Distributed Privacy-Protecting DTN Routing: Concealing the Information Indispensable in Routing

Kang Chen
Department of Electrical and Computer Engineering
Southern Illinois University, Carbondale, IL 62901
Email: kchen@siu.edu

Haiying Shen
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29631
Email: shenh@clemson.edu

***Abstract*—Nodes in Delay Tolerant Networks (DTN) rely on routing utilities (e.g., probabilities of meeting nodes) to decide the packet forwarder. As the utilities reflect user privacy, nodes may be reluctant to disclose such information directly. Therefore, we propose a distributed strategy to protect the aforementioned private information in utility-based DTN routing algorithms while still guarantying the correctness of packet forwarding, namely meeting Relationship Anonymity (ReHider). We also present an enhanced version that can better prevent certain malicious behaviors (probing attack and brute-force attack). Initial analysis show the effectiveness of the proposed strategy.**

I. Introduction

In most DTN routing algorithms [1]–[8], packet forwarder is selected according to the routing utility when nodes meet. For this purpose, the routing utility usually is deduced from node encountering records and/or social properties, e.g., meeting frequency [1]–[3], social closeness [4]–[8], and network centrality [4], [5]. Such a design rationale means that the routing utility reflects much private information, which can be exploited by adversaries for harmful attacks. Therefore, it is essential to protect the routing utility in DTN routing.

However, concealing such information in DTN routing is non-trivial as it is indispensable for efficient routing. This **paradox** poses a formidable challenge: *how to anonymize the routing utilities in DTN routing while guaranteeing the correctness of packet forwarding?* Therefore, in this paper, we propose a distributed strategy, namely meeting Relationship Anonymity (ReHider), to solve the challenge. ReHider exploits commutative encryption algorithm [9], order-preserving hash function [10], and a set of novel routing utility exchange and packet forwarding sequences to fulfill the design goal.

II. Preliminaries

We use $\mathcal{U}_{ij}$ to denote node n_i's routing utility for n_j:

$$\mathcal{U}_{ij} = \{n_i, n_j, v_{ij}\}, \tag{1}$$

where n_i, n_j, and v_{ij} denote the **source**, **target**, and **value** of $\mathcal{U}_{ij}$, respectively. We refer a node's routing utility for a packet as its routing utility for the packet's destination.

A commutative encryption algorithm $\mathcal{E}(\cdot)$ satisfies the properties below for any keys k_1 and k_2, message M, rational number s and $\epsilon < 1/2^s$

- $\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(M)) = \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(M))$
- $\forall\ M_1 \neq M_2,\ Pr(\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(M_1)) = \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(M_2))) < \epsilon,$

where $\mathcal{E}_{k_1}(M)$ means encrypting message M with key k_1.

An order-preserving hash function $\mathcal{H}(\cdot)$ satisfies properties below for v_1 and v_2 [10]

- If $\mathcal{H}(v_1) = \mathcal{H}(v_2)$, $v_1 = v_2$
- If $v_1 > v_2$, $\mathcal{H}(v_1) > \mathcal{H}(v_2)$.

III. System Design

We use the case when n_1 and n_2 meet for packet routing for illustration. After delivering packets to each other, they compare their routing utilities for the destinations of all remaining packets on them, denoted by $\{n_a, n_b, n_c\}$ $(a, b, c \in [3, N])$. We let x denote an element in set $\{a, b, c\}$, i.e., $x \in \{a, b, c\}$. Such a setting is an example and our strategy can be applied to cases with different numbers of destinations.

A. Baseline Meeting Relationship Anonymity (B-ReHider)

B-ReHider realizes anonymous routing utility comparison between two encountered nodes.

1) Design of B-ReHider:

(a) **Initial Setup:** Each of the two nodes first creates an encryption key, say k_1 and k_2. The two nodes also select a node from them as the *comparison node*, say n_1 is selected. They then compare their routing utilities for $\{n_a, n_b, n_c\}$ to determine the packet forwarder.

(b) **Utility Encryption:** Each node first encrypts the targets of its utilities with its key. Beside, n_2 also hashes the values of its utilities in order to hide this information from n_1. After this, each node sends all encrypted utilities to the other node.

$$n_1 \rightarrow n_2 : \mathcal{U}'_{1x} : \{n_1, \mathcal{E}_{k_1}(n_x), v_{1x}\}$$
$$n_2 \rightarrow n_1 : \mathcal{U}'_{2x} : \{n_2, \mathcal{E}_{k_2}(n_x), \mathcal{H}_2(v_{2x})\}$$

n_1 and n_2 further encrypt the target of all received utilities with their keys. n_2 also hashes the values of received utilities with its hash function. As a result, n_1 has $\mathcal{U}''_{2x} : \{n_2, \mathcal{E}_{k_1}(\mathcal{E}_{k_2}(n_x)), \mathcal{H}_2(v_{2x})\}$, and n_2 has $\mathcal{U}''_{1x} : \{n_1, \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x)), \mathcal{H}_2(v_{1x})\}$. Finally, n_2 sends the encrypted n_1's utilities to the comparison node n_1 for comparison.

$$n_2 \rightarrow n_1 : \mathcal{U}''_{1x} : \{n_1, \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x)), \mathcal{H}_2(v_{1x})\}$$
$$n_1 \text{ has} : \mathcal{U}''_{2x} : \{n_2, \mathcal{E}_{k_1}(\mathcal{E}_{k_2}(n_x)), \mathcal{H}_2(v_{2x})\} \ \ and \ \ \mathcal{U}''_{1x}$$

(c) **Utility Comparison:** n_1 compares $\mathcal{U}''_{2x}$ and $\mathcal{U}''_{1x}$ to decide the packet forwarder for each destination. Due to the commutative encryption, if $\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(n_x)) = \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_y))$, we

can conclude that $n_x = n_y$. This means that routing utilities for the same target in $\mathcal{U}''_{2x}$ and $\mathcal{U}''_{1x}$ have the same encrypted target and can be easily identified for comparison.

***(d)* Decrypting the Comparison Result:** The comparison in the previous step determines which node (n_1 or n_2) is the forwarder for each encrypted destination, e.g., $\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(n_x))$. Then, n_1 first decrypts those destinations with k_1 and sends the result to n_2 for further decryption. As a result, n_2 can know it is the forwarder for which destinations, which is shared to n_1 too. Finally, utility comparison is done anonymously.

2) Privacy Protection Analysis: In this section, we analyze B-ReHider's capability to resist attacks mentioned early.

Anonymize Routing Utilities: We first summarize the information that a node can collect in B-ReHider in Table I to analyze whether routing utilities are anonymized.

TABLE I: Information collected by in each node in B-ReHider.

Node	Information
n_1	$\mathcal{U}'_{1x} : \{\mathcal{E}_{k_1}(n_x), v_{1x}, n_1\}$ $\mathcal{U}''_{1x} : \{\mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x)), \mathcal{H}_2(v_{1x}), n_1\}$ $\mathcal{U}'_{2x} : \{\mathcal{E}_{k_2}(n_x), \mathcal{H}_2(v_{2x}), n_2\}$ $\mathcal{U}''_{2x} : \{\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(n_x)), \mathcal{H}_2(v_{2x}), n_2\}$
n_2	$\mathcal{U}'_{2x} : \{\mathcal{E}_{k_2}(n_x), \mathcal{H}_2(v_{2x}), n_2\}$ $\mathcal{U}'_{1x} : \{\mathcal{E}_{k_1}(n_x), v_{1x}, n_1\}$ $\mathcal{U}''_{1x} : \{\mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x)), \mathcal{H}_2(v_{1x}), n_1\}$

We see from the table that each node can only get the utilities with encrypted targets and/or hashed values. This means each node's routing utilities are anonymized against the other node during the packet routing in B-ReHider.

Eavesdropping: By examining the utilities transmitted in B-ReHider, we find that they cannot be understood by any eavesdropper because for each transmitted utility, its target is encrypted, i.e., $\mathcal{E}_{k_1}(n_x)$ or $\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(n_x))$, and its utility values are hashed. Therefore, eavesdroppers cannot obtain any meaningful information without knowing k_1, k_2, and the hash functions ($H_1()$ and $H_2()$).

Probing Attack and Brute-Force Attack: B-ReHider cannot resist the probing attack and the brute-force attack. First, since the utility comparison result is shared between the two nodes in B-ReHier, a malicious node can easily probe another node's routing utilities by repetitively conducting packet routing (i.e., comparing routing utilities) with it. After each packet routing, the malicious node can adjust its routing utility values based on the comparison result. Then, after several rounds, the node's routing utility values can be gradually deduced.

Second, by examining Table I, we find that n_1 can easily access multiple clear-text and cipher-text pairs of $\mathcal{E}_{k_2}()$ and $\mathcal{H}_2()$. In detail, n_1 can sort $\mathcal{U}'_{1x} : \{n_1, \mathcal{E}_{k_1}(n_x), v_{1x}\}$ by v_{1x} and $\mathcal{U}''_{1x} : \{n_1, \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x)), \mathcal{H}_2(v_{1x})\}$ by $\mathcal{H}_2(v_{1x})$. Since $\mathcal{H}_2()$ is order-reserving, $\mathcal{H}_2(v_{1x})$ has the same order as v_{1x}. As a result, $\mathcal{E}_{k_1}(n_x)$ and $\mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x))$ appear on the same position in each sorted set. This means that n_1 can get multiple clear-text and cipher-text pairs: $< \mathcal{E}_{k_1}(n_x), \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x)) >$ and $< v_{1x}, \mathcal{H}_2(v_{1x}) >$ to break $\mathcal{E}_{k_2}(\cdot)$ and $\mathcal{H}_2(\cdot)$.

B. Enhanced Relationship Anonymity (E-ReHider)

We further propose an enhanced version, named E-ReHider, to prevent the two attacks suffered by B-ReHider.

1) Preventing the Probing Attack: To prevent this attack, we let nodes 1) use a pseudonym to communication with the encountered node for packet routing and 2) change the pseudonym after conducting the packet routing. This means that a node presents a different pseudonym each time when it meets a node. Consequently, the malicious node cannot identify the same node for the probing attack.

2) Preventing the Brute-force Attack: As introduced in Section III-A2, B-ReHider suffers from the brute-force attack mainly because v_{1x} and $\mathcal{H}_2(v_{1x})$ on n_2 have the same order. We then solve the problem by breaking such a property. The general idea is to create zombie destinations, say n_z, which do not exist in packets on both nodes, and let n_2 modify utilities for those destinations received from n_1, i.e., n_z to $\widetilde{v_{1z}}$. Then, $\{v_{1x}, v_{1z}\}$ and $\{\mathcal{H}_2(v_{1x}), \mathcal{H}_2(\widetilde{v_{1z}})\}$ do not have the same order since $\mathcal{H}_2(\widetilde{v_{1z}})$ is different from $\mathcal{H}_2(v_{1z})$. Consequently, n_1 cannot correlate $\mathcal{E}_{k_1}(n_x)$ with $\mathcal{E}_{k_2}(\mathcal{E}_{k_1}(n_x))$ or v_{1z} with $\mathcal{H}_2(v_{1z})$, i.e., cannot easily collect pairs of clear-text and cipher-text for the brute-force attack.

IV. Conclusion

In this paper, we propose ReHider to protect the routing utilities in utility-based DTN routing algorithms. ReHider uses commutative encryption and order-preserving hashing to realize the designn goal. We also propose an enhanced version that can better thwart malicious attacks. Analytical results show that the proposed strategy can effectively protect the private information without sacrificing routing efficiency. In the future, we plan to investigate the protection of private information under more complicated attacks.

Acknowledgement

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, IBM Faculty Award 5501145 and Microsoft Research Faculty Fellowship 8300751.

References

[1] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks." *Mobile Computing and Communications Review*, vol. 7, no. 3, 2003.

[2] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *Proc. of INFOCOM*, 2006.

[3] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem." in *Proc. of SIGCOMM*, 2007.

[4] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *Proc. of MobiHoc*, 2008.

[5] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. of MobiHoc*, 2007.

[6] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multi-copy routing in mobile social networks." in *Proc. of INFOCOM*, 2013.

[7] W. Gao and G. Cao, "On exploiting transient contact patterns for data forwarding in delay tolerant networks." in *Proc. of ICNP*, 2010.

[8] X. Zhang and G. Cao, "Transient community detection and its application to data forwarding in delay tolerant networks." in *Proc. of ICNP*, 2013.

[9] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data." *IEEE TKDE*, vol. 16, no. 9, pp. 1026–1037, 2004.

[10] E. A. Fox, Q. F. Chen, A. M. Daoud, and L. S. Heath, "Order preserving minimal perfect hash functions and information retrieval." in *Proc. of SIGIR*, 1990.