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

Kang Chen
Department of ECE
Southern Illinois University
kchen@siu.edu

Haiying Shen
Department of Computer Science
University of Virginia
hs6ms@virginia.edu

Background

What is Delay Tolerant Networks (DTNs)

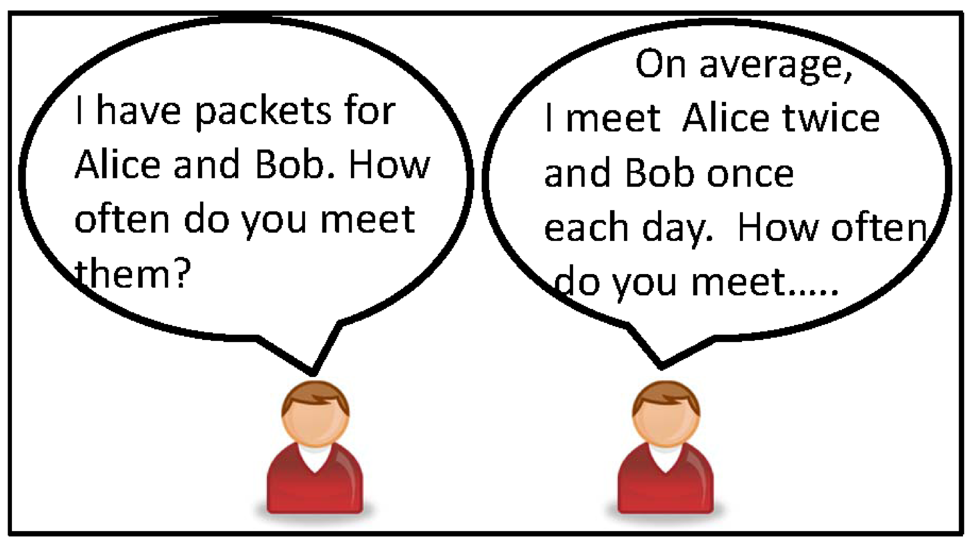
1. A type of mobile ad hoc networks with sparsely distributed nodes.
2. A challenging network environment that lacks end-to-end path;.

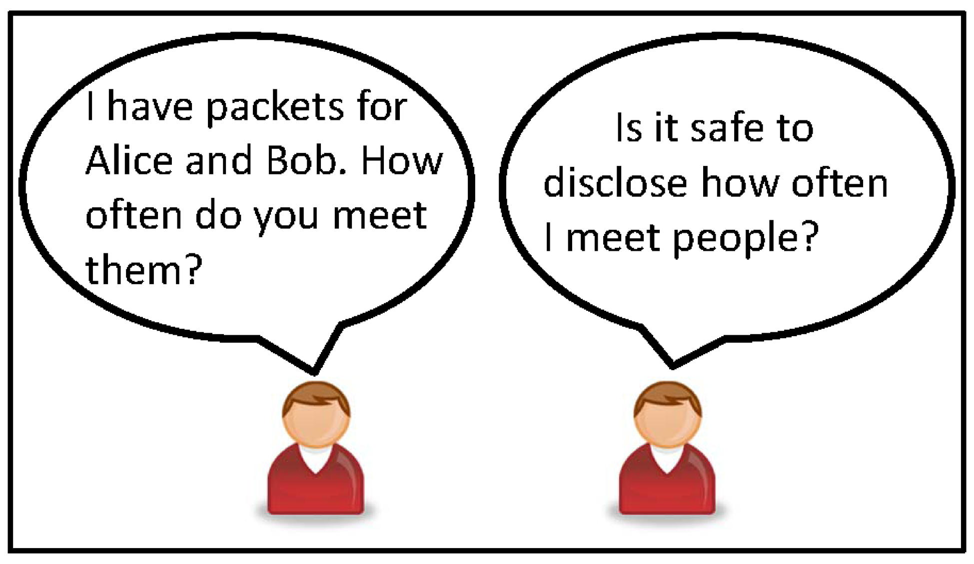
Routing in DTNs

1. Rely on opportunistic encountering between nodes for packet forwarding.
2. Often in a store-carry-forward manner: a packet is carried on current holder until a better node is encountered .
3. The key is to determine "what makes a better forwarder"?
4. We can define a routing utility for forwarder selection: always forward a packet to the node with a higher routing utility.

Problem Formation

1. Current routing methods deduce routing utility based on:
   - Past encountering frequencies
   - Social closeness
   - Network centrality
2. This indicates that routing utilities reflect the privacy of mobile nodes and their owners (if applicable).
3. When nodes meet for routing, they have to exchange routing utilities to determine packet forwarder, which create security concerns.





Design Goal

Anonymize the routing utilities in DTN routing for safety while guaranteeing the correctness of packet forwarding

System Design

Preliminary:

- Routing utility $U_{ij} = \{n_i, n_j, v_{ij}\}$
  - Denotes the source, target, and value of U_{ij}, respectively. We refer a node's routing utility for a packet as its routing utility for the packet's destination.
- Commutative encryption $\epsilon(\,)$
  - Encrypt the same content with two keys in different orders will generate the same result

$$\varepsilon_{k_1}(\varepsilon_{k_2}(M)) = \varepsilon_{k_2}(\varepsilon_{k_1}(M))$$

- Order-preserving hashing H( )
  - Hashing that preservers the order
  - If $\mathrm{H}(v_1) = \mathrm{H}(v_2)$, then $v_1 = v_2$
  - If $v_1 > v_2$, then $\mathrm{H}(v_1) > \mathrm{H}(v_2)$

Baseline Meeting Relationship Anonymity (B-ReHider)

General idea: anonymize the routing utilities before the exchange with the encountered nodes and make sure that the comparison of routing utilities can be conducted correctly.

We show this process when n_1 meets n_2 for packet routing:

1. **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.
2. **Utility Encryption**: Each node 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. They then repeat this process on received utilities. This means (next column):

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

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

3. **Utility Comparison**: n_1 compares U''_{1x} and U''_{2x} to decide the packet forwarder for each destination. Due to the commutative encryption, if $\varepsilon_{k_1}(\varepsilon_{k_2}(n_x)) = \varepsilon_{k_2}(\varepsilon_{k_1}(n_y))$, we can conclude that $n_x = n_y$.
4. **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., $\varepsilon_{k_1}(\varepsilon_{k_2}(n_x))$.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 and correctly.

The information a node can collect in B-ReHider shows the protection on privacy

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

Enhanced Meeting Relationship Anonymity (E-ReHider)

1. Prevent probing: Let nodes change the pseudonym used to communicate from time to time.
2. Prevent brute-force attack: create zombie destinations that do not exist in packets on both nodes.

Future Work

We will further study how to protect packet forwarder information

Acknowledgments

- U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, IBM Faculty Award 5501145, and Microsoft Research Faculty Fellowship 8300751.

*The majority of the work was done when at Clemson University