A Framework for Anonymous Routing in Delay Tolerant Networks

Kazuya Sakai*, Min-Te Sun†, Wei-Shinn Ku‡, and Jie Wu§
*Department of Info. and Commun. Systems, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan.
†Department of Computer Science and Information Engineering, National Central University, Taoyuan 320, Taiwan.
‡Department of Computer Science and Software Engineering, Auburn University, Auburn, Alabama 36849–5347.
§Center for Networked Computing, Temple University, 1925 N. 12th St. Philadelphia, PA 19122.
ksakai@tmu.ac.jp, msun@csie.ncu.edu.tw, weishinn@auburn.edu, and jiewu@temple.edu

***Abstract*—Security and privacy issues are considered to be two of the most significant concerns to organizations and individuals using mobile applications. In this paper, we seek to address anonymous communications in delay tolerant networks (DTNs). While many different anonymous routing protocols have been proposed for ad hoc networks, to the best of our knowledge, only variants of onion-based routing have been tailored for DTNs. Since each type of anonymous routing protocol has its advantages and drawbacks, there is no single anonymous routing protocol for DTNs that can adapt to the different levels of security requirements. In this paper, we first design a set of anonymous routing protocols for DTNs, called anonymous Epidemic and zone-based anonymous routing, based on the original anonymous routing protocols for ad hoc networks. Then, we propose a framework of anonymous routing (FAR) for DTNs, which subsumes all the aforementioned protocols. By tuning its parameters, the proposed FAR is able to outperform onion-based, anonymous Epidemic, and zone-based routing. In addition, numerical analyses for the traceable rate and node anonymity models are built. Extensive simulations using randomly generated graphs as well as real traces are conducted to demonstrate that given appropriate parameter settings, our FAR outperforms all the existing anonymous routing protocols for DTNs.**

***Index Terms*—Delay tolerant networks, DTNs, anonymous routing.**

I. Introduction

Delay tolerant networks (DTNs) seek to address data communications within networks that lack continuous connectivity, such as people/pocket-switched networks, vehicular networks, battlefield communications, and so on. In these DTNs, security, privacy, and network performance, are of significant concern. For instance, one of the communicating parties in a battlefield is most likely to be a gateway to the infrastructure or a command operator. The identities and locations of such nodes should not be disclosed to the adversaries. Motivated by these observations, we are interested in anonymous wireless communications that prevent adversaries from violating mobile users' privacy, e.g., deriving users' identities, locations, and routing paths, by traffic analyses.

A great deal of effort has been invested in designing anonymous routing protocols for the internet [1] and mobile ad hoc networks [2]–[5]. The message that is protected by a number of encrypted layers, a so-called *onion* [6], is widely used to preserve the privacy of end hosts as well as routing paths. In onion-based routing, onion routers serve as proxies, and any given intermediate node will never know where the source and sink of the message are located. In mobile ad hoc networks, the location-based deanonymization attack [7] may reveal the physical location of nodes. To this end, the zone-based anonymous routing is proposed in [5] where the source and the last proxies perform restricted flooding, as to make sure that the source and destination nodes are not identifiable within the flooding zone.

In the DTN research community, a few anonymous routing protocols, which use the idea of onion groups [7]–[9] and the threshold [10], have been proposed in order to improve the degree of privacy, such as the traceable rate and node anonymity. However, the following research challenges that particularly arise in anonymous routing in DTNs are yet to be addressed.

First, it is known that the use of a number of onions results in lower traceable rate. As a consequence, onion-based protocols [7]–[9] experience slow packet delivery. Second, the anonymous set of the source and destination nodes can be reduced, should the first and last onion relay be compromised. Third, although the zone-based approach improves node anonymity, neither Epidemi-like nor zone-based protocol has been proposed so far. One reason for this is the difficulty in defining a zone in DTNs where the network graph is constructed from the past contact history, rather than from physical locations of nodes. At last, to the best of our knowledge, there is no work that balances the pros and cons of these different approaches. It is interesting to design an anonymous routing framework that subsumes all the aforementioned protocols and optimizes the anonymous DTN routing based on a number of metrics, e.g., delivery rate, anonymity, delay, and forwarding cost, by tunable parameters.

To address the above challenges, we propose the framework of anonymous routing for DTNs. The contributions of this paper are as follows.

- We first design a set of anonymous DTN protocols, including Anonymous Epidemic (AE), Restricted Epidemic Routing (RER), and Zone-Based Anonymous Routing (ZBAR), based on anonymous routing protocols originally proposed for mobile ad hoc networks. The key difference from the

existing solutions is the definition of "zone," where senders and receivers stay anonymous. The proposed RER guarantees that a message reaches at least one of the nodes in the next onion group, with a certain probability specified by the threshold. In addition, RER can be used as a subroutine of ZBAR.

- We next propose a framework of anonymous routing (FAR) for DTNs that subsume all the Epidemic, zone-based, and onion-based routing protocols with tunable parameters. In FAR, a message travels along a set of onion groups with router-by-router encryption, and every communication between two consecutive onion routers on the routing path is performed by either Epidemic routing or spray-and-wait forwarding with a time constraint. By doing this, FAR enjoys the advantages of these baseline protocols, and DTN users can balance the performance, privacy, and cost base on their preferences.
- We then quantitatively analyze the privacy metrics provided by FAR. To be specific, the closed form solutions used to estimate the traceable rate and source/destination anonymity are provided. The proposed mathematical models help DTN users to select appropriate routing parameters that meet their security and privacy requirements.
- Finally, we conduct extensive simulations using one of the well-known real traces, CRAWDAD dataset cambridge/haggle [11], as well as random graphs to demonstrate the performance and degree of privacy of the proposed scheme. Furthermore, the simulation results are compared with analytical results, and the comparisons show that our analyses provide very close approximations.

The rest of this paper is organized as follows. Related works are reviewed in Section II. In Section III, the AE, RER, and ZBAR protocols specifically revised for DTNs are presented. These protocols will serve as the building blocks for the proposed FAR, which is introduced in Section IV. The mathematical analysis of the proposed FAR is presented in Section V. The performance of the proposed scheme is evaluated by computer simulations in VI. The discussions on how to select parameters is provided in Section VII. Section VIII concludes this paper.

II. Related Work

A. DTN routing

Epidemic routing [12] is a flooding-like message forwarding scheme that allows nodes to copy a message at every contact. While this approach maximizes the delivery rate and minimizes the delay when buffer constraint is not considered, it incurs a large amount of overhead. A ticket-based protocol, e.g., spray-and-wait [13], balances the trade-off between the performance and control overhead by limiting the number of copies of a message. Based on how the tickets are controlled, there are two types of spray-and-wait protocols: source and binary spray-and-wait. In the source spray-and-wait protocol, the source node has L tickets and consumes one ticket by forwarding a message at every contact. Thus, the source can duplicate up to L copies of a message. In the binary spray-and-wait, the source node with L tickets gives $L/2$ tickets at the first node it has a contact with. That is, every node with a message consumes half a ticket at every contact. To improve the message delivery with limited tickets, probabilistic analysis based on knowledge oracles [14], e.g., past contact history, queueing, and traffic demand, is incorporated to improve the delivery rate [15] and/or reduce the redundant message forwarding [16]. Depending on what metric a system administrator likes to emphasize the most, such as the average delay and worst-case delay, a suite of utility functions are proposed in [17].

B. Anonymous Routing for Ad Hoc Networks

Anonymous routing protocols in ad hoc networks are divided into either onion-based [2]–[5] or location-based protocols [5]. In onion-based routing, the layered encryption, with different sets of secret keys, is applied to sensitive data and/or routing information, and such encrypted information is called an *onion*. This data structure forces traffic to travel through a set of *onion routers* so that each layer of the onion can be peeled off, one by one, for the destination node to obtain the message. Onion routers neither store a network log, nor know who is communicating with whom. For the protocols in this category [2]–[5], an onion is generated by adding encrypted layers during the route discovery phase.

The location-based protocol [5] preserves the anonymity of end hosts by making their locations ambiguous. For instance, ZAP [5] selects two proxies for delegate source and destination nodes as shown in Figure 1. While unicast routing is used in the communications between two proxies, anonymous flooding is applied to the communications within anonymous zones where a proxy and source node or destination node are located. By doing this, the source and destination nodes are not identifiable within the zone. The definition of a zone can be defined by a topology-based zone, such as the number of hops from a node, or by a geographical area including one of the end points.

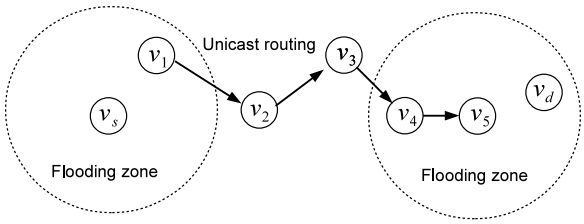

Fig. 1. An example of zone-based routing.

C. Anonymous Routing Protocols in DTNs

The most relevant research is the anonymous routing protocol design in DTNs. ALAR [7] preserves the location privacy of a source node by dividing a message into several segments, and then forwarding them via different neighbors. However, this approach hides the location but not the identity of the source node. A natural approach to preserving node anonymity involves the use of proxies, such as onion routers or pivot. Based on the threshold secret sharing [18], TPS [10] routes a

TABLE I
DEFINITION OF NOTATIONS.

Symbols	Definition
n	The number of nodes in a network
v_i	Node i
$1/\lambda_{i,j}$	The inter-contact time between v_i and v_j
m, σ	A message and an encrypted message
$E(.)/D(.)$	Encryption/decryption functions
L	The number of copies
K	The number of onion routers that a message travels
η	The number of hops between two nodes
R_i	A set of onion routers for the i-th hop
G_i	The size of onion group R_i
G	The average number of nodes in an onion group
$r_{i,j}$	The j-th node in R_i
T, t_i	The end-to-end deadline and the zone i's deadline
τ	The threshold to determine t_i
c	The number of compromised nodes

message through at least τ groups out of s groups, and the last intermediate node serves as a pivot. The difference between TPS and onion-based routing is that the layered encryption is not performed, and thus, the pivot knows the identity of the destination. To the best of our knowledge, the most viable protocols at this moment are group onion-based protocols, such as ARDEN [8] and OGR [9] in which a set of nodes share a secret key to form an onion group, and any node in the same group can encrypt/decrypt the corresponding layer of an onion.

III. PROTOCOL DESIGN

In this section, we first design a set of protocols for DTNs based on anonymous broadcast and the zone-based protocols, which are originally designed for mobile ad hoc networks. These revised protocols, as well as the onion-based protocols, will serve as the building blocks for the proposed FAR protocol introduced in Section IV.

A. Definitions and Assumptions

A DTN is represented by an undirected graph which is constructed from contact histories among nodes. Let v_i be a node i, and two nodes, say v_i and v_j, are connected in a graph if v_i and v_j have at least one contact in the past. The weight of a link between v_i and v_j is given by $\lambda_{i,j}$, where $1/\lambda_{i,j}$ is the inter-meeting time between two nodes v_i and v_j. In [9], [19], the inter-contact time between nodes in a DTN is assumed to be exponential distribution. We adopt this assumption in this paper for the protocol design and analysis. However, we will relax this assumption in the performance section by using the real trace dataset and use this dataset to access the performance of our derived protocol in the real-world DTN scenarios. The probability density function that v_i meets v_j at time t is obtained by $\lambda_{i,j}e^{-\lambda_{i,j}t}$. In addition, the probability that v_i meets v_j within T is computed by:

$$P_{i,j}(T) = \int_0^T \lambda_{i,j}e^{-\lambda_{i,j}t}dt = 1 - e^{-\lambda_{i,j}T} \quad (1)$$

In onion-based routing, a message, denoted by m, travels a set of onions in the specified order by which each layer of an onion is to be peeled off. We denote R_i as the set of nodes for the i-th onion group by which m travels. For convenience, The j-th node in R_i is labeled by $r_{i,j}$, and the size of R_i is G_i. In addition, the average group size is denoted by G.

The nodes in a DTN are assumed to have enough computational power to perform public and private key operations. For cryptographic operations, PK_i and SK_i are defined as the public and private keys of node v_i. In addition, GK_i represents the group key of onion group R_i. The encryption and decryption are denoted by $E(.)$ and $D(.)$.

The notations used in this paper are summarized in Table I.

B. The Attack Model

While the network model in DTNs differs from that of ad hoc networks, the similar security threats such as eavesdropping and traffic analysis are possible in DTNs. For example, an adversary clandestinely stalks a legitimate mobile user to monitor whom the user meets and eavesdrop on wireless channels. Another possible attack is that an adversary blackmails a user to obtain the network log, which contains the information about from/to which node she receives/sends a message.

In this paper, we abstract the aforementioned threats by the compromise attack, where some nodes in a network are marked as being compromised and the message transmissions/receptions are monitored. Then, an adversary reasons possible routing paths and identifies source/destination based on the information disclosed from compromised nodes. Let $\{v_s, r_1, r_2, ..., r_K, v_d\}$ be a path with $K+1$ hops and the link between two relays be $r_k \rightarrow r_{k+1}$. Then, we define the two security attacks as follows.

Attack 1 (The path tracing) *An adversary tries to discover links $v_s \rightarrow r_1$, $r_k \rightarrow r_{k+1}$ for $1 \leq k \leq K-1$, and $r_K \rightarrow v_d$ which constitutes a path as many as possible. Should r_k be compromised, an adversary will be able to find the next relay r_{k+1} by stalking r_k.*

As a privacy metric against Attack 1, the traceable rate [4] can be applied, which is a weighted metric indicating how much portion of a path is disclosed to adversaries when some nodes are compromised. Let η be the number of hops between the source and destination, C_{seg} be the number of compromised segments, and $c_{seg,i}$ be the length of the i-th compromised segments. Then, the traceable rate, denoted as P_{trace}, is defined by Equation 2.

$$P_{trace} = \frac{1}{\eta^2}\sum_{i=1}^{C_{seg}}(c_{seg,i})^2 \quad (2)$$

For example, let $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$ be a routing path where the number of hops is four, i.e., $\eta = 4$. Assume that the link between nodes v_i and v_{i+1} is disclosed to an adversary when v_i is compromised. For instance, when three nodes, v_1, v_3, and v_4, are compromised, the traceable rate will be $\frac{1^2+2^2}{4^2} = \frac{5}{16}$. If three consecutive nodes, v_1, v_2, and v_3, are compromised, the traceable rate will be $\frac{3^2}{4^2} = \frac{9}{16}$. As indicated by these cases, the traceable rate is weighted in the sense

that the greater the length of the consecutive compromised segments, the greater the portion of path that is traceable.

Attack 2 (The node deanonymizing) *An adversary tries to identify v_s and v_d. Should the first onion router r_1 or the last onion router r_K be compromised, the adversary may narrow the anonymous set to which v_s or v_d belongs.*

Anonymity is the state of not being identifiable among an anonymous set. Anonymity is generally modeled as an entropy-based metric [20]. Let ϕ be all the possible elements, and p be the probability that a given element is original. The elements could be nodes and routing paths in our context. The entropy of the system is given by Equation 3.

$$H(\phi) = -\sum_{\forall i \in \phi} p_i \log_2(p_i). \tag{3}$$

When $p_i = p_j$ for all $i, j \in \phi (i \neq j)$, the set of elements is anonymous. For example, assume that 10 nodes exist in an anonymous zone, and one of them is the receiver of a message. If a broadcast scheme is an anonymous protocol, then the receiver is not identifiable among the 10 nodes. In other words, any node in the set has the same probability of being the receiver.

Let ϕ' be a set of suspicious elements in the system (in this case, ϕ' is a set of nodes), $H_{\phi'}$ be the entropy of the system, and H_{max} be the maximal entropy that the system can achieve. Then, the degree of anonymity is defined as $D(\phi') = H_{\phi'}/H_{max}$. Computing p_i in Equation 3 is application-dependent. The definitions of anonymity for source and destination nodes are modeled in Section V-B.

C. Anonymous Epidemic Routing

Let v_s be the node who wishes to deliver message m to destination node v_d. First, v_s encrypts m by v_d's secret key, say PK_d. Let σ be the ciphertext computed by $E(PK_d, m)$. For each message, the message ID, denoted by $m.id$, is defined, which can be either a unique sequence number or a random number. Note that m cannot be deduced from $m.id$. The information about the source node is not included in the header, as to preserve the source anonymity. Such information should be stored in m so that only v_d can tell where the message comes from.

Afterwards, a pair of $m.id$ and σ is sent based on Epidemic routing. Consider that node v_i has m and meets another node v_j. Nodes v_i and v_j check if v_j has σ by exchanging $m.id$. If so, no action will be taken. If it is the first time for v_j to see $m.id$, v_i forwards ($m.id$, σ) to v_j. If the receiver, v_j, is the destination, it decrypts σ by computing $D(SK_d, \sigma)$. Otherwise, v_j continues the Epidemic process. For message m, the end-to-end deadline is defined as $m.T$, which is initialized by parameter T, and m is discarded if the deadline has passed.

In the case of mobile ad hoc networks, v_d will also broadcast m to pretend as if it is not the destination against the location-based deanonymization attack. However, in DTNs, a network is constructed by contact events, and thus, such an attack is not of concern. The pseudo code of anonymous Epidemic routing is described in Algorithm 1.

Algorithm 1 AE(v_s, v_d, m, T)

```
/* v_s does the following */
v_s sets v_d and m.T ← T.
v_s computes σ ← E(PK_d, m).
/* v_i does the following at a contact with v_j */
v_i and v_j establish a secure link.
if v_j has not seen m.id then
    v_i sends (m.id, σ) to v_j.
/* When v_j is v_d, it does the following */
if v_j = v_d then
    v_d obtains m by m ← D(SK_d, σ), return SUCCESS.
/* Error handling */
if m is not delivered within T then
    v_i discards m, and returns FAIL.
```

D. Restricted Epidemic Routing Mode

For the proposed protocol to use anonymous Epidemic routing as a subroutine, we extend Algorithm 1 in the previous subsection as anonymous restricted Epidemic routing (RER). Specifically, not only source and destination nodes, but also any two relay nodes like onion routers, for example, can use anonymous Epidemic routing. One example is to apply Epidemic as a variant of partial flooding, which is used in the zone-based routing.

The first extension is the introduction of a *zone*, where Epidemic routing is performed. Note that the zone in anonymous Epidemic routing is the entire contact graph. In the zone-based protocol for ad hoc networks, an anonymous zone is defined by Euclidean distance or topological distance, i.e., the number of hops. However, the network representation of a DTN does not indicate the physical location of nodes, and so Euclidean distance cannot be applied. For topological distance, a small value of TTL, say two or three hops, is normally used as an anonymous zone. The small TTL value will unfortunately make a protocol susceptible to the topology-based deanonymization attack. Therefore, in order to anonymously control the area of Epidemic zone, the zone deadline, which is denoted by t, is used. Here, the value of t is much smaller than the end-to-end deadline T, but is large enough for a message to reach the expected receiver within the deadline with high probability.

Let v_i be the node with message m, and v_j be the expected receiver. We define τ as the probability that v_j receives m within the zone deadline, t. Here, τ is a system parameter required by v_i, and t is dynamically computed from a given τ. Let $P_{i,j}(t)$ be the probability that v_i and v_j have a contact within t. If we set τ to be $P_{i,j}(t)$ in Equation 1, i.e., the probability that v_i has a contact with v_j within t, the appropriate zone deadline t can then be computed as shown in Equation 4.

$$t = -\frac{ln(1-\tau)}{\lambda} \tag{4}$$

In the case of anycast-like forwarding, i.e., a message transmission from node v_i to any node r in R_k, we may set λ to be $\sum_{\forall r \in R_k} \lambda_{i,r}$.

The second extension is the introduction of a group, where any node in the next group can serve as a relay. Let $v_i \in R_k$ be the node which wishes to relay message m to any node

Algorithm 2 RER(v_i, R_{k+1}, σ_k, τ, T)

/* $v_i \in R_k$ does the following */
v_i sets $m.t_{k+1}$ from τ.
/* v_i does the following at a contact with v_j */
v_i and v_j establish a secure link.
if v_j has not seen $m.id$ **then**
  v_i sends ($m.id$, σ_k) to v_j.
  if $v_j \in R_{k+1}$ **then**
    v_j computes $\sigma_{k+1} \leftarrow D(SK_{GK_{k+1}}, \sigma_k)$.
    return SUCCESS;
/* Error handling */
if m is delivered within neither $m.t_k$ nor $m.T$. **then**
  v_i discards σ_k from its buffer.

$r_{k,j} \in R_{k+1}$. At every contact between v_i and v_j, v_j checks if it has seen $m.id$ before. If so, they ignore the message and do nothing. Otherwise, v_i sends σ_k to v_j. Then, Epidemic routing is repeated until the zone deadline, $m.t$, has expired. If $v_i \in R_k$, it identifies itself as a next-relay by the group ID, denoted by g_{id}. Using the corresponding group key of R_{k+1}, denoted by GK_{k+1}, v_j peels off a layer of encrypted message, i.e., $\sigma_{k+1} \leftarrow D(GK_{k+1}, \sigma_k)$. If either a zone or end-to-end deadline has passed, m is discarded.

The pseudocode of RER is presented in Algorithm 2.

E. Zone-Based Anonymous DTN Routing

A zone-based anonymous DTN routing (ZBAR) can be constructed from Epidemic and spray-and-wait protocol, each of which is replaced with partial flooding and unicast routing (e.g., geographical routing). That is, Algorithm 2 is used for message transmission from the source to its proxy and from the destination proxy to the destination. Between the proxies, source/binary spray-and-wait is used.

An anonymous spray-and-wait forwarding between two proxies is basically the same as the one used between two intermediate relays in onion-based routing. Based on these ideas, we construct a zone-based anonymous DTN routing (ZBAR), as follows.

The source node v_s selects the source and destination proxies, say r_s and r_d, respectively. Then, $\sigma_d \leftarrow E(PK_d, m)$, $\sigma.r_d \leftarrow E(PK_{r_d}, \sigma_d)$, and $\sigma.r_s \leftarrow E(PK_{r_s}, \sigma_{r_d})$ are computed. The encryption structure is the same as that of an onion, where v_d can decrypt the encrypted data after r_s and r_d peel off the outer layers. In addition, $m.id$ and $m.t$ are calculated. A message is composed of ($m.id$, $m.t$, $m.T$, $mode$, $\sigma.r_s$). The value of $mode$ could be either the restricted epidemic RE or spray-and-wait SW forwarding mode.

From v_s to r_s, restricted Epidemic routing is performed. A receiving node first attempts to decrypt σ_{r_s}. If it fails, the node is not the proxy, and the Epidemic process continues as long as $m.t$ has not expired. Otherwise, r_s decrypts the outmost layer of the onion, and it switches the mode of the message to the spray-and-wait forwarding mode. From r_s to r_d, a message ($m.id$, $m.T$, $mode$, $\sigma.r_d$) is forwarded by anonymous spray-and-wait with single-copy forwarding. When the destination proxy, r_d, receives the message, the corresponding layer of $\sigma.r_d$ is decrypted, and $m.t$ is computed. Then, the restricted Epidemic routing for message ($m.id$, $m.t$, $m.T$, $mode$, $\sigma.d$) is again performed. The destination identifies itself by successfully decrypting σ_d using SK_d. The pseudo code of ZBAR is presented in Algorithm 3.

Algorithm 3 ZBAR(v_s, v_d, m, T)

/* v_s does the following */
v_s selects two proxies, r_s and r_d.
v_s computes $\sigma_d \leftarrow E(PK_d, m)$,$\sigma.r_d \leftarrow E(PK_{r_d}, \sigma_d)$, and $\sigma.r_s \leftarrow E(PK_{r_s}, \sigma_{r_d})$.
v_s sets $m.T$, $m.t$, and $m.mode \leftarrow RE$.
v_s executes Algorithm 2 RER(v_s, $\{r_d\}$, m, $m.t$).
/* r_s meets node v_i. */
v_i and v_j establish a secure link.
if v_i identifies itself as r_d **then**
  v_i computes $\sigma_d \leftarrow D(SK_{r_d}, \sigma.r_s)$.
  v_i sets $m.t$ and $m.mode \leftarrow RE$.
  v_i executes Algorithm 2 RER(r_d, $\{v_d\}$, m, T).
if $m.t$ expires **then**
  v_i removes m from its buffer.
/* v_d does the following */
v_d obtains m by $m \leftarrow D(SK_d, \sigma)$, return $SUCCESS$.
/* Error handling */
if m is not delivered in T **then**
  v_i discards m, and returns $FAIL$.

IV. Framework of Anonymous Routing

A. Motivation and Basic Idea

We first point out two problems regarding the existing anonymous routing with onion-based [8], [21] and threshold-based [10] schemes for DTNs. The first issue is that the source (or the destination) node is anonymous only within its onion group. Hence, the identity of a source or destination node will be revealed if the first or the last onion router is compromised. The second issue is that an intermediate onion router knows the previous and subsequent onion routers. These problems significantly reduce the node anonymity and the path untraceability.

To alleviate the first problem, we have proposed the ZBAR protocol based on zone-based routing [5] in Section III. However, the second issue still remains unresolved with the zone-based approach. To preserve anonymity, an intermediate onion router should not know the exact previous and next forwarding nodes. In addition, the first and last onion routers should not know they are located at the edge of an onion path.

To achieve these desirable properties, we propose a Framework for Anonymous Routing (FAR) for DTNs that subsumes all the anonymous routing protocols. That is, the source node sets up a set of onion routers, and then all nodes on the path forward a message with the restricted Epidemic routing. Note that the proposed FAR does not just combine different anonymous routing protocols, but creates a framework that subsumes all the protocols. In other words, FAR serves as either an anonymous Epidemic, ZBAR, or onion-based protocol, when its parameters are set differently. By adjusting the parameters appropriately, FAR enjoys the advantages of all these anonymous routing protocols.

B. The Protocol Overview

In this section, we describe the high-level overview of the proposed FAR. Let v_s be the source node which wishes to

Algorithm 4 FAR(v_s, v_d, m, K, L, G, F, T, τ)

1: /* v_s does the following */
2: v_s selects K onion groups.
3: v_s computes $\sigma_0 \leftarrow E(PK_d, m)$.
4: v_s computes $\sigma_i \leftarrow E(GK_{R_i}, \sigma_{i-1})$ for $1 \leq i \leq K$.
5: v_s sets $\sigma_1.t_1$.
6: v_s executes $RER(v_s, R_1, \sigma_1, T)$.
7: /* On receiving σ_k from $v_j \in R_{k-1}$, $v_i \in R_k$ does the following */
8: **if** $v_i \in R_k$ receives σ_k from $v_i \in R_{k-1}$ **then**
9: **if** v_i identifies itself as v_d **then**
10: v_d obtains m by $m \leftarrow D(SK_d, \sigma_k)$.
11: returns $SUCCESS$.
12: **else**
13: v_i sets $\sigma_k.t_k$.
14: **if** $\sigma_k.f_k$ is RE **then**
15: /* Restricted Epidemic mode */
16: v_i executes $RER(v_i, R_{k+1}, \sigma_k, T)$.
17: **else if** $\sigma_k.f_k$ is SW **then**
18: /* Anonymous spray-and-wait mode */
19: v_i forwards σ_k when it has a contact $r \in R_{k+1}$ if r has not seen σ_k.
20: /* Error handling */
21: **if** m is not delivered in T **then**
22: v_i discards m, and returns $FAIL$.

deliver message m to destination v_d. The routing parameters, $\{K, L, G, F\}$, are selected by v_s, where K is the number of onion relays that m shall travel, L is the number of copies, G is the size of the onion group, and $F = \{f_1, f_2, ..., f_K\}$ is a set of forwarding modes. A forwarding mode can be either restricted Epidemic RE or source spray-and-wait SW.

After initializing the routing parameters, v_s randomly selects a set of K onion groups ($K \geq 1$), along which m travels and creates an onion. How to forward m from one node to another differs, depending on the forwarding mode utilized. In the RE mode, a node, say v_i, with m sends a copy to all the nodes contacted by v_i within the zone deadline. In the SW mode, a node with m sends a copy to any node in the next onion group as long as the tickets (the number of copies allowed to duplicate) are available. The forwarding mode for the i-th hop is determined by f_i. When a node, say r_j, in the next onion group R_{i+1} receives m, the outer layer of the onion is peeled off by the corresponding group key. Then, the forwarding process continues based on the forwarding mode specified in f_{i+1}. This process is repeated until the destination v_d receives m.

C. Framework of Anonymous Routing

To initialize the anonymous network system, an approach for onion group routing, proposed in [8], can be used. The nodes in a network are divided into $\lceil n/G \rceil$ groups, where G is the average number of nodes in a group. For simplicity, we assume n to be divisible by G. Nodes in the same group are assumed to be able to encrypt/decrypt the corresponding layer of an onion by a common secret or public/private keys. Note that the header size is generally much smaller than the amount of data which can be transmitted during a contact, e.g., the message header size is in the order tens of bytes, while the most of contact durations between mobile devices with bluetooth and WiFi capabilities is approximately 250 seconds in CRAWDAD dataset [11]. As a consequence, such overhead is not considered.

The pseudo code of FAR is provided in Algorithm 4. As inputs, the system parameters $\{K, L, G, F\}$ and the end-to-end deadline, T, are selected by v_s. Lines 1 to 5 represent the initialization phase. The source node, v_s, randomly selects a set of onion groups by which m travels. First, v_s obtains σ_0 by computing $E(PK_d, m)$ with v_d's public key. Then, an encrypted onion is created by applying a set of group keys associated with R_i, i.e., $\sigma_i \leftarrow E(GK_{R_i}, \sigma_{i-1})$ for $1 \leq i \leq K$. Finally, v_s sets the timer, denoted as $\sigma.t$ by Equation 1.

The forwarding process at the k-th Epidemic zone is shown from Lines 7 to 19. For each zone, RER or spray-and-wait forwarding is executed until m reaches v_d. During the RE forwarding mode, σ is discarded if the zone deadline $\sigma.t$ expires. When the destination node, v_d, receives σ, it applies its private key to obtain the original message, m. If the destination does not obtain m by the deadline T, the routing process fails.

FAR subsumes Epidemic, zone-based, and onion-based anonymous routing protocols. The parameters ($K = 0, null, null, S = \{RE\}$) indicate an AE protocol, in which Epidemic is performed by hiding the source and destination nodes. In the case of ($K, L, G, \{f_1 = SW, f_2 = SW, ..., f_K = SW\}$), the protocol is reduced to onion-based routing. In addition, depending on G and L, the protocol can be onion ($G = 1$) or onion group ($G \geq 2$) routing with single/multi copies ($L = 1$ or $L \geq 2$). The configuration of ($K = 2, L = 1$, G, $\{f_1 = RE, f_2 = SW, ..., f_{K-1} = SW, f_K = RE\}$) serves as the ZBAR protocol.

V. Security Analyses

In this section, analytical models are built for traceable rate and node anonymity of the proposed FAR under Attacks 1 and 2, respectively. Our analysis provides the closed form solutions to different metrics, by which DTN users select the system parameters that meet their security and privacy requirements. Note that the analyses of AE (Algorithm 1) and ZBAR (Algorithm 3) are trivial and thus omitted. In the following discussions, the uniform distribution is used for compromised nodes.

A. Traceable Rate

The traceable rate is computed by Equation 2 against the path tracing attack defined in Attack 1. The proposed FAR employs anonymous Epidemic forwarding, and the path can be revealed only by the reverse order from the destination. Thus, the number of compromised segments C_{seg} in Equation 2 equals either 0 or 1. Let X be the random variable that represents the length of the compromised segments $c_{seg,1}$, then $E[X]$ can be computed by the geometric distribution with the limited number of trials. The probability of a node being

compromised is c/n. Denoting $p = 1 - c/n$ and $q = c/n$, $E[X]$ can be obtained as follows:

$$E[X] = \sum_{i=1}^{\eta} iq^{i-1}p + \eta q^{\eta} = qE[X] + \sum_{i=1}^{\eta} q^{i-1}p + \eta q^{\eta} \quad (5)$$

By defining $\epsilon_1 = \sum_{i=1}^{\eta} q^{i-1}p$ and $\epsilon_2 = \eta q^{\eta}$, we will have

$$E[X] = \frac{n(\epsilon_1 + \epsilon_2)}{n - c}. \quad (6)$$

Since the traceable rate is weighted, we need to compute $E[X^2]$, which can be obtained as follows:

$$E[X^2] = \sum_{i=1}^{\eta} i^2 q^{i-1}p + \eta^2 q^{\eta} \quad (7)$$

$$= qE[X^2] + 2qE[X] + \sum_{i=1}^{\eta} q^{i-1}p + \eta q^{\eta} \quad (8)$$

$$= \frac{n(n+c)(\epsilon_1 + \epsilon_2)}{(n-c)^2} \quad (9)$$

Since $(c_{seg,1})^2 = E[X^2]$, the traceable rate is computed by $\frac{1}{\eta^2}E[X^2]$, and therefore, we derive Equation 10.

$$P_{trace} = \frac{1}{\eta^2}\left\{\frac{n(n+c)(\epsilon_1 + \epsilon_2)}{(n-c)^2}\right\} \quad (10)$$

The number of hops, η, increases in proportion to the value of the number of onion routers, K. This is because all the messages must travel at least one onion relay in a particular onion group, in the predefined order. Thus, in a high-level view, we can consider that one hop from an onion router to the next onion router is a link. For a DTN user to find an appropriate routing parameter K, we may simply set η to be $K + 1$.

B. Source and Destination Anonymity

How to quantify anonymity is application-dependent, and thus, we model source and destination anonymity as follows. In FAR, the anonymity of source and destination nodes are computed in the same way, and only two parameters, the number of nodes n and the number of compromised nodes c, are related to this metric. In the case of a node not being compromised, the node is identified among the non-compromised nodes with the probability of $1/(n-c)$. Thus, the maximal entropy of a node is defined as

$$H_{max}^{node} = -\sum_{\forall \text{nodes in } \phi} \frac{1}{n-c}\log_2\left(\frac{1}{n-c}\right). \quad (11)$$

If a node is compromised, it is identified with 100% probability. In other words, the entropy of a node is always 0 if the node is compromised. Otherwise, it is still anonymous among the set with size $1/(n-c)$. Let ϕ' be a set of suspicious nodes. The entropy of a node, denoted by H_{ϕ}^{node}, is obtained by:

$$H_{\phi'}^{node} = -\sum_{\forall \text{nodes in } \phi'} (1 - \frac{c}{n}) \cdot \frac{1}{n-c}\log_2\left(\frac{1}{n-c}\right). \quad (12)$$

TABLE II
SIMULATION PARAMETERS FOR RANDOM GRAPHS.

Parameter	Value (default value)
The number of nodes	200
The inter-contact time	0 to 720 unit time
The group size	10 or 20
The number of onion routers	3
The number of copies	10 or 20
The message due	10 to 2,000 unit time
The threshold of RER	0.8 to 0.99
The % of compromised nodes	0% to 50% (10%)

Here, $|\phi'| = n - c$. Therefore, we will derive the anonymity of a node as follows:

$$D_{node}(\phi') = \frac{H_{\phi'}^{node}}{H_{max}^{node}} = 1 - \frac{c}{n} \quad (13)$$

VI. PERFORMANCE EVALUATION

In this section, computer simulations are conducted to evaluate the performance of the proposed scheme. A set of the proposed protocols, AE, ZBAR, FAR, as well as the existing, onion-group routing (OGR) [9] are implemented. To the best of our knowledge, OGR in [9] is the latest and the most viable anonymous routing protocol for DTNs. Note that AE, ZBAR, and OGR are special cases of FAR. For simplicity, we refer to them as AE, ZBAR, and OGR instead of FAR with specific parameters.

A. Simulation Configurations

In most of DTN researches, random graphs as well as real traces are used for mobility models [9], [19]. In our simulations, two scenarios are considered. One is a randomly generated contact graph for evaluating the proposed schemes in large-scale DTNs; the other is a real contact trace [11] to demonstrate that our FAR works well in realistic environments.

Random graphs - A contact graph with 200 nodes is generated by assigning inter-contact times to each pair of two nodes. redNote that the number of nodes is fixed, as it does not directly affect the security and privacy metrics. The inter-contact time is exponentially distributed with parameter $\lambda_{i,j}$ for a pair of nodes v_i and v_j ($i \neq j$). The initial value of $1/\lambda_{i,j}$ is generated by the normal distribution in which the mean and variant are set to be 360 and 720 time units, respectively. The group size is set to be 10 or 20, the number of onion routers is set to be 3, and the number of copies is set to be the same as the group size (i.e., $L = g$). The message deadline T is randomly selected in the range of 10 to 2,000 time units, and the percentage of compromised nodes is set to be $0\% \leq c/n \leq 50\%$, where c is the number of compromised nodes and n the total number of nodes. The simulation parameters are summarized in Table II.

The source and destination nodes are randomly selected, and each node runs an anonymous routing protocol with given parameters. If a message is delivered from source to destination within the deadline, T, the message delivery is successful. For a given percentage of compromised nodes, i.e., c/n, randomly selected nodes of such a portion are marked

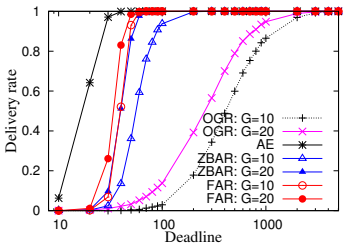

Fig. 2. The CDF of delivery rate.

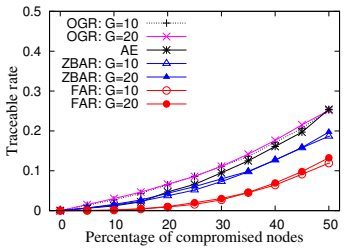

Fig. 3. The traceable rate.

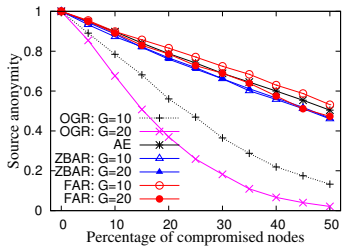

Fig. 4. The source anonymity.

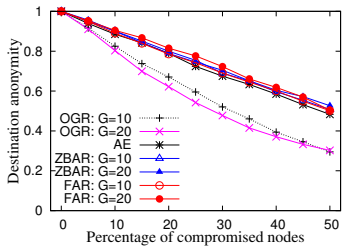

Fig. 5. The destination anonymity.

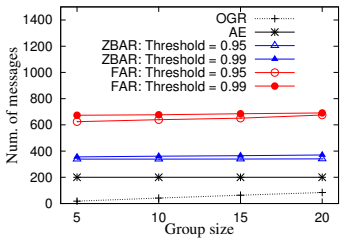

Fig. 6. The number of messages.

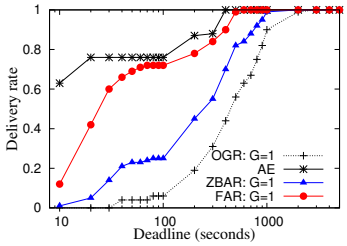

Fig. 7. The delivery rate w/ the Cambridge trace.

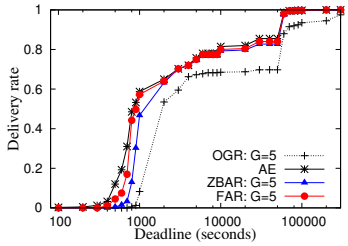

Fig. 8. The delivery rate w/ the Infocom 2005 trace.

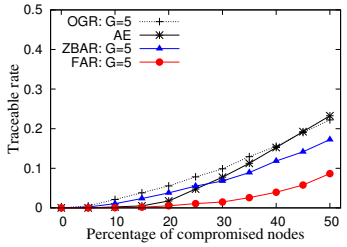

Fig. 9. The traceable rate w/ the Infocom 2005.

as compromised, and then security metrics are computed. For each set of parameters, 1,000 contact graphs are generated for the simulation.

Real traces - CRAWDAD dataset cambridge/haggle [11] contains a set of contact trace experiments. In our simulations, Experiments 2 and 3, the so-called Cambridge and Infocom 2005 traces, are used as inputs. In these scenarios, we only consider the contacts between mobile nodes, i.e., iMotes, and omit contacts among stationary nodes and external devices. There are 12 and 41 mobile nodes in the Cambridge and Infocom 2005 traces, respectively. Each piece of contact information contains two node IDs, the time that the two nodes meet, the time that they lose a connection, the number of contact times, and the elapse time of the last time the two nodes met. Contact events are recorded in the order of seconds. Since the contact events are traced over three to five days, there exist time periods in which there is no contact, e.g., off-business hours and night time. Thus, a source node is assumed to initiate a message transmission at any time after it has a contact with any node, which implies that message delivery starts during business hours, but not at night time.

For a given trace file, the number of nodes and inter-meeting times are calculated. The other simulation parameters, i.e., K, L, G, c, and T are set in the same way as the random graphs. For each trace file, 500 different sets of source, destination, and intermediate onion routers are randomly selected, and the average performance is computed.

B. Results Using Synthesize Graphs

Figure 2 shows the cumulative distribution function (CDF) of the delivery rate with respect to the deadline. AE results in the fastest delivery, and the CDF of FAR reaches 0.95 within 70 time units. This indicates that Epidemic-based routing delivers a message much faster than does OGR. ZBAR incurs slightly longer delay than AE and FAR, since it forwards a message by the stop-and-wait between the first and last onion routers. In addition, it is intuitive that a larger group size leads to faster message delivery, and this can be clearly observed in this figure.

Figure 3 illustrates the traceable rate with respect to the percentage of compromised nodes. Since every path is considered independently, the group size does not affect the traceable rate. In addition, it is intuitive that the traceable rate gradually increases as the percentage of compromised nodes increases. In the proposed FAR, a routing path can be traced only by the consecutive compromised segments from the destination node, and thus, the traceable rate is much lower than that of the other protocols. From the figure, the traceable rate resulting from FAR is at most half of that by OGR. Similar to OGR, ZBAR forwards a message between intermediate onion routers by spray-and-wait forwarding. As a result, the traceable rate of ZBAR is higher than that of FAR, but smaller than that of OGR.

Figures 4 and 5 illuminate the source and destination anonymity with respect to the percentage of compromised nodes. In OGR, the large group size results in low source and destination anonymity due to its design issue. On the contrary, the source and destination anonymity resulting from FAR is independent of the group sizes, since each of the communications between onion routers is performed by the RER (Algorithm 2). This indicates that the onion routers are indistinguishable if they are the first/last onion routers, or the intermediate ones. Hence, unless the source and destination nodes are compromised, adversaries cannot confine the anonymous set in which the source/destination is included. Similarly, AE reveals no information about the identity of source and destination nodes unless they are compromised. In ZBAR, the size of the anonymous set to which the source/destination belongs decreases if at least one of the nodes in the first/last onion groups is compromised. Therefore, ZBAR results in slightly smaller node anonymity than FAR and AE. For OGR, the destination anonymity is better than the source anonymity. This is because the destination can be ambiguous in identifying the onion group as the destination, as proposed in [8]; however,

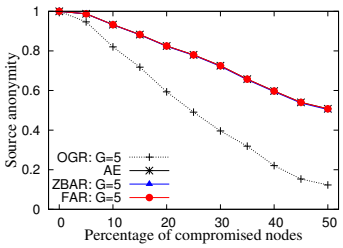

Fig. 10. The source anonymity w/ the Infocom 2005 trace.

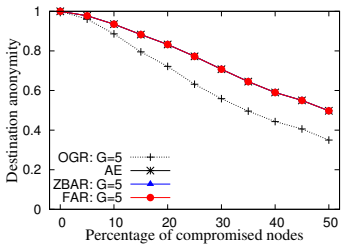

Fig. 11. The destination anonymity w/ the Infocom 2005.

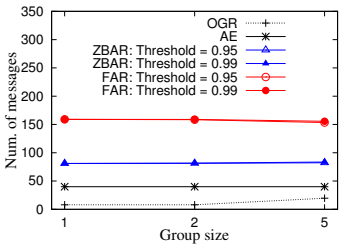

Fig. 12. The number of messages w/ the Infocom 2005.

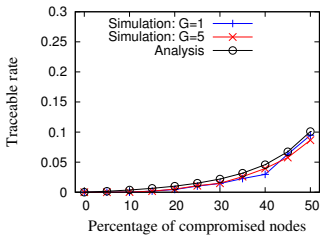

Fig. 13. The traceable rate analysis w/ the Infocom 2005..

this technique cannot be applied to the source node.

Figure 6 depicts the amount of message forwarding, introduced by anonymous protocols with respect to the size of onion groups. Note that AE does not use intermediate onion routers, and so it is independent of the group size. Apparently, Epidemic-based, i.e., AE, ZBAR, and FAR, incur more message overhead than OGR. FAR introduces the greatest amount of message forwarding, as it forwards a message by RER (Algorithm 2) at every communication between two onion routers. However, we claim that achieving the highest privacy in terms of the traceable rate and node/path anonymity with FAR is still worth a large amount of control overhead.

C. Results Using Real Traces

The Cambridge trace, i.e., Experiment 2 in [11] is relatively small-scale and dense (12 mobile nodes), and thus, the number of onion routers and the group size are set to be $K = 3$ and $G = 1$, respectively. The number of copies in OGR and in the stop-and-wait mode in ZBAR are set to be $L = G$. Note that having more than one copy in OGR and ZBAR does not help message delivery when $G = 1$. On the other hand, the Infocom 2005 trace (i.e., Experiment 3 in [11]) is a medium-sized contact network with 41 mobile nodes. The number of onion routers, the group size, and the number of copies are set to be $K = 3$, $G = 5$, and $L = G$, respectively.

Figures 7 and 8 show the delivery rate for different protocols resulting from the Cambridge and Infocom 2005 traces, respectively. In Figure 7, the proposed FAR achieves faster delivery than ZBAR and OGR. In addition, the message delivery is mostly completed within 1,000 seconds, which is much faster than the results shown in Figure 8. This is because the Cambridge trace is generated by the students and faculty members of the same lab group, and there is a landmark where they meet very often.

The Infocom 2005 trace contains fewer contact events than the Cambridge trace. The x-axis of Figure 8 is scaled longer. As can be seen from the figure, the delivery rate of all the protocols increases toward 1,000 seconds, and then a stable period is observed from 5,000 to 30,000 seconds. This implies that there are no contact events during business off-hours. The delivery rate of all the protocols reaches 99% around 60,000 seconds (approximately 16.5 hours), and most of the message transmissions are likely to go through business off-hours. OGR always results in smaller delivery rate than the other protocols, and no significant difference between AE and FAR can be seen. ZBAR incurs a slightly longer delay than

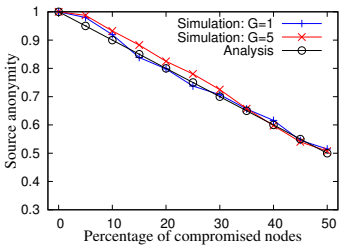

Fig. 14. The source anonymity analysis w/ the Infocom 2005..

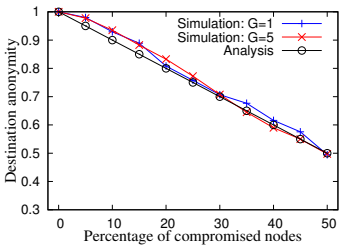

Fig. 15. The destination anonymity analysis w/ the Infocom 2005..

AE and FAR, as it uses the onion-based forwarding between source and destination proxies.

Figure 9 presents the traceable rate using the Infocom 2005 trace with respect to the percentage of compromised nodes. Note that traceable rate is independent of the inter-meeting time among nodes. As can be seen in the figure, the traceable rate of FAR is at least half of AE, ZBAR, and OGR when 50% of the nodes are compromised.

Figures 10 and 11 illustrate the source and destination anonymity resulting from the Infocom 2005 trace. The node anonymity of AE, ZBAR, and FAR linearly decreases when the percentage of compromised nodes increases. Since the contact trace is not large, i.e., the Infocom 2005 trace contains 41 mobile nodes, the difference among AE, ZBAR, and FAR is not significant. On the other hand, OGR always results in smaller source and destination anonymity than the other protocols.

Figure 12 depicts the number of messages for different protocols resulting from the Infocom 2005 trace. While OGR results in the smallest message overhead, its delivery rate is not acceptable as shown in Figure 8. FAR and ZBAR introduce more redundant message forwarding than AE and OGR do. However, we stress that they provide lower traceable rate and high node anonymity. Since the trace is relatively a small scale network containing 41 mobile nodes, the difference value of the thresholds does not affect the message overhead.

D. Comparisons Between Simulation and Analysis

Figure 13 shows the traceable rate resulting from simulations and analysis. Note that, according to our analysis, the traceable rate is independent of the size of onion groups, and thus simulation results with different group sizes are very close to each other. This figure demonstrates that the analytical result provides a very close approximation for the traceable rate.

Figures 14 and 15 provide the source and destination anonymity resulting from simulations and analyses. As the

proposed analysis indicates, both the source and destination anonymity decrease as the number of compromised nodes increases. In addition, a significant difference between different group sizes is not observed, since the node anonymity is independent of the size of onion groups.

VII. Considerations on Parameter Selection

In this section, we discuss how to select parameters that satisfy a given security, performance, and cost requirements. The first consideration is a set of forwarding modes, SW and RE. There is no obvious advantage of using SW except smaller amount of message overhead. Thus, RE mode should be applied for achieving both the faster delivery and higher degree of privacy as long as the message overhead is acceptable.

There are three tunable parameters: the number of message copies L, onion group size G, and number of intermediate onion routers K. Among them, G and K are specified by the network administrator. The centralized setting of G is required to initialize the public/private keys. In many scenarios, K is a constant, e.g., $K = 3$ in Tor [1]. Even in wired communications, the use of onion routers significantly reduces the throughput, and thus, the value of K should not be greater than three for DTNs. The value of L is tunable by users for each message transmission request, and $L \leq G$ holds because letting $L > G$ has no effect. As a rule of thumb, the larger value of L improves the delivery rate. Note that our analysis in Section V implies that the traceable rate and source/destination anonymity are independent from L, although they slightly affect these metrics in simulations. Thus, in the following, we discuss how users can select a proper value of L based on their performance requirement subject to given acceptable message overhead.

Recall that n is the number of nodes in a network and c is the number of compromised nodes. For given parameters K and G specified by the administrator, the function of the message overhead, denoted by $C(L, K, G, n)$, is defined by

$$C(L,K,G,n) \leq \begin{cases} LG(K+1) & \text{for OGR} \\ n & \text{for AE} \\ 2nLG(K-1) & \text{for ZBAR} \\ nLG(K+1) & \text{for FAR.} \end{cases} \tag{14}$$

Let M be the acceptable number of forwarded messages. The desirable value of $L \leq G$ to maximize the delivery rate can be obtained by introducing the number of copies L, subject to $C(L, K, G, n) \leq M$.

VIII. Conclusion

In this paper, we first construct anonymous Epidemic (AE) and zone-based routing protocols (ZBAR) for DTNs by porting the existing solutions designed for ad hoc networks. Then, we design a framework for anonymous routing (FAR) that subsumes all the Epidemic, zone-based, and onion-based routing. By tuning parameters, the proposed FAR enjoys the advantages of these protocols, but at the same time offsets disadvantages. With this design, FAR accommodates compatibility problems among DTNs with different routing policies, and thus, it can be deployed to DTNs with different security and anonymous requirements with ease. In addition, quantitative analyses are studied in terms of node anonymity as well as traceable rate. Furthermore, the extensive simulations resulting from randomly generated graphs as well as one of the well-known real traces called CRAWDAD dataset Cambridge/haggle demonstrate that the proposed scheme outperforms the existing solutions. Moreover, simulations and numerical results are compared and validated by each other. We believe our framework serves the foundation of anonymous routing for many types of contact-based networks.

Acknowledgments

This research has been funded in part by JSPS KAKENHI Grant Number JP17K12675 and by the U.S. National Science Foundation grants IIS-1618669 (III) and ACI-1642133 (CICI).

References

[1] M. Backes, A. Kate, S. Meiser, and E. Mohammadi, "(nothing else) MATor(s): Monitoring the Anonymity of Tor's Path Selection," in *CCS*, 2014, pp. 513–524.

[2] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti, "Secure Pebblenets," in *Mobihoc*, 2001, pp. 156–163.

[3] Y. Zhang, W. Liu, and W. Lou, "Anonymous Communications in Mobile Ad Hoc Networks," in *Infocom*, 2005, pp. 1940–1951.

[4] J. Kong and X. Hong, "ANODR: Anonymous on Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks," in *Mobihoc*, 2003, pp. 291–302.

[5] X. Wu and E. Bertino, "An Analysis Study on Zone-Based Anonymous Communication in Mobile Ad Hoc Networks," *IEEE Trans. Dependable Secur. Comput.*, vol. 4, no. 4, pp. 252–265, 2007.

[6] D. Goldschlag, M. Reed, and P. Syverson, "Onion Routing," *Commun. of ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[7] X. Lu, P. Hui, D. Towsley, J. Pu, and Z. Xiong, "Anti-localization Anonymous Routing for Delay Tolerant Network," *Comput. Netw.*, vol. 54, no. 11, pp. 1899–1910, 2010.

[8] C. Shi, X. Luo, P. Traynor, M. H. Ammar, and E. W. Zegura, "ARDEN: Anonymous Networking in Delay Tolerant Networks," *Ad Hoc Netw.*, vol. 10, no. 6, pp. 918–930, 2012.

[9] K. Sakai, M.-T. Sun, W.-S. Ku, J. Wu, and F. S. Alanazi, "An Analysis of Onion-Based Anonymous Routing for Delay Tolerant Networks," in *ICDCS*, 2016, pp. 609–618.

[10] R. Jansen and R. Beverly, "Toward Anonymity in Delay Tolerant Networks: Threshold Pivot Scheme," in *MILCOM*, 2010, pp. 587–592.

[11] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.org/cambridge/haggle/20090529, May 2009.

[12] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Tech. Rep., 2000, CS-200006, Duke University.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *SIGCOMM Workshop on Delay-Tolerant Networking*, 2005, pp. 252–259.

[14] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," in *SIGCOMM*, 2004, pp. 145–158.

[15] C. Liu and J. Wu, "An Optimal Probabilistic Forwarding Protocol in Delay Tolerant Networks," in *Mobihoc*, 2009, pp. 105–114.

[16] W. Gao, Q. Li, and G. Cao, "Forwarding Redundancy in Opportunistic Mobile Networks: Investigation and Elimination," in *Infocom*, 2014, pp. 2301–2309.

[17] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN Routing As a Resource Allocation Problem," in *SIGCOMM*, 2007, pp. 373–384.

[18] A. Shamir, "How to Share a Secret," *Commun. of ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[19] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, "Supporting Cooperative Caching in Disruption Tolerant Networks," in *ICDCS*, 2011, pp. 151–161.

[20] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards Measuring Anonymity," in *PET*, 2002, pp. 54–68.

[21] G. Vakde, R. Bibikar, Z. Le, and M. Wright, "EnPassant: Anonymous Routing for Disruption-Tolerant Networks with Applications in Assistive Environments," *Security and Commun. Netw.*, vol. 4, no. 11, pp. 1243–1256, 2011.