcal{DST}}^{Y}$. In either case, Y encrypts M_i so that, by the time M_i reaches X, it cannot be used to compute m. Obviously, this method offers only best-effort-type security and only extensive simulations and/or experiments can measure its effectiveness.

[2]Each K_i is at least s bits long, where s is a security parameter.

6. INTER-REGION MESSAGING

Inter-region messaging occurs primarily between gateway nodes and is based on high-speed link access and gateway capabilities. Gateways with access to high-speed links (e.g., 3G, WiFi or WiMax), can use multi-round protocols to establish secure channels and forward messages.

Regional gateways without high-speed link access route messages over the DTN utilizing their individual capabilities. If a gateway can perform public-key operations, a secure channel can be established, as in the previous case, to forward confidential messages. Otherwise, techniques based on symmetric cryptography must be used. In our target scenario regions are defined by geographic boundaries. The number of gateways and their locations remain relatively constant. This lets us employ techniques originally developed for MANETs and sensor networks, including: probabilistic key exchange protocols [7], bipartite key agreement [18], and interleaved encryption [4]. In the rest of this section we describe how to use these techniques in conjunction with intra-region routing described above.

6.1 Authenticated Interleaved Encryption

In [4], Castelluccia presents a method for WSN nodes to exchange messages securely without first sharing a secret key or using public key cryptography. Keys are assigned to sensors according to the so-called *Leap-Frog* key distribution model [13]. Each node shares a key with every immediate neighbor and every neighbor of immediate neighbor, i.e, with all nodes that are one and two hops away. Keys shared with immediate neighbors are not known to 2-hop neighbors. Nodes use these keys along with a commutative

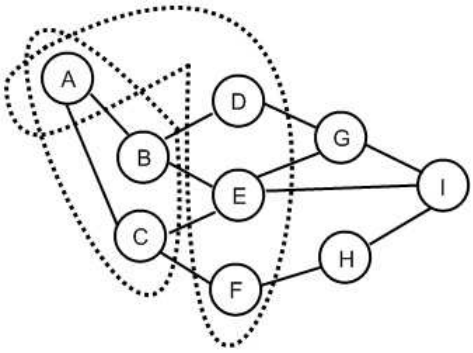

Figure 3: Interleaved Encryption Setup

encryption scheme (e.g., any stream cipher) to "interleave" message encryption. An example is shown in Figure 3: node A shares keys with one-hop neighbors B and C, as well as with two-hop neighbors D, E and F. If A wants to send a message to G, it first encrypts the message under K_B^A and then, under K_E^A. When B receives this message it removes one encryption layer (under K_B^A) and adds a new layer under K_G^B. Next, E removes its encryption layer (under K_E^A) and adds one under K_G^E. Finally, G receives the message and

removes both layers using K_G^E and K_G^B, respectively.

It might seem that this simple scheme is suitable for both intra- and inter-region routing in DTNs. However, it turns out that it does work if: (1) nodes are highly mobile, (2) network size is dynamic, or (3) two adjacent nodes collude. It does work well in the environment for which it was designed, i.e., WSNs that remain fixed/static after initial deployment.

In our DTN model, gateways interconnecting regions are part of a fixed (in size) and non-mobile infrastructure. If we consider only the gateways, they form a network that is a perfect candidate for the interleaved encryption scheme. Furthermore, we do not need to worry about collusion attacks since infrastructure nodes are trusted.

The remaining issue, discussed below, is how to incorporate interleaved encryption among gateways into the broader DTN context.

6.2 Using Interleaved Encryption

The main idea in adopting interleaved encryption is to use well-known local gateways of $\mathcal{SRC}$ and $\mathcal{DST}$ regions as some of the affiliated entities on the path to $\mathcal{DST}$. Interleaved encryption, coupled with with intra-region routing, enables secure messaging across the entire DTN, as described below.

Step 1:.

$\mathcal{SRC}$ determines $\mathcal{DST}$'s home region by examining the *RegionName* field of the $\mathcal{DST}$'s EID tuple. $\mathcal{SRC}$ identifies its local region gateway $\mathcal{GW}_{\mathcal{SRC}}$ and $\mathcal{DST}$' region gateway $\mathcal{GW}_{\mathcal{DST}}$. It then selects $(t-1)$ affiliated entities (AE-s). For each AE_i ($0 < i \leq t-1$), $\mathcal{SRC}$ already has either a public key PK_{AE_i} or a shared secret key $K_{AE_i}^{\mathcal{SRC}}$. In addition, $\mathcal{SRC}$ has $(PK_{\mathcal{GW}_{\mathcal{SRC}}})$.

$\mathcal{SRC}$ constructs the DTN message as: $M_1 = [HDR_1, BODY_1]$, where:

$$\begin{aligned} HDR_1 &= < 1,\ [AE_1, E_{AE_1}^{\mathcal{SRC}}(K_1)], ..., \\ &\quad [AE_{t-1}, E_{AE_{t-1}}^{\mathcal{SRC}}(K_{t-1})], \\ &\quad [\mathcal{GW}_{\mathcal{SRC}}, \mathcal{E}_{\mathcal{GW}_{\mathcal{SRC}}}^{\mathcal{SRC}}(\mathcal{K}_{\sqcup})]] > \end{aligned}$$

and

$$BODY_1 = [m \bigoplus PRF(K_1) \bigoplus ... \bigoplus PRF(K_t)]$$

$\mathcal{SRC}$ then sends the message to AE_1.

Step 2:.

At some point, M_1 reaches the first LSRR hop AE_1. Recall that AE_1, like other AE-s in the route, is assumed to "know" $\mathcal{DST}$ and either shares with it a secret key $K_{\mathcal{DST}}^{AE_1}$ or has $\mathcal{DST}$'s public key. AE_1 decrypts $E_{AE_1}^{\mathcal{SRC}}(K_1)$ found in HDR_1, re-encrypts K_1 as $E_{\mathcal{DST}}^{AE_1}(K_1)$ and constructs $M_2 = [HDR_2, BODY_2]$, where: $BODY_2 = BODY_1$ and:

$$\begin{aligned} HDR_2 &= < 2,\ [\mathbf{AE_1}, \mathbf{E}_{\mathcal{DST}}^{\mathbf{AE_1}}(\mathbf{K_1})], [AE_2, E_{AE_2}^{\mathcal{SRC}}(K_2)], \\ &\quad ..., [AE_{t-1}, E_{AE_{t-1}}^{\mathcal{SRC}}(K_{t-1})], \\ &\quad [\mathcal{GW}_{\mathcal{SRC}}, \mathcal{E}_{\mathcal{GW}_{\mathcal{SRC}}}^{\mathcal{SRC}}(\mathcal{K}_{\sqcup})] > \end{aligned}$$

This process repeats itself until the message ($M_{t-1} == [HDR_{t-1}, BODY_{t-1}]$) reaches AE_{t-1}. At that time, $BODY_{t-1} = BODY_1$ and:

$$\begin{aligned} HDR_{t-1} &= < t,\ [\mathbf{AE_1}, \mathbf{E}_{\mathcal{DST}}^{\mathbf{AE_1}}(\mathbf{K_1})], ..., \\ &\quad [\mathbf{AE_{t-1}}, \mathbf{E}_{\mathbf{AE_{t-1}}}^{\mathcal{SRC}}(\mathbf{K_{t-1}})], \\ &\quad [\mathcal{GW}_{\mathcal{SRC}}, \mathcal{E}_{\mathcal{GW}_{\mathcal{SRC}}}^{\mathcal{SRC}}(\mathcal{K}_{\sqcup})] > \end{aligned}$$

AE_{t-1} decrypts $E_{AE_{t-1}}^{\mathcal{SRC}}(K_{t-1})$ found in HDR_{t-1}, re-encrypts K_{t-1} as $E_{\mathcal{DST}}^{AE_{t-1}}(K_{t-1})$, constructs $M_t = [HDR_t, BODY_t]$ and sends it to $\mathcal{GW}_{\mathcal{SRC}}$.

Step 3:.

$\mathcal{GW}_{\mathcal{SRC}}$ receives $M_t = [HDR_t, BODY_t]$ and decrypts $E_{\mathcal{GW}_{\mathcal{SRC}}}^{\mathcal{SRC}}(K_t)$ and re-encrypts K_t with $PK_{\mathcal{GW}_{\mathcal{DST}}}$ and generates $M_{t+1} = [HDR_{t+1}, BODY_{t+1}]$ where:

$$\begin{aligned} HDR_t &= < t,\ [\mathbf{AE_1}, \mathbf{E}_{\mathcal{DST}}^{\mathbf{AE_1}}(\mathbf{K_1})], ..., \\ &\quad [\mathbf{AE_{t-1}}, \mathbf{E}_{\mathcal{DST}}^{\mathbf{AE_{t-1}}}(\mathbf{K_{t-1}})], \\ &\quad [\mathcal{GW}_{\mathcal{DST}}, \mathcal{E}_{\mathcal{GW}_{\mathcal{DST}}}^{\mathcal{GW}_{\mathcal{SRC}}}(\mathcal{K}_{\sqcup})] > \end{aligned}$$

and $BODY_t = BODY_{t-1}$. Succeeding routing can occur across multiple regions (using interleaved encryption); it is transparent to $\mathcal{SRC}$ and $\mathcal{DST}$.

Step 4:.

When the message reaches the destination region it is received by $\mathcal{GW}_{\mathcal{DST}}$ which decrypts $E_{\mathcal{GW}_{\mathcal{DST}}}^{\mathcal{GW}_{\mathcal{SRC}}}(K_t)$ and re-encrypts it with the key of $\mathcal{DST}$. The message M_{t+1} is then simply forwarded using the specific intra-region routing protocol. Upon receiving M_{t+1}, $\mathcal{DST}$, for each AE_i and $\mathcal{GW}_{\mathcal{DST}}$, decrypts the corresponding $E_{\mathcal{DST}}^{AE_i}(K_i)$ (using either $PK_{\mathcal{DST}}$ or $K_{\mathcal{DST}}^{AE_i}$) to obtain K_i. Finally, it computes:

$$m = BODY_{t+1} \bigoplus_{i=1}^{t} PRF(K_i)$$

This scheme works regardless of locations of affiliated entities. Also, despite the description above, we do not assume that AE nodes are traversed in any specific order. (In fact AE-s can be in different regions).

The main difference between inter- and intra-region routing is the involvement of gateways in the former. We cannot assume that gateways share keys with (or know keys of) all users in the network. Instead, we assume that a gateway "knows" keys for users of its own region. Gateways work together and use this knowledge to add another layer of encryption to the message. This is particularly important in cases of $t = 2$. With only one common affiliation, AE_1, gateways add a second layer of encryption which keeps the message secure.

7. SIMULATIONS AND RESULTS

In this section we describe the simulation and analysis results for our proposed schemes. We start by presenting our simulation and mobility model parameters and then discuss and analyze results.

7.1 Simulation Parameters

We simulated our DTN confidential messaging schemes using the DTN ONE simulator [15]. Since regions are based

on geographic boundaries we simulated each region by restricting the movement of nodes to the downtown area of a large city.

Intra-region routing results look at routing contained within the whole downtown area while inter-region restricts movements to sub-areas of the downtown area. The exact divisions and gateway-locations are shown Figure 4. Results represent the average of five iterations for each scenario.

Figure 4: Map area for intra-/inter-region routing

7.2 Synthetic Mobility Model

We select parameters that simulate users with portable devices walking around during the course of a day sending emails and pictures.

Mobility: The map of the Helsinki, Finland downtown area (about 14 km^2) was used to restrict the mobility model. i.e., only existing roads and transportation lines could be traveled. 250 nodes move for 24 hours along the shortest path between two points on the map. This time frame allows for enough time to deliver all generated messages. Nodes move at a speed selected uniformly at random between 0.5 and 1.5 m/s. Upon reaching its destination nodes pause for a period between 0 and 120s, selected uniformly at random, before selecting a new destination. The different seeds generate different mobility patterns for each simulation run.

Connectivity and transmission: User nodes can only communicate with one other node at a time. Communication is bi–directional with a constant transmission rate of 250 kB/s, and continues until all messages have been exchanged or until nodes move outside of the 10 m communication range.

Gateway nodes can establish up to ten simultaneous connections. Gateways have a communication range of 1500 m and a transmission rate of 1 MB/s.

If multiple nodes are within communication range, a node will connect to one at random. If there is no data to exchange with the chosen node, or if the chosen node cannot respond, e.g., because it is exchanging data with another node, a new node is randomly selected.

Each node reserves 100 MB of memory for DTN traffic. Routing is handled by the Spray and Wait protocol [24].

Intra-region Traffic model: Nodes generate on the average one message per hour for twelve hours. The destination of each message is selected uniformly at random among all nodes. Messages have a time-to-live (TTL) attribute which is varied for the different simulation scenarios. Message sizes are uniformly distributed between 100 kB and 2 MB.

Inter-region Traffic model: Nodes generate on average two messages per hour. The destination of each message is randomly chosen among all nodes in all four regions. Message TTL and size distributions are the same as in the intra-region model.

7.3 Detailed Simulation Results

Intra-Region: The first set of simulations aim to find out if the "Poor Man's Approach", discussed in Section 5.3, is suitable for DTNs. We want to analyze the percentage of messages that can be recovered by honest-but-curious nodes in the network. Under this model nodes do not go out of their way to retrieve messages from a particular source to a particular destination. However, nodes can keep copies of any messages received in a separate storage buffer. This allows nodes to recover messages that would have normally otherwise expired. Recall that in the "Poor Man's Approach" the encrypted message and its corresponding keys are sent along different paths in the network. We modeled the simplest case in which a single key is used. This gives us an upper bound for interception probability.

We look at the probability of capturing a message and its encryption key, along with corresponding delivery ratios, by varying message TTLs and varying the delay between sending the two messages. We varied message TTLs from 2-12 hours to cover the full duration of the simulation. The delay between sending the message and its corresponding key varied from one up to six hours. The maximum number of copies of a message, as limited by Spray and Wait, was varied from 20-80 copies.

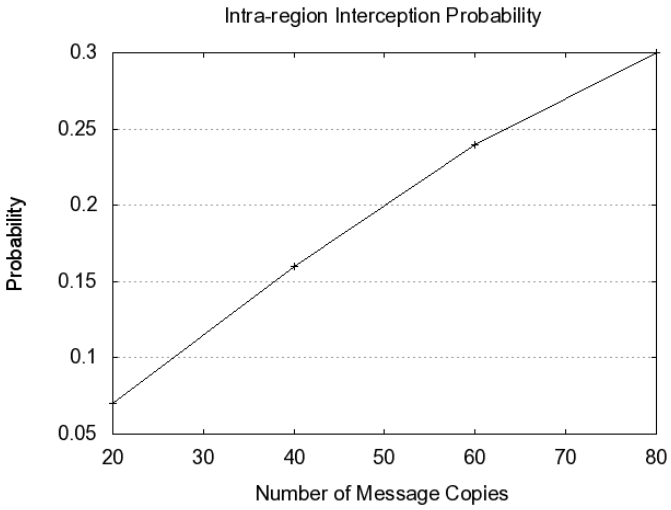

Figure 5: Traffic Recoverable by Intermediate Nodes

Figure 5 shows the amount of traffic recoverable by intermediate nodes in the network for varying message copies. Figure 6 shows the delivery ratios under each of the given

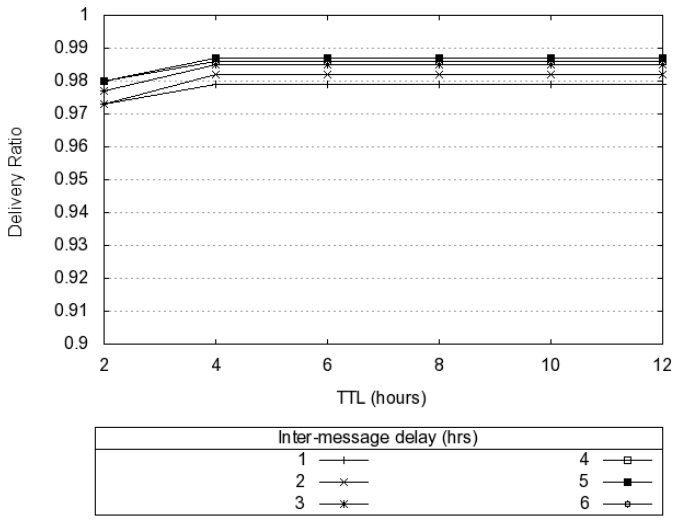

Figure 6: Delivery Ratios

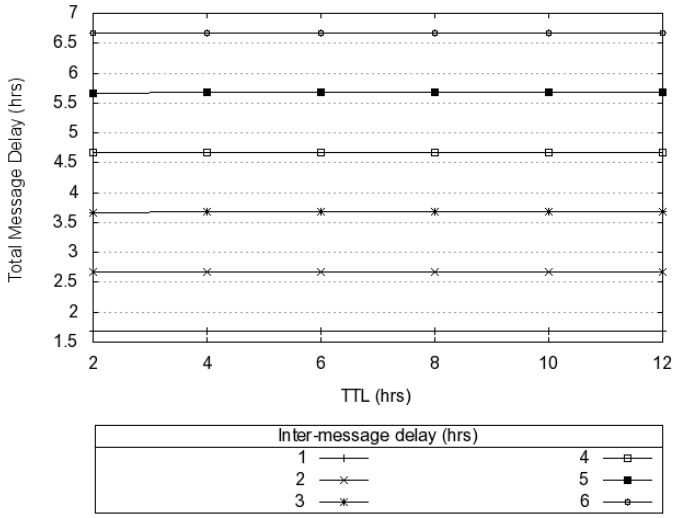

Figure 7: Message Latency

scenarios. These graphs show that sending the message and its corresponding key along different routes offers a fair degree of message confidentiality without impacting overall delivery ratio. What it does impact, as shown in Figure 7, is the overall latency of the delivered messages.

Somewhat surprisingly, increasing the TTL in our scenario only marginally helps to improve delivery ratios. We would expect that increasing the TTL allows for messages to stay alive longer and thus increase their chances at being successfully delivered. However, our results show that for our scenario, all messages are delivered within four hours and there is no gain in using large TTLs.

Since all messages are delivered within four hours the number of intermediary hops traveled before reaching the destination becomes independent of the TTL. Instead, as Figure 5 shows, the probability of interception is directly proportionally to the number of hops a message travels. Each hop in the network represents an intermediary who could potential recover the message. Increasing hop counts, by

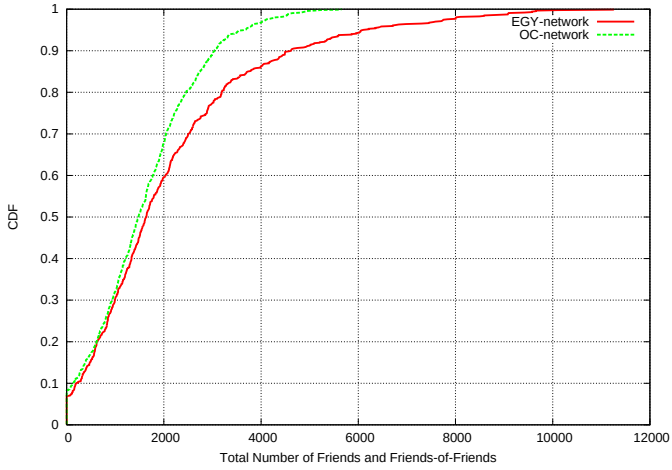

Figure 8: CDF of Social Network Reach of Users in Facebook (OC and EGY Networks)

increasing the number of copies of a particular message, increases the number of intermediaries who see a particular message.

Another observable result is that increasing the duration between messages does nothing to decrease the probability of recovery. This is because under the honest-but-curious adversarial model any node can keep a message for an indefinite period of time by copying it to a separate buffer. Nodes can recover messages even if the second part of the message is sent long after the TTL of the first.

Delay between messages directly correlates to total delivery time. Figure 7 shows that, as expected, messages sent six hours apart have a longer total delivery time. From a security perspective, a user on the average gains little by waiting for long durations between sending messages. If the user is willing to accept a best-effort form of confidentiality, then the the two messages can be sent closer together with a low probability of recovery.

*Inter-Region:*The second set of simulations aimed to test the same parameters in the inter-region scenario. While we were easily able to get delivery ratios of 40%, we felt this was too low to draw meaningful conclusions. A low probability of interception coupled with a low delivery ratio does not imply satisfactory security of the scheme. As part of our future work we plan to test additional parameters to increase the overall delivery ratio. We expect to see trends similar to those found in the intra-region scenario. In particular, we expect the probability of interception to increase with the number of hops needed to traverse the network.

7.4 Network Coverage

To determine the social coverage of friends and of friends-of-friends (FoFs) inside a region we analyze social connections from a well known social network, Facebook. Facebook is a social network that contains more than 40 million users from all over the world. Users can join smaller networks inside Facebook which correspond to countries, cities, organizations and institutions. We select two networks that corresponds to two geographical areas (Orange County (OC) and Egypt (EGY)) and crawl their users. We choose these

two networks as representatives of small geographical areas (counties) and developing countries where DTNs may provide a cost efficient solution [3]. We are interested in answering the following questions in our analysis:

- How many friends (and FoFs) does each user have on average?
- Does the size of the social graph depend on the number of immediate (one hop) friends?

It is important to study the two characteristics above because they directly affect the applicability of our schemes. If users do not have a lot of friends and their friends have only a small number of friends, then the reachability of their social network will be limited and they would need to store a large number of keys (one for each friend). On the other hand if users have a number of friends that is one or two orders (or more) of magnitude smaller than the number of people they can reach through them (FoF), then only a small number of keys will be needed while at the same time allowing a high social reachability.

Figure 8 shows the CDF of the total number of friends and FoFs for 870 and 900 random Facebook users in the OC and EGY networks respectively [4]. On average half the users in both networks can reach less than 1700 users, the other half more than 1700. In the OC network users the highest number of reachable people was 5881 while the EGY it was around 11000. Our analysis indicates that by storing only several hundred of keys at max, one can reach several thousands of users. This can be seen in more detail in Figure 9. The figure shows the number of immediate friends of the sampled users from the OC and EGY networks. Clearly the number of direct friends for 800 of them is below 100 and the number of reachable FoFs is in the order of thousands for over 400 users. A small number of users (less than 10%) have more than 100 friends and up to 500 and their reachable FoF network goes as high as 10000. It is interesting to note that although we selected two completely unrelated networks on Facebook, and randomly selected the set of users to crawl, the distribution of the number of friends and FoF look similar, although people in the EGY network seem to have a larger total social graph on average. More exhaustive measurements need to be conducted to come up with strong conclusions about the exact average distributions, but this is out of scope of this paper.

Up to this point we have only considered the analysis of the social coverage of users by considering only their friends. As we have described in previous sections one can also use the affiliations of users (e.g. their work, study or organizations) for confidential messaging. Unfortunately such a concept does not exist in the social networks and is hard to map into. Users in Facebook for examples can belong to groups, but these groups are formed randomly between

[3]We do not claim that this small study of the social graphs of these two networks is exhaustive, but it serves to prove our points.

[4]Note that Facebook has the option to make users' profiles closed, so that only their friends can see them. We can not crawl such profiles so the numbers presented in our results are conservative and do not include such closed profiles; thus the results here should be considered as a lower bound on the social reachability of users.

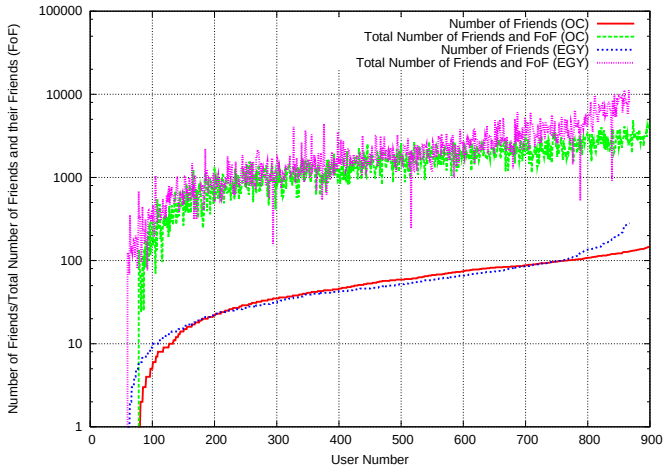

Figure 9: Number of Friends and the FoFs for a Sample of Users (OC and EGY Networks)

friends and are very transient (with some exceptions of course). On the upside, the capability to use such affiliation will strictly increase the applicability of our proposed scheme and allow users to reach more people.

8. SECURITY ANALYSIS

In this section, we present a preliminary – and quite informal – security assessment of the proposed schemes. (A more formal treatment of security is underway.) Recall that the goal is data secrecy for messages sent between two DTN users who neither share a common key nor know each other's public keys.

Security Model:.

We assess the security of our schemes under the *honest-but-curious* adversarial model. In this model, nodes do not mount active attacks to learn messages. Also, nodes do not modify or drop messages. (Hence, any violations of message integrity is the result of corrupted transmission or corrupted memory/storage). However, nodes can retain copies of messages beyond their TTL expiration in separate storage. This allows them, for example, to correlate message components sent with variable delays in our inter-region scheme.

Collusion:.

The biggest threat to message confidentiality is collusion between the intermediate affiliated entities $AE_1, ..., AE_t$. We argue that this attack is unlikely, since these entities are selected by $\mathcal{SRC}$ who would use plain common sense to avoid entities naturally prone to collusion (e.g., all government agencies or all cellphone service providers). Thus, unless these entities are already colluding beforehand, they are not all likely to collude in real time. Another issue is that many of these entities are public and operate on a long-term basis; therefore, they have little incentive to sacrifice their reputation by engaging in collusion attacks.

If affiliated entities are friends (i.e., regular DTN end-users) we assume that $\mathcal{SRC}$ trusts its friends not to collude. $\mathcal{SRC}$ is also well-advised to pick friends who do not know each other, but are socially connected to both $\mathcal{DST}$ and

$\mathcal{SRC}$. (The counter-argument is that two people who have two friends in common are apt know each other. Further investigation is needed to verify this claim).

MitM Attacks:.

Man-in-the-Middle (MitM) attacks are not applicable to our setting since public keys for affiliated entities (selected by $\mathcal{SRC}$) are obtained from appropriate public key certificates signed by trusted Certificate Authorities (CAs).

Authentication and Integrity:.

Origin authentication and data integrity are not our primary goals. Moreover, they are not within the scope of the *honest-but-curious* adversary model. However, we note that $\mathcal{SRC}$ can easily authenticate itself to $\mathcal{DST}$ by enclosing its public key certificate with the message and signing the message before encryption. Whereas, if $\mathcal{SRC}$ has no public key certificate, meaningful sender authentication is impossible. But, a weak form of message integrity is possible even without a public key certificate: $\mathcal{SRC}$ can compute and append a message hash before encryption.

9. RELATED WORK

DTNs are an active research topics with numerous recent results. An architecture for "challenged" networks characterized by very large delay paths and frequent network disruptions is presented in [9]. This design uses messages as the underlying unit of transmission. The architecture proposes a two-tier naming structure, which we briefly described earlier in the paper. [9] also proposes a hop-by-hop security model, addressing issues like authentication, confidentiality and integrity by requiring end-hosts to have certificates which bind identities to public keys. The problems with this simple approach are:

1. Key management is problematic
2. Certificate Revocation is difficult, especially, since presence of on-line entities cannot be assumed (frequent disconnections)
3. Retrieving public keys of nodes is potentially time-consuming (long delays)

In our approach, nodes are not required to obtain public keys of peers. They only need to obtain (or already have) keys for affiliated entities.

The idea of using social information in DTNs was first proposed in [6] which describes a publish-subscribe routing framework based on social interaction information among users. The principal idea is that socially-related people frequently co-locate. Information of interest to people sharing a common interest can be quickly spread to these users using social routing. Our work uses social information/social contacts to send a confidential message between users. Routing is seen as an orthogonal problem, since our approach is orthogonal to the routing protocol.

There have been a few research results in DTN security. For example, [11] discusses various threats and issues but focuses primarily on so-called *bundles* and their security. It briefly addresses security services on an end-to-end basis (e.g. confidentiality), but does not go into specifics nor considers the case of initial communication between two nodes without any prior security context.

Note: Related work utilizing Identity Based Cryptography (IBC) [10, 12, 21, 22] has already been discussed in Section 2.

10. LIMITATIONS

While the schemes proposed in this paper represent a step forward, some of the underlying assumptions may be problematic. In particular, we assumed that affiliated entities know or can easily obtain the $\mathcal{DST}$'s public key (or share a secret with it). This may not be the case nor does this assumption scale. Alternative mechanisms should be explored to relax this assumption. One potential consideration is the availability of a pervasive cellular telephone network (e.g., GSM), alongside the envisaged DTN. We can use SMS to transmit keys used for bulk encryption. Another possible extension is to send keys on an explicitly circuitous route to ensure that they do not travel close to the encrypted message.

11. CONCLUSION

This paper motivated and presented techniques for secure messaging in large-scale DTNs. They allow DTN users to leverage social contact information in order to exchange confidential and authentic content. We showed how to address the problem in both intra- and inter-region settings and assessed performance and security of out techniques. Future work includes a more formal security analysis and extensive simulations to help identify performance and security bottlenecks and limitations.

12. REFERENCES

[1] National institute of standards and technology (nist), fips publication 180-3: Secure hash standard, June 2003.

[2] Amitanand Aiyer, Lorenzo Alvisi, and Mohamed Gouda. Key grids: A protocol family for assigning symmetric keys. In *ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, pages 178–186, Washington, DC, USA, 2006. IEEE Computer Society.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, April 2006.

[4] C. Castelluccia. Securing very dynamic groups and data aggregation in wireless sensor networks. *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1–9, Oct. 2007.

[5] V. Cerf and et al. Delay-tolerant network architecture. April 2007.

[6] P. Costa, C. Mascolo, M. Musolesi, and G.P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *Selected Areas*

in Communications, IEEE Journal on, 26(5):748–760, June 2008.

[7] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002. ACM.

[8] Scott Burleigh et al. Delay-tolerant networking : An approach to interplanetary internet. *IEEE Communications Magazine*, pages 128–136, June 2003.

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

[10] Kevin Fall. Identity based cryptosystem for secure delay tolerant networking. December 2003.

[11] Stephen Farrell and Vinny Cahill. Security considerations in space and delay tolerant networks. In *SMC-IT '06: Proceedings of the 2nd IEEE International Conference on Space Mission Challenges for Information Technology*, pages 29–38, Washington, DC, USA, 2006. IEEE Computer Society.

[12] C. Gentry. Certificate-based encryption and the certificate revocation problem, 2003.

[13] M.T. Goodrich. Leap-frog packet linking and diverse key distributions for improved integrity in network broadcasts. *Security and Privacy, 2005 IEEE Symposium on*, pages 196–207, May 2005.

[14] David B. Johnson. Mobile host internetworking using ip loose source routing. Technical report, 1993.

[15] Ari Keränen and Jörg Ott. Increasing reality for dtn protocol simulations. Technical report, Helsinki University of Technology, Networking Laboratory, July 2007. Available at: `http://www.netlab.tkk.fi/tutkimus/dtn/theone/`.

[16] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[17] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. Practical loss-resilient codes. In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 150–159, New York, NY, USA, 1997. ACM.

[18] N. Mittal. Space-efficient keying in wireless communication networks. *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, pages 75–75, Oct. 2007.

[19] Alex Pentland, Richard Fletcher, and Amir Hasson. Daknet: Rethinking connectivity in developing nations. *IEEE Computer*, 37(1):78–83, January 2004.

[20] C. Peoples, G. Parr, B. Scotney, and A. Moore. A reconfigurable context-aware protocol stack for interplanetary communication. *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, pages 281–285, Sept. 2007.

[21] A. Seth, S. Fung, and S. Keshav. A secure tetherless computing architecture. Technical report, University of Waterloo.

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

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

[24] Thrasyvoulos Sphyropoulos, Konstantinos Psounis, and Cauligi Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM SIGCOMM'05*, August 2005.

[25] Amir Vahdat and David Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.

[26] Wizzy digital courier. `http://www.wizzy.org.za`.