An Analysis of Onion-Based Anonymous Routing for Delay Tolerant Networks

Kazuya Sakai*, Min-Te Sun†, Wei-Shinn Ku‡, Jie Wu§, and Faisal S. Alanazi¶
*Dept. 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.
§Department of Computer and Information Sciences, Temple University, 1925 N. 12th St. Philadelphia, PA 19122.
¶Department of Electrical and Computer Engineering, The Ohio State University, 2015 Neil Ave., Columbus, Ohio 43120.
ksakai@tmu.ac.jp, msun@csie.ncu.edu.tw, weishinn@auburn.edu, jiewu@tmple.edu, alanazi.3@buckeyemail.osu.edu

***Abstract*—Delay tolerant network (DTN) routing provides a communication primitive in intermittently disconnected networks, such as battlefield communications and human-contact networks. In these applications, the anonymity preserving mechanism, which hides the identities of communicating parties, plays an important role as a defense against cyber and physical attacks. While anonymous routing protocols for DTNs have been proposed in the past, to the best of our knowledge, there is no work that emphasizes the theoretical aspects. In this paper, we first design an abstract of anonymous routing protocols for DTNs and augment the existing solution with multi-copy message forwarding. Then, we construct simplified mathematical models, which can be used to understand the fundamental performance and security guarantees of onion-based anonymous routing in DTNs. The numerical and simulation results using randomly generated contact graphs and the real traces demonstrate that our models provide very close approximations to the performance of the anonymous DTN routing protocol.**

***Index Terms*—Anonymous communications, onion routing, delay tolerant networks, DTNs.**

I. Introduction

Delay tolerant network (DTN) routing [1]–[7] enables message delivery in intermittently disconnected networks such as battlefield communications, bus-to-bus networks [8], PeopleNet [9], pocket switched networks [10], and so on. In these DTN applications, not only is improving the message delivery and minimizing the forwarding cost important, but providing security and privacy preserving mechanisms are also both theoretically and practically important.

While the messages exchanged between two nodes can be protected with end-to-end encryption, a large amount of information, including node identifiers, the locations of end hosts, and routing paths, may be revealed by traffic analyses. It is crucial to protect these types of sensitive information in critical communication environments. For instance, in a battlefield, one of the communicating end hosts is most likely to be a commander, and thus, disclosing the location of the end host will likely result in a mission failure. As a defending mechanism, anonymous communications [11] are widely studied to hide where a message comes from and where it goes to, as well as the identities of communicating end hosts.

Although significant efforts have been made to design anonymous routing protocols for the Internet [12], [13] and ad hoc networks [14]–[21], these approaches are not appropriate for DTNs due to key differences between them. First, the graph representation of a DTN is contact-based, while that of an ad hoc network indicates the physical topology formed by nodes. Second, neither stable end-to-end communication links nor transmission opportunities are assumed due to the fact that intermittent connectivity is very limited. Third, a DTN routing is implemented in the *Bundle* layer which is located between the transport and application layers. These factors make the existing ad hoc anonymous routing protocol unfeasible in such a network, and therefore, these are the reasons that we again study anonymous communications primarily designed for DTNs.

To the best of our knowledge, very few anonymous routing protocols are designed for DTNs. One of the well-known DTN routing protocols to preserve anonymity is onion routing [22], where layered encryption, each by different secret keys, is applied to a message, and each layer can be peeled off with the corresponding secret key. To accommodate the limited forwarding opportunities, the idea of onion groups is introduced in [23]–[25], in which a set of nodes form an onion group so that any node in the same onion group is able to encypt/decrypt the corresponding layer. However, theoretical analysis is yet to be done. Therefore, we are interested in the theoretical aspects of onion-based anonymous routing in DTNs.

The goal of this paper is to build performance and security models for anonymous communications in DTNs. To this end, we first design an abstract anonymous routing protocol based on onion-based routing protocols proposed in [24], [25]. Our simplified protocol captures the essence of understanding the performance and security issues of anonymous DTN routing, and in addition, it can be easily extended to auxiliary protocols. The main contributions of this paper are listed as follows.

First, we propose an onion-based anonymous routing with multi-copy forwarding in which L copies of a message are allowed. Note that the proposed protocol can be considered as the generalization of the existing protocol [25], and no existing anonymous routing for DTNs considers the case of multiple copies. Then, we build analytical models for the delivery rate by defining *opportunistic onion path*. The key difference from the existing model, called *opportunistic path* [26], is that an anycast-like property is incorporated, i.e., a node can forward a message to any node in the next onion group. We provide

the bound of the message transmission cost introduced by anonymous DTN routing, which is defined as the factor of the shortest path between two nodes without the consideration of anonymous communications. We analyze the traceable rate, which indicates how many segments of a routing path are disclosed to adversaries. Our approach to estimating the traceable rate is unique in that it reduces the problem to merely computing the run length of the bit string representing a routing path. In addition, we introduce an entropy-based metric, called *path anonymity*, to measure the state of not being identifiable, and then formulate path anonymity for onion-based anonymous routing for DTNs. To validate our analysis, we evaluate and compare the numerical and simulation results with randomly generated contact graphs, which demonstrate that our models provide close approximations and/or the same trend as the simulation results. Moreover, we conduct simulations using CRAWDAD dataset cambridge/haggle [27], which is one of the well-known contact traces among mobile nodes. The comparisons indicate that our analyses present similar trends as the simulations resulting from the real trace when the contact graph is dense and enough contact events are provided.

The rest of this paper is organized as follows. The background knowledge for this paper is provided in Section II, and an abstract anonymous routing protocol is presented in Section III. In Section IV, we build performance and security models of anonymous routing in DTNs. Numerical and simulation results under various conditions are compared in Section V. Section VI reviews the existing works for DTN routing and anonymous communications. Section VII concludes this paper.

II. Preliminary

A. Onion Routing

In onion routing [22], the connection between source and destination nodes remains anonymous by connecting the end hosts via a set of relay nodes, called *onion routers*. Each onion router is also not identifiable to any node except to the previous and next onion routers. To achieve this, a layered encryption is applied to a message as shown in Figure 1, where message m is encrypted using an encryption function $E(.)$ with the public keys of onion routers, denoted by r_1, r_2, and r_3, respectively. Only the corresponding onion router can peel off an encrypted layer.

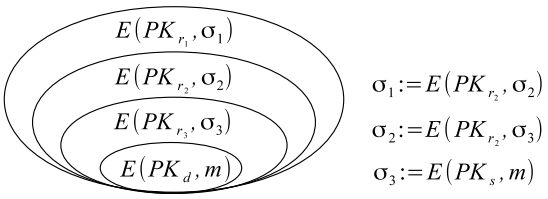

Fig. 1: An example of message encryption in onion routing.

Onion routing works as follows. Assume that source node v_s wishes to send a message via onion routers to destination node v_d. The onion in Figure 2 indicates the routing path. The first layer of the onion is encrypted using the public key PK_{r_1}. Only r_1 can peel it off by the private key corresponding to PK_{r_1} and can identify the next onion router, i.e., r_2, which is shown in Figure 2. Similarly, r_2 and r_3 must decrypt the corresponding layer before v_d to obtain message m. By doing this, where the message originally comes from and where it goes to remain unknown to intermediate nodes at each hop.

Onion routing is commonly used for anonymous communications in ad hoc networks [14]–[21].

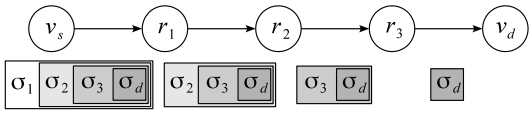

Fig. 2: An example of onion routing.

B. Group Onion Routing

Applying onion routing to DTNs will significantly reduce the performance due to opportunistic contacts among nodes. To accommodate for this, the concept of onion groups has been proposed in [23]–[25], where a set of nodes forms an onion group and any node in the same group can encrypt/decrypt the corresponding layer of an onion. Figure 3 illustrates routing with onion groups, where v_s is the source of a message, v_d is the destination, and $r_{i,j}$ is the j-th node in onion group R_i. The routing process works as follows. Node v_s can forward a message to any $r_{1,j}$ in R_1 upon a contact, and a node in R_{i-1} can forward a message to any node in R_i. Finally, the message reaches v_d. The complete protocol description, such as how to initialize onion groups and keys, and how to improve the anonymity at the last hop, can be found in [25].

C. Traceable Rate

The traceable rate [17] indicates the percentage of path segments disclosed to adversaries when some nodes are compromised. Let η be the number of hops (or message forwarding) between two nodes, C_{seg} be the number of compromised segments in a path, and $c_{seg,i}$ be the hop count of the i-th compromised segment. Then, the traceable rate, denoted by P_{trace}, is defined by Equation 1.

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

Equation 1 implies that the longer the consecutive compromised segments, the higher the traceable rate. For example, let $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$ be a path, where the number of hops is four, i.e., $\eta = 4$. Note that if a node, say v_i, is compromised, the link between v_i and v_{i+1} is disclosed to an adversary. For example, when three nodes, v_1, v_2, and v_4, are compromised, the traceable rate will be $\frac{2^2+1^2}{4^2} = \frac{5}{16}$. If three consecutive nodes, v_2, v_3, and v_4, are compromised, the traceable rate will be $\frac{3^2}{4^2} = \frac{9}{16}$.

D. Anonymity

Anonymity [28] is the state of not being identifiable within an anonymous set; an anonymous set is a set of all the possible entities. For instance, a bit string, say 01XX1, where X could be either 0 or 1, is known to an adversary. The adversary can guess the original bit string within an anonymous set, $\{01001, 01011, 01101, 01111\}$. While the degree of anonymity can be modeled by an entropy-based analysis, the concrete definition is application-dependent [29]. Therefore, we will formulate the anonymity for anonymous routing paths in DTNs in Section IV-E.

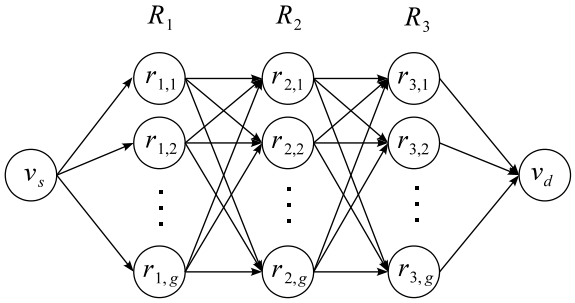

Fig. 3: The overview of onion-based anonymous routing.

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	A message
T	The message deadline
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
$r_{i,j}$	The j-th node in R_i
$1/\lambda_k$	The inter-contact time between nodes at the k-th hop
g	The number of nodes in an onion group
c	The number of compromised nodes
c_o	The number of compromised nodes on a path

III. Abstract Onion-Based Anonymous Routing Protocol

A. Network Model and Definitions

A DTN is represented by a contact graph with n nodes. Each pair of nodes, say v_i and v_j, is connected in the graph if they have at least one contact. Node v_i can forward a message m to v_j at a contact. The link duration at every contact is assumed to be long enough to transmit a complete message.

The inter-contact time between v_i and v_j is defined by $1/\lambda_{i,j}$. The probability that node v_i has a contact with node v_j (hence after we refer to it as the *contact probability*) at time t follows the exponential distribution, i.e., $\lambda_{i,j}e^{-\lambda_{i,j}t}$. Thus, the contact probability of v_i and v_j within T is defined by Equation 3.

$$P[v_i \text{ contacts with } v_j \text{ in } T] = \int_0^T \lambda_{i,j}e^{-\lambda_{i,j}t}dt \quad (2)$$

$$= 1 - e^{-\lambda_{i,j}T} \quad (3)$$

To initialize onion groups, the nodes in a network are divided into n/g groups, where g is the group size. Any node in the same onion group can encrypt/decrypt the corresponding layer of an onion by sharing secret or public/private keys. The work [25] is used for the onion groups and public/private key initialization.

Integer values, K, L, and T, are system parameters, where K is the number of onion routers that a message travels through, L is the number of messages a source node can duplicate, and T is the message deadline. The i-th group of the selected onion groups is denoted by R_i. Hence, a message travels along R_1, R_2, ... R_K before reaching its destination.

Algorithm 1 Single-Copy(v_s, v_d, m, K, T)

1: /* v_s does the following */
2: v_s selects K onion groups.
3: v_s generates an onion.
4: /* v_i does the following at a contact with v_j for the k-th hop. */
5: v_i and v_j establish a secure link.
6: v_i sends m to v_j if v_j is in R_k
7: v_i deletes m from its buffer.
8: increments k by 1.
9: /* v_d does the following */
10: when v_d receives m, returns $SUCCESS$.
11: /* Error handling */
12: **if** m is not delivered in T **then**
13: v_i discards m, and returns $FAIL$.

Depending on the value of L, the message forwarding process and resource requirements are different, and thus, we have two versions of the abstract protocol, in the case of $L = 1$ and $L \geq 2$. In this paper, the abstract protocols with $L = 1$ and $L \geq 2$ are termed *single-copy forwarding* and *multi-copy forwarding*, respectively. Note that single-copy forwarding can be considered as a special case of multi-copy forwarding. However, we explicitly distinguish these two versions, since the protocol with $L \geq 2$ requires more resources, i.e., more variables in its implementation, than the protocol with $L = 1$.

The notations and their definition used in this paper are listed in Table I.

B. Single-Copy Forwarding

Single-copy forwarding is the baseline of the proposed abstract protocol. It works as follows. Let v_s be the node which wishes to deliver message m to v_d. Given system parameters, K, $L = 1$, and T, v_s selects K onion groups, say R_1, R_2, ..., and R_K, and then creates an onion for routing information.

Node v_i (or source node v_s) establishes a secure link with v_j at a contact and checks if v_j is a member of R_k for the k-th hop. If so, v_i forwards m to v_j and deletes m from its buffer. This process continues until m is delivered to v_d. Every message must be delivered to its destination within T. If node v_i holding m detects that the deadline of m is past, m is discarded during a forwarding process.

The pseudo code of single-copy forwarding is provided in Algorithm 1.

C. Multi-Copy Forwarding

While the single-copy forwarding scheme is cost effective, its performance is limited in terms of delivery rate and delay due to the opportunistic nature of DTNs. To improve the performance, a natural approach is to allow multiple copies of a message. However, the multi-copy forwarding scheme not only introduces transmission cost, but also might affect the security measures. Therefore, there is the trade-off between

Algorithm 2 Multi-Copy(v_s, v_d, m, K, L, T)

/* v_s does the following */
v_s selects K onion relays.
v_s generates an onion.
v_s sets $v_s.ticket$ to be L.
/* v_i does the following at a contact with v_j for the k-th hop. */
v_i and v_j establish a secure connection.
if v_j is in R_k and $Forward(v_i, v_j, L, K, T)$ returns true **then**
    v_i sends m to v_j if v_j is in R_k.
    v_i decrements $v_i.ticket$ by 1.
    if $L = 0$ **then**
        v_i deletes m from its buffer.
    v_j sets $v_j.ticket = 1$.
/* v_d does the following */
when v_d receives m, returns $SUCCESS$.
/* Error handling */
if m is not delivered in T **then**
    v_i discards m, and returns $FAIL$.

performance and cost/privacy. This is the motivation to model the onion routing with multi-copy forwarding.

In multi-copy forwarding, up to L copies of a message are allowed in the network. The number of copies that a node can forward is maintained by *tickets*, and thus, an additional variable, $v_i.ticket$, is introduced. In addition, we define a new function, denoted by $Forward(v_i, v_j, m, L, K, T)$, that returns true if v_i determines it forwards m to v_j at a contact, and is false otherwise. The implementation of $Forward(.)$ is left to protocol designers. In our simplified model, $Forward(v_i, v_j, L, K, T)$ returns true when v_j does not have m, and is false otherwise.

The multi-copy forwarding scheme works as follows. Given system parameters, K, L, and T, source node v_s wishes to deliver m to v_d via R_1, R_2, ..., R_K, which are randomly selected. At every contact with v_j, v_i checks if v_j is in R_k for the k-th hop and runs function $Forward(.)$. If the forwarding decision is made, v_i forwards m to v_j and decrements $v_i.ticket$ by 1. When v_i consumes all its tickets, i.e., $v_i.ticket$ becomes 0, m is discarded from v_s's buffer. On the other hand, v_j sets $v_j.ticket$ to be 1 upon receiving m from v_i. This process is repeated until m reaches v_d. During the forwarding process, m is discarded if the delay exceeds the message deadline, T.

The pseudo code of the multi-copy forwarding scheme is shown in Algorithm 2.

IV. Performance and Security Analyses

A. Delivery Rate Analysis for Single-Copy Forwarding

We assume that all onion groups are of the same size, i.e., $g = |R_i|$ for $1 \leq i \leq n/g$. Note that there may exist a group with a smaller size if n is not divisible by g. This factor is ignored in our analyses. For convenience, we refer to the j-th node in R_k, which serves as the k-th onion router, as $r_{k,j}$. The probability that source node v_s contacts any node in the next onion group R_1 is obtained by modifying Equation 3. Since v_s can forward a message to any $r_{1,j}$ in R_1, the inverse of the inter-contact time between v_s and $r_{1,j}$ is simply the summation of $\lambda_{s,r_{1,j}}$ for all $r_{1,j} \in R_1$. Forwarding from a node in the last onion group R_K to destination v_d is similar. The inverse of the inter-contact time between a node in R_{k-1} and any node in R_k can be computed by taking the average of $\lambda_{r_{k-1,i},r_{k,j}}$, where $r_{k-1,i} \in R_{k-1}$ and $r_{k,j} \in R_k$. Thus, λ_k for the k-th hop is obtained by

$$\lambda_k = \begin{cases} \sum_{j=1}^{g} \lambda_{s,r_{k,j}} & \text{for } k = 1 \\ \frac{1}{g} \sum_{i=1}^{g} \sum_{j=1}^{g} \lambda_{r_{k-1,i},r_{k,j}} & \text{for } 2 \leq k \leq K \\ \sum_{j=1}^{g} \lambda_{r_{k-1,j},d} & \text{for } k = K+1. \end{cases} \quad (4)$$

Note that there exist K onion routers between v_s and v_d, and thus the number of hops is $K + 1$. For simplicity, we define $\eta := K + 1$. The contact probability of v_i in contact with any of $r_{k,j} \in R_k$ within T is obtained by $1 - e^{\lambda_k T}$. Since this contact probability is for a single hop, we may take the product of the contact probability of each hop to compute the probability that v_i can deliver a message to v_d within T. According to [26], a routing path in DTNs is called an *opportunistic path*, which is modeled by the hypoexponential distribution. Now, we introduce an *opportunistic onion path* to incorporate the property of group onion routing where a message travels along any node in specified onion groups in the specified order.

Let $A_k^{(\eta)}$ be a coefficient of the hypoexponential distribution as defined by Equation 5.

$$A_k^{(\eta)} = \prod_{j=1, j \neq k}^{(\eta)} \left(\frac{\lambda_j}{\lambda_j - \lambda_k}\right) \quad (5)$$

Consequently, the delivery rate $P_{delivery}(T)$ of a message from v_s to v_d via R_1, R_2, ..., R_K is obtained by Equation 6.

$$P_{delivery}(T) = \sum_{k=1}^{\eta} A_k^{(\eta)} (1 - e^{-\lambda_k T}) \quad (6)$$

Note that our opportunistic anonymous path model differs from [26] in the sense that a node can forward a message to any node in the next onion group. That is, λ_k in Equation 6 is not simply the inverse of the inter-contact time between two nodes.

B. Delivery Rate Analysis for Multi-Copy Forwarding

In the multi-copy forwarding scheme, L copies of a message are allowed in the network. According to [3], the expected delay with L replicas will be the inter-contact time divided by L. Applying this observation, we can deduce the probability that v_i successfully delivers a message to any v_j in R_k for the k-th hop within T as $1 - e^{-\lambda_k LT}$. Therefore, the delivery rate of the L-copy forwarding scheme is given by Equation 7.

$$P_{delivery}(T, L) = \sum_{k=1}^{\eta} A_k^{(\eta)} (1 - e^{-\lambda_k LT}) \quad (7)$$

C. Message Forwarding Cost

We will formulate the message cost with respect to the number of message transmissions between two nodes without the consideration of anonymous communications. In the best case, two nodes are directly connected, i.e., the distance between two nodes is one, if the time duration is infinite. Thus, any DTN routing protocol introduces only $2L-1$ message transmissions, where L is the number of copies of a message, when the delivery delay is not considered. Therefore, the message forwarding cost incurred by anonymous DTN routing is simply the number of message transmissions.

In the single-copy forwarding scheme, the node forwards the message only when it has a contact with a node in the next onion group. Thus, the message transmission cost in terms of the number of forwardings is simply $K+1$, where K is the number of intermediate onion routers.

In the multi-copy forwarding, the source node can forward $L-1$ copies of a message to any node, and one copy to a node in the next onion group. The nodes which receive a copy forward the copy when they have a contact with any node in the next onion group. Hence, the forwarding cost at the first hop is at most $1+2(L-1)$. The forwarding process after the second hop is the same as single-copy forwarding, since the node receiving a message has one copy. Since there are L-copies in the network, the forwarding cost between the second and last hops is at most KL. Therefore, the number of message transmissions is at most $(K+2)L-1$.

D. Traceable Rate Analysis

We analyze the traceable rate of the anonymous onion path when nodes in a network are compromised. In our model, an adversary is assumed to intrude on the node with a message at a contact. Thus, compromising a node causes it to disclose the next node in a routing path. For example, if v_2 is compromised in a path, say $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$, then the link from $v_2 \rightarrow v_3$ is considered traceable.

Let c be the number of compromised nodes during message forwarding from v_s to v_d. For a given c, we can obtain the expected traceable rate of an anonymous path by reducing the problem to compute the expected run length. Note that the run length is the length of the same consecutive 0s or 1s. Let $b=\{b_1,b_2,...,b_\eta\}$ be the binary representation of a path, where $\eta=K+1$. The value of b_i is equal to 0 when the sender of the link is not compromised. Otherwise, b_i equals 1. For instance, if v_2, v_3, and v_5, are compromised on $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6$ then the binary representation of the path will be 01101. Here, the bit strings 11 and 1 have the run length of 2 and 1, respectively. Now, the problem is equivalent to computing the number of the runs of 1s and their length in η bits. By doing this, the geometric distribution can be applied to computing the expected traceable rate.

The probability that a node is compromised is obtained by c/n. Let X_i be the random variable that represents the run length of the first compromised segment starting from b_i. Equation 1 indicates that the weight of the i-th compromised segment is the square of the hop counts, i.e., $(c_{seg,i})^2$, which is equal to the square of the corresponding run length. Thus, we will have the following series for the square of X_i.

$$E[X_1^2] = \sum_{k=1}^{\eta} k^2 \left(\frac{c}{n}\right)^k \left(1-\frac{c}{n}\right) \quad (8)$$

$$E[X_2^2] = \sum_{k=\lceil E[X_1]+1 \rceil}^{\eta} k^2 \left(\frac{c}{n}\right)^k \left(1-\frac{c}{n}\right) \quad (9)$$

$$E[X_i^2] = \sum_{k=\lceil E[X_{i-1}]+1 \rceil}^{\eta} k^2 \left(\frac{c}{n}\right)^k \left(1-\frac{c}{n}\right) + \epsilon \quad (10)$$

Here, ϵ is a negligible value. In addition, $E[X_i]$ is also computed by the geometric distribution, i.e.,

$$E[X_i] = \sum_{k=1}^{\eta-i} k \left(\frac{c}{n}\right)^k \left(1-\frac{c}{n}\right). \quad (11)$$

To estimate the number of compromised segments C_{seg} on a path, we assume c is relatively small compared with n to ensure $E[X_i] \leq 1$. By doing this, we will have $C_{seg} \leq \lceil \eta/2 \rceil$. From Equation 1, we can deduce the expected traceable rate $P_{trace}(c)$ as shown in Equation 12.

$$P_{trace}(c) = \frac{1}{\eta^2} \sum_{i=1}^{\lceil \eta/2 \rceil} E[X_i^2] \quad (12)$$

Both the single-copy and multiple-copy forwarding protocols yield the same traceable rate regardless of the number of L copies, since the the routing paths in the multiple-copy forwarding scheme are considered to be independent from each other. However, an adversary can confine the possible routing path sets once nodes are compromised. That is, the next onion router can be identified within the next onion group, should a relay node be compromised. Such a metric is modeled as path anonymity in the following subsections.

E. Anonymity Analysis for Single-Copy Forwarding

In this section, we will formulate the path anonymity of an anonymous onion path under the single-copy forwarding scheme. Let ϕ be all the possible subjects, and p be the probability of a given subject being the original. The entropy of the system is given by

$$H(\phi) = -\sum_{\forall i \in \phi} p_i log_2(p_i). \quad (13)$$

In our scenario, subjects are routing paths. The system has the maximal entropy, denoted by H_{max}, when no node is compromised. Assuming a routing path is acyclic, for the k-th hop, there are $n-k$ possible next routers, and thus, the number of all possible paths is computed by the permutation of η nodes out of n nodes. Hence, we will have

$$H_{max} = -\sum_{\forall \text{paths in } \phi} \frac{(n-\eta)!}{n!} log_2 \left(\frac{(n-\eta)!}{n!}\right). \quad (14)$$

Let c_{o_k} $(0 \leq c_{o_k} \leq g)$ be the number of compromised nodes in the k-th onion group on a path. The probability of a node being compromised is $\frac{c}{n}$. Since there are g nodes in each onion group, the expected number of compromised nodes in an onion group is $\frac{cg}{n}$. The probability of a compromised node being on a path depends on λ_{i,r_k} for node r_k in R_k for the k-th hop and a sender, v_i. For simplicity, we may approximate the probability of being selected as an onion router (i.e., having the

first contact with sender v_i) as $\frac{1}{g}$. Thus, the joint probability that a node is selected as an onion router and is compromised is equal to $\frac{1}{g} \cdot \frac{cg}{n} = \frac{c}{n}$.

Should a node on a path be compromised, an adversary will be able to confine the next onion router within the next onion group. Let Y ($0 \leq Y \leq \eta$) be the random variable that represents the number of compromised nodes on a path. Given the number of compromised nodes c out of n nodes in a network, $E[Y]$ can be obtained using the Binomial distribution, as follows.

$$E[Y] = \sum_{i=1}^{\eta} i \binom{\eta}{i} \left(\frac{c}{n}\right)^i \left(1 - \frac{c}{n}\right)^{\eta - i} \tag{15}$$

An adversary can guess the next hop with the probability $P_{guess}(v_i, n, g, k)$, where v_i is the k-th node on a path, by

$$P_{guess}(v_i, n, g, k) = \begin{cases} \frac{1}{g} & \text{if } v_i \text{ is compromised} \\ \frac{1}{n-k} & \text{if otherwise.} \end{cases} \tag{16}$$

For simplicity, we refer to $c_o = E[Y]$ as the number of compromised nodes on a path. Thus, we can define the probability of successfully guessing path i's identity in Equation 13 as $p_i = \frac{(n-K+c_o)!}{n!} \frac{1}{g^{c_o}}$. The entropy of the system is obtained by Equation 17.

$$H(\phi') = - \sum_{\forall \text{paths in } \phi'} \frac{(n-\eta+c_o)!}{g^{c_o} n!} log_2 \left(\frac{(n-\eta+c_o)!}{g^{c_o} n!} \right) \tag{17}$$

The path anonymity, denoted by $D(\phi')$ ($0 \leq D(\phi') \leq 1$), can be obtained by $H(\phi')/H_{max}$. Every possible path in an anonymous set has the equal probability of being original, and thus $-\sum_{\forall i \phi} p_i$ in Equation 13 is equal to -1. Hence, only the logarithmic parts of $H(\phi')$ and H_{max} need to be computed, i.e., the part corresponding to $log_2(p_i)$ in Equation 13.

To further simplify $H(\phi')$ and H_{max}, we assume that the number of nodes in a network is large enough with respect to the number of onion groups that a message travels, i.e., $n \gg K$. This assumption is a common case in real networks. For example, a Tor system [12] uses three proxies out of more than 3000 Tor nodes to hide client identity from a server. Thus, we may say $ln(n-K) \simeq ln(n)$. In addition, Stirling's approximation can be applied, i.e., $ln(n!) \simeq nln(n) - n$ for large n. Note that the logarithmic base can be changed by $log_2(n) = ln(n)/ln(2)$.

By applying these approximations to Equations 14 and 17, we have $D(\phi')$ as follows:

$$D(\phi') = \frac{H_(\phi')}{H_{max}} = \frac{log_2 \left(\frac{(n-\eta)!}{n!} \right)}{log_2 \left(\frac{(n-\eta+c_o)!}{g^{c_o} n!} \right)} \tag{18}$$

$$= \frac{(\eta - c_o)(ln(n) - 1) + c_o ln(g)}{\eta (ln(n) - 1)} \tag{19}$$

F. Anonymity Analysis for Multi-Copy Forwarding

In the case of the L-copy forwarding scheme, there could be up to L paths. Hence, the probability that at least one node in an onion group is compromised is equal to $1 - \left(1 - \frac{c}{n}\right)^L$. Let Y' be the random variable that represents the number of onion groups with at least one compromised node on a set of L paths. Similar to Equation 15, the following equation may be obtained.

$$E[Y'] = \sum_{i=1}^{\eta} i \binom{\eta}{i} \left[1 - \left(1 - \frac{c}{n}\right)^L\right]^i \left(1 - \frac{c}{n}\right)^{L(\eta - i)} \tag{20}$$

For simplicity, let $c'_o = E[Y']$, and then, the path anonymity for the L-copy forwarding scheme can be obtained by replacing c_o with c'_o in Equation 19.

V. Simulations

In this section, we will validate our delivery rate, traceable rate, and path anonymity analyses by comparing the numerical and simulation results.

The anonymous routing models that we build are based on the abstract protocols presented in Section III. On the other hand, in the simulation, we have implemented ARDEN [25], which can be seen as a version of the proposed abstract protocol with single-copy forwarding in Algorithm 1. For the multi-copy version, we augment ARDEN with the source spray-and-wait with L-copy forwarding, which can be considered as an extension of Algorithm 2.

With the implemented protocols, our simulations incorporate the consideration of the implementation issues. For example, the last hop forms an onion group to improve the destination anonymity. In addition, some onion groups may have the different group sizes g when the number of nodes n is not divisible by g. Therefore, these factors make the assumptions used by analytical models and simulations different.

A. Simulation Configurations

In these simulations, two kinds of contact graphs, randomly generated contact graph and real traces, are considered, which are elaborated as follows.

Random graphs - A contact graph with 1000 nodes is generated by assigning inter-contact time to each pair of two nodes. The inter-contact time is exponentially distributed with parameter $1/\lambda_{i,j}$ for a pair of nodes v_i and v_j ($i \neq j$), and the initial value of $1/\lambda_{i,j}$ ranges from 0 to 360 minutes. The group size is set to be $1 \leq g \leq 10$ (the default value is 5), the number of onion routers is set to be $1 \leq K \leq 10$ (the default value is 3), the number of copies is set to be $1 \leq L \leq 5$, the message due is set to be $60 \leq T \leq 1800$ minutes, and the percentage of compromised nodes is set to be $0\% \leq c/n \leq 50\%$. The simulation parameters are listed in Table II.

TABLE II: Simulation parameters.

Parameter	value (default value)
The number of nodes	1000
The inter-contact time	0 to 360 minutes
The group size	1 to 10 (3)
The number of onion routers	1 to 10 (3)
The number of copies	1 to 5
The message deadline	60 to 1800 minutes
The % of compromised nodes	0% to 50% (10%)

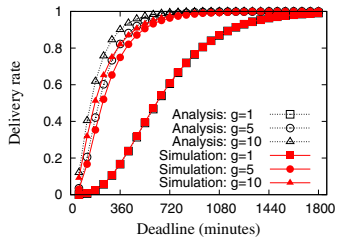

Fig. 4: Delivery rate w.r.t. deadline.

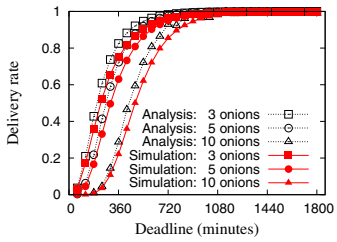

Fig. 5: Delivery rate w.r.t. deadline.

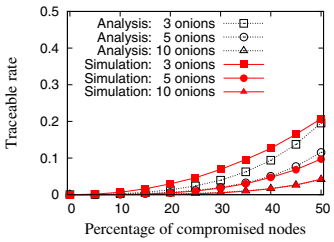

Fig. 6: Traceable rate w.r.t. compromised rate.

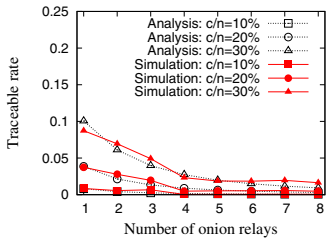

Fig. 7: Traceable rate w.r.t. group size.

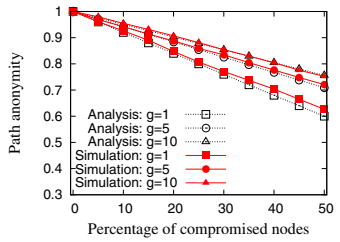

Fig. 8: Path anonymity w.r.t. compromised rate.

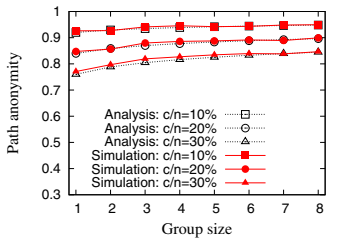

Fig. 9: Path anonymity w.r.t. group size.

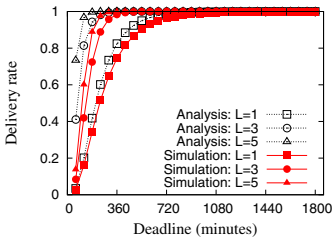

Fig. 10: Delivery rate w.r.t. deadline ($g = 5$.)

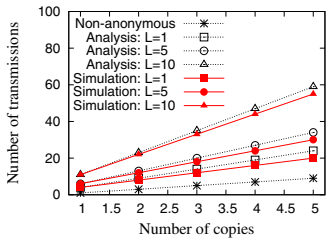

Fig. 11: Message transmission cost w.r.t. the number of copies.

For randomly selected source and destination nodes, each node runs an anonymous onion routing protocol with either single-copy or multi-copy forwarding. If a message is delivered from a source to a destination within the deadline, T, the message delivery succeeds. The numerical results for the delivery rate are computed for each contact graph realization with a given source and destination pair. For security evaluations, nodes are randomly selected as compromised nodes with a given compromised rate, i.e., c/n. The numerical values of the expected traceable rate and path anonymity are computed from the given simulation parameters, (n, g, c, K, L). Unlike the delivery rate, the traceable rate and path anonymity are independent of a contact graph realization, i.,e v_s, v_d, and $1/\lambda_{i,j}$ for all v_i and v_j $(i \neq j)$, and thus, these numerical results are simply computed from configuration parameters.

For each generated contact graph, 1000 simulations are conducted, and the average values for different metrics are compared with numerical results.

Real traces - CRAWDAD dataset cambridge/haggle [27] is a real DTN trace. To be specific, Experience 2 and 3 traces (we refer them as Cambridge and Infocom 2005, respectively) are used in our simulations. In both scenarios, we only consider the contacts between mobile devices, i.e., iMotes, by excluding stationary nodes and external devices. There are 12 and 41 mobile nodes in the Cambridge and Infocom 2005 traces, respectively. Contact events are recorded in the order of seconds. The contact events are traced over several days, and most likely there is no contact in off-business hours. Thus, we assume that a source node initiates a message transmission at any time after it has a contact with any node, which implies that message delivery starts in business-hours but not at night time. By training the traces, the accuracy of the proposed models can be improved.

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

B. Numerical and Simulation Results of Single-Copy Forwarding

Figure 4 shows the delivery rate for different group sizes with respect to the deadline. The results support the intuition that the delivery rate increases as the onion group size increases. This is because having a larger group size brings more forwarding opportunities.

Figure 5 illustrates the delivery rate for different numbers of onion routers, with respect to the deadline. It is clear that a smaller number of onion routers results in a higher delivery rate (or equivalently shorter delay). Although there exists a gap between numerical and simulation results, the same trend can be clearly observed. From Figures 4 and 5, we can say that our delivery rate analysis provides a reasonable approximation.

Figure 6 demonstrates the traceable rate for different numbers of onion routers with respect to the percentage of compromised nodes. As can be seen in the figure, the traceable rate increases in proportion to the percentage of compromised nodes. In addition, larger group sizes lead to smaller traceable rates. This is because the denominator of Equation 1, i.e, the number of hops between a source and destination, becomes relatively larger than the numerator, i.e., the weighted hop counts of compromised segments.

Figure 7 depicts the traceable rate for different percentages of compromised nodes with respect to the number of onion routers. Due to the same reason as Figure 6, adversaries can trace smaller portions of a path as the number of onion routers increases. Figures 6 and 7 indicate that our traceable rate analysis is valid, since the numerical and simulation results are close to each other.

Figure 8 presents the path anonymity for different group sizes with respect to the percentage of compromised nodes.

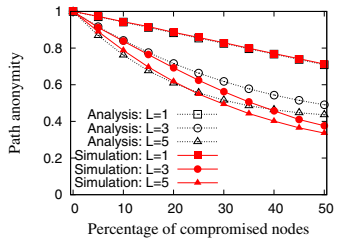

Fig. 12: Path anonymity w.r.t. compromised rate ($g = 5$.)

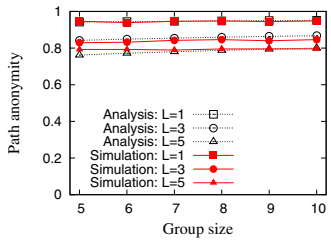

Fig. 13: Path anonymity w.r.t. group size ($c = 10\%$.)

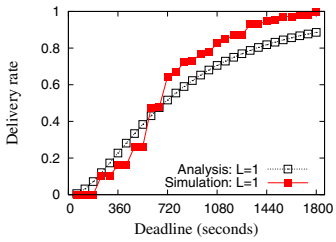

Fig. 14: Delivery rate w.r.t. deadline w/ Cambridge.

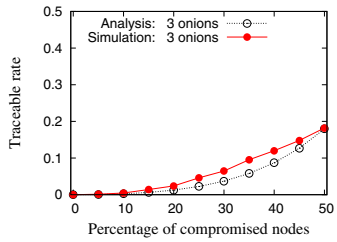

Fig. 15: Traceable rate w.r.t. compromised rate w/ Cambridge.

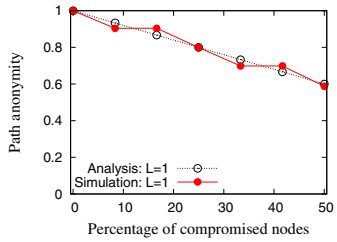

Fig. 16: Path anonymity w.r.t. compromised rate w/ Cambridge.

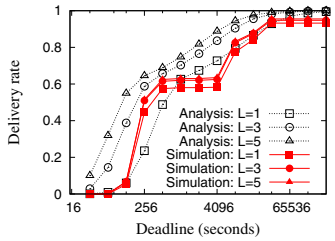

Fig. 17: Delivery rate w.r.t. deadline w/ Infocom.

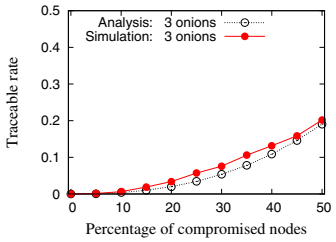

Fig. 18: Traceable rate w.r.t. compromised rate w/ Infocom.

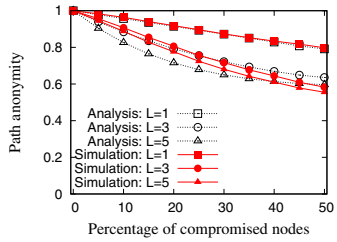

Fig. 19: Path anonymity w.r.t. compromised rate w/ Infocom.

As can be seen in the figure, the larger group size results in higher anonymity, since the possible set of next onion routers increases proportionally when the number of nodes in an onion group increases. This property is observed from Equation 16, i.e., the next onion router is identified with the probability of $1/g$, should the node holding a message be compromised. On the other hand, one may be concerned that a larger group size seems to be insecure, since more nodes share a key to decrypt the corresponding onion layer. However, such a factor is not that crucial compared with the percentage of compromised nodes as shown in Equation 19.

Figure 9 gives the path anonymity for different percentages of compromised nodes with respect to the group size. For the single-copy forwarding scheme, the path anonymity gradually increases as the group size increases. From Figures 8 and 9, we can conclude that our anonymity analysis approximates the simulation results with very high accuracy.

C. Numerical and Simulation Results of Multi-Copy Forwarding

Figure 10 shows the delivery rate for different values of L with respect to the deadline. In this setting, the group size is set to be 5, i.e., $g = 5$, to make sure that $L \leq g$ holds. It is clear that the delivery rate increases as the value of L increases. It is simply that allowing more copies results in more forwarding opportunities. Although there exists a gap between the numerical and simulation results, especially in the case when the deadline is less than 360 minutes, our analysis still displays the same trend as the simulation results.

Figure 11 illuminates the number of message transmissions with respect to the number of message copies L. The upper bound of the message transmission cost is determined by the number of copies L and the number of intermediate onion routers K. As the value of either L or K increases, the number of message transmissions increases. As shown in the figure, the analytical and simulation results are very close to each other. The message transmission cost without the consideration of anonymity results in the smallest message cost. However, we claim that the onion-group-based routing protocols preserve privacy by introducing more message overhead.

Figure 12 illustrates the path anonymity for different values of L with respect to the percentage of compromised nodes. From this figure, we can validate our intuition that the anonymity decreases when L increases. This is because any onion path traverses a node in the specified onion group, and adversaries can correlate the information about paths from compromised nodes. The numerical and simulation results of $L = 3$ and $L = 5$ are very close to each other when the percentage of compromised nodes is less than 30%. However, these lines gradually grow apart from each other when the percentage of compromised nodes increases. The reason for this gap is that our models assume that the number of compromised nodes c is much smaller than the number of nodes n.

Figure 13 demonstrates the path anonymity for different values of L with respect to the group size, where the percentage of compromised nodes is set to be 10%. As can be seen in the figure, the numerical and simulation results are very close to each other. From Figures 10 to 13, we can observe a tradeoff between the delivery rate and anonymity, i.e., the delivery rate increases, but the anonymity decreases as L increases.

D. Results with Cambridge Trace

Figures 14 to 16 are the results with the Cambridge trace (i.e., Experiment 2 in [27]), which is relatively small scale and dense. The number of onion routers, the group size, and the number of copies are set to be $K = 3$, $g = 1$, and $L = 1$, respectively. Since there exist 12 mobile nodes in the Cambridge trace, we consider one set of configuration. Note that making $L \geq 2$ will not help when $g = 1$.

Figure 14 shows the delivery rate with respect to the deadline. Since the Cambridge trace is relatively dense and has enough contact events, a message can be delivered to its destination within relatively small delay. Thus, a message transmission is initiated during business hours, the delivery rate reaches 100% in 1800 seconds (or 30 minutes). With these assumptions, our analysis presents the similar trend as the real trace.

Figure 15 presents the traceable rate with respect to the percentage of compromised nodes. Similar to the case of the randomly generated graphs, the proposed traceable rate analysis provides close approximation even with the real traces. This is because our security model is independent from the inter-contact times among nodes and thus can be applied to any graph.

Figure 16 illustrates the path anonymity with respect to the percentage of compromised nodes. From the figure, the path anonymity decreases linearly as the percentage of compromised nodes increases. The results from simulations with the Cambridge trace and the analysis are very close to each other. Again, this metric is independent from the inter-meeting times among nodes, and therefore, the path anonymity analysis can adapt to a variety of contact traces.

E. Results with Infocom 2005 Trace

Figures 17 to 19 are the results with the Infocom 2005 trace (i.e., Experiment 3 in [27]), which is a medium size contact network. The number of onion routers, the group size, and the number of copies are set to be $K = 3$, $g = 5$, and $L = \{1, 3, 5\}$, respectively.

Figure 17 depicts the delivery rate with respect to the deadline. Note that the x-axis is set to be the logarithmic order, since the contact trace does not have enough contact events for a message to be delivered during business hours in a day. While the delivery rate increases from 16 to 256 seconds, there is no increment between 256 to 4096. This is because there is no contact during this period of time. The delivery rate increases as the deadline becomes longer. Since our delivery rate analysis does not consider the business and off-hours, the analytical results do not capture the simulations results well in the Infocom 2005 trace. However, when $L = 1$, our analysis still presents the similar trend as the simulations except during the off-hours. The multi-copy forwarding scheme of $L = 3$ and $L = 5$ slightly improve the delivery rate compared with the single-copy forwarding scheme $L = 1$, but the difference is not significant. This implies that the diversity in path selection is very small due to the connectivity issue among onion routers. Another possible reason is that the first message is delivered to the destination, but copied messages fail to be delivered due to fewer contact events. In other words, each copy of a message tends to travel the same onion routers.

Figure 18 demonstrates the traceable rate with respect to the percentage of compromised nodes. The traceable rate depends on the number of onion routers and the number of compromised nodes. The difference between the analysis and simulation results are up to only a few percents.

Figure 19 illuminates the path anonymity with respect to the percentage of compromised nodes. When $L = 1$, the numerical and simulations results are perfectly matched. In the case of $L = 3$, our model is very close to the simulation result when the percentage of compromised nodes is less than or equal to 30%. The simulation result with $L = 5$ has slightly lower path anonymity than that with $L = 3$, but not as significant as the case of the randomly generated contact graphs shown in Figure 12. This implies that the paths, via which a set of message copies travel, do not diverse. Hence, the path anonymity slightly decreases from $L = 3$ to $L = 5$, but the delivery rate increases by only a few percents as shown in Figure 17. However, our analysis still shows the same trend as the simulation results.

VI. Related Work

A. Routing in DTNs

To achieve message delivery in a DTN, node mobility is exploited in a routing process, so called *carry-and-forward*. Epidemic routing [1], which is a flooding-like scheme, maximizes the delivery rate since a message is forwarded at every contact. However, this approach introduces a large amount of forwarding overhead. Ticket-based protocols, such as spray-and-wait [2], alleviate this by limiting the number of forwardings based on the number of tickets that a node has for a message. To balance the tradeoff between the delivery rate and forwarding cost, a utility function is introduced to optimize administrator specified metrics [3]. It is known that the use of past contact history significantly improves the delivery rate for a given forwarding cost/message cost [4] and eliminates unnecessary forwarding [5]. In community-based networks, social features among mobile users are exploited for routing [6]. Available knowledge, e.g., contacts, queuing, and traffic demand, differs from application to application, and such factors are classified in [7].

B. Anonymous Communications

Anonymous communications lay in wide areas from mixnet-based systems [12] to Tor [13]. Among them, our interests are in routing-based anonymous systems in wireless networks. As cryptographic-based protocols, a key management scheme to securely update secret keys [14] and an anonymous neighborhood authentication algorithm called MASK [15] to preserve sender and receivers' anonymity at each hop have been proposed. AnonDSR [16] implements the idea of onion routing [22] by collecting symmetric key of intermediate nodes, but a path information is visible to source and destination nodes. ANODR [17] and its variants [18]–[20] generate an onion during route discovery phase by adding an encrypted layer to a request packet. The zone-based protocol [21] first sets up two proxies for source and destination nodes, and then broadcast is used in communications between the proxy and the source/destination node. However, when it comes to DTNs, all the aforementioned anonymous routing protocols ignore the important characteristic of DTNs, i.e., the lack of persistent end-to-end connectivity.

C. Anonymous Communications in DTNs

The works most closely related to this paper are anonymous communications in DTNs. ALAR [23] is an Epidemic-like protocol that hides the source location by dividing a message into

several segments and then sends them to different receivers; meanwhile the sender's identifier is not protected. In onion-based protocols [23]–[25], the idea of onion groups, where a set of nodes share secret keys to allow for any node in the same onion group to encrypt and decrypt the corresponding layer of an onion, is introduced to accommodate the opportunistic nature of DTNs. To establish such groups, attribute-based encryption (ABE) [30] or identity-based cryptography (IBC) [31] is used. In TPS [32], a message must travel for at least τ groups out of s groups, based on the threshold secret sharing [33], and then a pivot forwards the message to its destination. While this threshold scheme alleviates the longer delay due to the use of onions, the final destination of a message is revealed to the pivot. The most viable solutions are ARDEN [25] and EnPassant [24] that forward a message along a set of onion routers in an order of specified onion groups. Nevertheless, no theoretical works related to the performance and security issues in onion-based anonymous routing have been addressed.

VII. Conclusion

In this paper, we first design an abstract onion-based anonymous routing protocol and then extend the existing protocol with group onions into multi-copy forwarding. The main contributions of this paper are performance and security analyses of onion-based anonymous routing for DTNs. The delivery rate is mathematically modeled by incorporating the consideration of anycast-like message forwarding by group onions. The traceable rate of an anonymous routing path is analyzed by reducing the problem to computing the run length of the bit string that represents a routing path. In addition, the path anonymity, which is an application-dependent entropy-based metric, is defined and formulated. Furthermore, we demonstrate that the numerical and simulation results are very close to each other, or share the same trend. Finally, the proposed analyses present close approximation to the simulation results with one of the well-known real traces, CRAWDAD dataset cambridge/haggle, as long as enough contact events are fed and the graph representation of a trace is dense. We believe that our theoretical work supports the fundamental understanding of the performance and security issues related to onion-based anonymous routing for DTNs.

References

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

[2] 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.

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

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

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

[6] Q. Li, G. Cao, and T. F. L. Porta, "Efficient and privacy-aware data aggregation in mobile sensing," *IEEE Trans. Dependable Secur. Comput.*, vol. 11, no. 2, pp. 115–129, 2014.

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

[8] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a Bus-based Disruption-Tolerant Network: Mobility Modeling and Impact on Routing," in *Mobicom*, 2007, pp. 195–206.

[9] M. Motani, V. Srinivasan, and P. S. Nuggehalli, "PeopleNet: Engineering a Wireless Virtual Social Network," in *Mobicom*, 2005, pp. 243–257.

[10] S. Wang, M. Liu, X. Cheng, and M. Song, "Routing in Pocket Switched Networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 67–73, 2012.

[11] J. Ren and J. Wu, "Survey on Anonymous Communications in Computer Networks," *Comput. Netw.*, vol. 33, no. 4, pp. 420–431, 2010.

[12] 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.

[13] S. B. Mokhtar, G. Berthou, A. Diarra, V. Quéma, and A. Shoker, "RAC: A Freerider-Resilient, Scalable, Anonymous Communication Protocol," in *ICDCS*, 2013, pp. 520–529.

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

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

[16] R. Song, L. Korba, and G. Yee, "AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-hoc Networks," in *SASN*, 2005, pp. 33–42.

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

[18] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng, "Anonymous Secure Routing in Mobile Ad-Hoc Networks," in *LCN*, 2004, pp. 102–108.

[19] L. Yang, M. Jakobsson, and S. Wetzel, "Discount Anonymous On Demand Routing for Mobile Ad hoc Networks," in *SecureComm*, 2006, pp. 1–10.

[20] D. Sy, R. Chen, and L. Bao, "ODAR: On-Demand Anonymous Routing in Ad Hoc Networks," in *MASS*, 2006, pp. 267–276.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

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

[29] Y. Guan, X. Fu, R. Bettati, and W. Zhao, "A Quantitative Analysis of Anonymous Communications," *IEEE Trans. on Reliability*, vol. 53, no. 1, pp. 103–115, 2004.

[30] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *SP*, 2007, pp. 321–334.

[31] A. Kate, G. M. Zaverucha, and U. Hengartner, "Anonymity and Security in Delay Tolerant Networks," in *SecureComm*, 2007, pp. 504–513.

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

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