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

Kang Chen¹ and Haiying Shen²

¹Dept. of ECE, Southern Illinois University, IL, USA
²Dept. of CS, University of Virginia, VA, USA

* Majority was done when at Clemson
Outline

• Introduction
• System Design
• Performance Evaluation
• Conclusion
Introduction

• Delay/Disruption Tolerant Networks (DTNs)
  • A challenging form of mobile network
  • Nodes are sparsely distributed
  • Opportunistic node encountering
  • No infrastructure, only Peer-to-Peer communication

• Network Features
  • Limited resources
  • Frequent network partition and disconnection
  • End-to-end path cannot be ensured
Introduction

• Routing is possible
  • Often in a store-carry-forward manner
  
  ![Routing Example](image)

  s → r → d

  • Utility based routing principle
    • Define a utility that represents how likely to meet a node (directly) or deliver a packet to a node (indirectly)
    • When two nodes meet, they exchange and compare routing utilities for each destination, and always forward a packet to the node with a higher utility value

• Common utility definitions
  • Meeting frequency; social closeness; network centrality, etc.
Introduction

• Privacy concerns
  • Those routing utilities contain much private information
    • Meeting frequency, social relationship, locations, etc.
    • More severe in DTNs involving human-operated devices
      • Pocket switched network, Vehicular DTNs, etc.
  
• Malicious nodes could take advantage of them
  • Fabricate routing utilities to attract and drop packets
  • Disseminate virus to specific targets or locations
Introduction

• Challenges
  • On one side, disclosing routing utilities is not privacy preserving
  • On the other side, DTN routing requires nodes to exchange such information

• Goal
  • Harmonizing both needs
  • Anonymizing such information by
    • Carefully disclosing partial routing utility information that is enough for correct routing
    • Altering the packet forwarding sequences
Outline

• Introduction
• System Design
• Performance Evaluation
• Conclusion
System Design : Utility Anonymity

• Some definitions
  • Routing utility: \( U_{ij} = \{n_i, n_j, v_{ij}\} \),
    • \( v_{ij} \) denotes \( n_i \)'s utility value for \( n_j \)
  
  • Commutative encryption: \( E(\cdot) \)
    • \( E_{k_1} (E_{k_2} (M)) = E_{k_2} (E_{k_1} (K)) \) for encryption key \( k_1 \) and \( k_2 \)
  
  • Order-preserving hashing: \( H(\cdot) \)
    • If \( v_1 > v_2 \), \( H(v_1) > H(v_2) \)
System Design: Utility Anonymity

• Observations
  • \( U_{ij} = \{n_i, n_j, v_{ij}\} \) is anonymized when any of the three elements is anonymized (assume enough nodes in the network)
  • To ensure correct routing, two nodes just need to know the order of their utility values for the same destination

• Solution
  • Nodes exchange partially encrypted/hashed routing utility
  • Nodes could identify and compare routing utility for the same destination node
  • But at least one of three element is not disclosed to the other node
System Design: Utility Anonymity

- Illustration scenario
  - \( n_1 \) meets \( n_2 \) for packet forwarding
  - \( n_1 \) is selected as the node that will do utility comparison
  - \( n_1 \) pick key \( k_1 \) and hashing function \( H_1 \), \( n_2 \) pick key \( k_2 \) and hashing function \( H_2 \)
- Step 1
  \[
  n_1 \rightarrow n_2 : U'_{1x} = (n_1, E_{k_1}(n_x), v_{1x})
  \]
  \( n_2 \) generates \( U''_{1x} = (n_1, E_{k_2}(E_{k_1}(n_x)), H_2(v_{1x})) \)
  \[
  n_2 \rightarrow n_1 : U''_{1x}
  \]
  \[
  n_2 \rightarrow n_1 : U'_2x = (n_2, E_{k_2}(n_x), H_2(v_{2x}))
  \]
  \( n_1 \) generates \( U''_{2x} = (n_2, E_{k_1}(E_{k_2}(n_x)), H_2(v_{2x})) \)
System Design: Utility Anonymity

- **Step 2**
  \[ n_1 \text{now has} \]
  \[ U''_{1x} = (n_1, E_{k_2}(E_{k_1}(n_x)), H_2(v_{1x})) \]
  \[ U''_{2x} = (n_2, E_{k_1}(E_{k_2}(n_x)), H_2(v_{2x})) \]
  Due to commutative encryption, routing utilities with the same \( n_x \) could be identified
  Due to order-preserving hashing, their utility values \( (H_2(v_{1x}) \) and \( H_2(v_{2x}) \)) could be compared

- **Step 3**
  \[ n_1 \text{informs } n_2 \text{ those destinations that it has a higher utility value} \]
  \[ n_1 \rightarrow n_2 : E_{k_2}(n_x) \text{ if } H_2(v_{1x}) > H_2(v_{2x}) \]
  \[ n_2 \text{ decrypts and knows that } n_1 \text{ is the forwarder for which dest. and informs } n_1 \]
  \[ \text{It further knows itself is the forwarder for which dest.} \]
System Design : Utility Anonymity

- Summary

  - Anonymity is attained:
    - Each node can only get the utilities with at least one element encryptedhashed
  
  - Routing is ensured:
    - Routing utilities are successfully compared
System Design : Forwarder Anonymity

• **Forwarder**
  • The node that holds the packet (i.e., the node with the highest utility for the destination of the packet)
  
  • Such information is private too
    • Targeting a specific destination by tracking packets destined to the destination

• $n_2$ has the highest utility value for $n_{10}$ among all neighbors.
• It is the forwarder for packets destined to $n_{10}$
System Design : Forwarder Anonymity

• How to protect such forwarder information?
  • Forwarder information contains two parts: <dest., forwarder>

• Hide one by changing the process of routing utility comparison and packet forwarding
  • Choose a relay node among the group of encountered nodes
  • The relay node knows the forwarder for each encrypted destination

• Only applies when a group of nodes meet
  • No way to hide when only two nodes meet
System Design : Forwarder Anonymity

• Illustration scenario
  • \( n_1, n_2, n_3, n_4 \) meet for packet forwarding
  • \( n_2 \) is selected as the relay node, the remaining form the Neighbor set
  • \( n_1 \) is the head of the neighbor set and decides a group key \( k_n \)

• Step 1
  • Each node in the neighbor set encrypts its routing utility with \( k_n \) and send to \( n_2 \)
System Design: Forwarder Anonymity

- Step 2

\( n_1 \) and \( n_2 \) compare routing utilities from the neighbor set and those on \( n_2 \) following the method for Utility Anonymity.

- Step 3

\( n_2 \) builds a relay table as the following:

<table>
<thead>
<tr>
<th>( k_n )-encrypted destination</th>
<th>Forwarder</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{E}_{k_n}(n_a) )</td>
<td>( n_1 )</td>
</tr>
<tr>
<td>( \mathcal{E}_{k_n}(n_c) )</td>
<td>( n_3 )</td>
</tr>
<tr>
<td>( \mathcal{E}_{k_n}(n_d) )</td>
<td>( n_4 )</td>
</tr>
</tbody>
</table>
System Design: Forwarder Anonymity

• Step 4

\[ n_1, n_3, \text{ and } n_4 \text{ encrypt its packets’ destination with } k_n \text{ and send to } n_2 \text{ for relay} \]

\[ n_2 \text{ searches the relay table and forward the packet if there is a hit, or keep the packet if not (itself is the forwarder)} \]

• Summary

  • \( n_2 \) only knows the forwarder for each \( k_n \)-encrypted destination, so it cannot know the complete forwarder information
  • Others only know that packets are relayed by \( n_2 \)
Outline

- Introduction
- System Design
- Performance Evaluation
- Conclusion
Evaluation

• Traces
  • Haggle: encountering of mobile devices in a conference
  • MIT Reality: encountering of mobile devices on a campus

• Methods
  • Privacy protection is analyzed in the paper
  • Measuring the routing performance with the proposed methods
    • Using PROPHET* as the baseline routing algorithm
    • PROPHET-G denotes extended pair-wise encountering assumption

Evaluation : Routing Performance

• MIT Reality trace

- B-ReHider and E-ReHider indicate utility anonymity and its extended version
- B-FwHider and E-FwHider indicate forwarder anonymity and its extended version
- Routing efficiency is not affected with the privacy protection schemes
Evaluation: Routing Performance

- Haggle trace

- The same result as in the MIT Reality trace
Conclusion

- Routing utilities in DTNs contain much privacy information but need to be disclosed for correct routing
- Solution:
  - Careful encryption to let nodes only share partial utility information that is enough for correct routing
  - Altering the packet forwarding sequences to further anonymity forwarder information
- Future work:
  - Energy consumption
  - Loose the limit and allow a white-list
Thank you!
Questions & Comments?