Anonymous DTN routing

October 5, 2013

1 Attack scenario

In our attack scenario, an adversary tries to track a victim node or infer trusted nodes of a victim node by exploiting 1) ephemeral ID of victim node and 2) packets stored in victim node. Since ephemeral ID of victim node keeps changing at reasonably short interval, leakage of ephemeral ID is not an issue in our protocol. From packets stored in victim node, an adversary is able to track or infer trusted nodes of the victim node through 1) ephemeral ID of packet destination, 2) packet ID or 3) packet payload.

1.1 Tracking a victim node

An adversary may track a victim node by injecting a "marker packet" to the victim node. Marker packet can be identified by its destination ephemeral ID, packet ID or payload.

Upon encountering a victim node V, an adversary A sends a marker packet p with non-existing or inactive destination. If V only can relay packets pulled by other encountered nodes, the marker packet p would not be relayed to other nodes. Therefore, A can track the victim node V until V drops the marker packet p by testing if a node has p.

Remedy (Applied to simulation code) Our protocol allows V relays packets that are not pulled by the encountered node when V relays all the pulled packets. (Or V can relay unpulled packets even before all pulled packets are relayed.) Therefore, A may be able to track V in the short term but in the long term, A cannot assure that a node with p is actually V or not.

1.2 Inferring victim's trusted nodes

An adversary A may infer trusted nodes of a victim node V by knowing which packets are pulled by V.

1.2.1 Attack 1: Destination ephemeral ID

Let's say V's trusted node are $\{a, b, c, d\}$. V has packets destined for $\{b, c\}$ and A prepares marker packets destined for $\{d, e\}$. When V and A encounter, A receives the packet digest of V, $\{b, c\}$. By the protocol, V pulls packet destined for $\{c\}$ through secure method such as PIR. Now A changes its ephemeral ID to A' and gets the packet digest of V. Since the packet digest of V is now $\{b, c, d\}$, A can infer that d is trusted by V.

Remedy (Applied to simulation code) Since V is able to change ephemeral ID of packets destined for its trusted nodes at the end of the current epoch, this attack is feasible only during the current epoch. If V does not include the packets received from untrusted nodes during the current epoch in its packet digest, the attack is not possible any more.

1.2.2 Attack 2: Packet ID

The attack is performed as in attack 1. Since packet ID is not changed at the end of the current epoch, A can infer the trusted nodes of V by getting all packets from V and checking packet IDs. This attack is feasible even after the current epoch is ended.

Remedy (Applied to simulation code) V makes copy of packets received from untrusted nodes and assign new packet IDs. When V encounters a untrusted node, V use the copy of packets, instead of the original packets.

1.2.3 Attack 3: Packet payload

The attack is performed as in attack 2. A can infer the trusted nodes of V by getting all packets from V and checking the payload of the packets.

Remedy First, we assume that the trusted nodes share a symmetric key. When a node sends a packet to its trusted node, it appends a nonce to the payload and encrypts the payload with the symmetric key. Later, when one of the trusted nodes gets the packet, it first decrypt the payload with the symmetric key, replace the nonce with a new one and encrypt the payload.

When V receives the marker packet from A, V tries to decrypt the payload and encrypt the payload. Therefore, A is not able to identify its marker packet by the payload.

2 Experimental Result

2.1 Overview

2.1.1 Simulation model

- ONE simulator, modified default scenario/setting
- Map: Helsinki (4500m * 3500m)
- Simulation running time: 12 hours
- Nodes: 246 (160 humans, 80 cars, 6 trams)
  - Packet buffer: Humans and cars (50MB), trams (500MB).
  - Contact interval: Humans (2 mins 30 secs), cars (1 min), trams (40 secs)
- Packet(message) generation
  - Packet size: 500KB - 1MB
  - Packet generation interval: 35sec - 50sec

  - TTL: 5 hours
  - Packet generation stopped when 5 hours (packet TTL) are left.
  - Total number of packets generated: about 575
- Movement: Random way point, map-based movement.
- Network interface: bluetooth, wlan (determine communication distance and bandwidth)
  - Humans, cars: Bluetooth (Bandwidth: 2Mbps, Communication range: 10m)
  - Trams: WLAN (Bandwidth: 10Mbps, Communication range: 100m)

2.1.2 Anonymous DTN routing setup

- # group: 1
- # nodes in a group: [5%, 10%, 15%, 20%, 25%]
- Epoch: [10mins, 20mins, 30mins, 60mins]
- Ephemeral ID duration: [3 epochs, 6 epochs]
- Base routing protocol: epidemic (flooding)

2.1.3 Assumptions & simplification

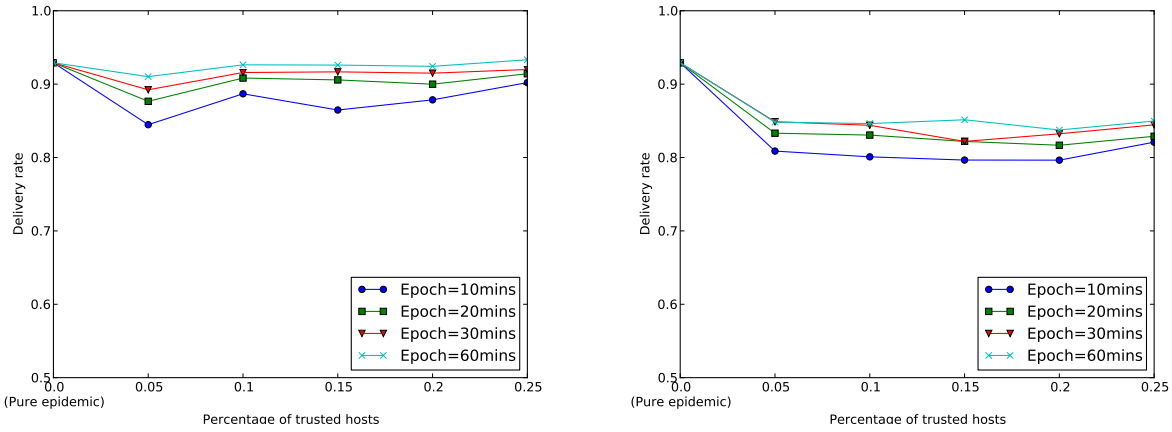
- Strict time sync
  Epoch starts exactly at the same time in all nodes
- No "beacon", "hello", "pull" messages
  Once two nodes are located within a specific distance, they know ephemeral addresses, packet digest, pulling list of each other without any message exchange.
- "Out-of-group" nodes do not use ephemeral IDs.
  Those nodes use permanent IDs which are not changed during the simulation.
- Forwarding policy
  On contact, a node first forwards packets whose destinations are either trusted by the next-hop node or in neighbor list of the next-hop node. Then it tries to forward remaining packets in FIFO manner.

2.2 Results

2.2.1 Communication among all nodes

In this test scenario, every node can send packets to any other nodes. Packet generation follows rules below:

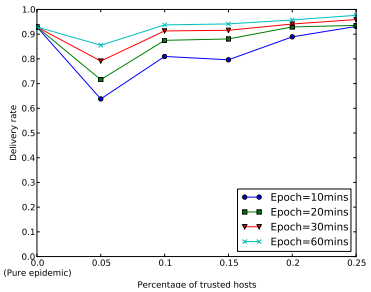
- Nodes belong to the group generate and receive about 20% of overall packets generated during the simulation.
- For the rest 80% of packet generation, sender and receiver are randomly selected from all node.



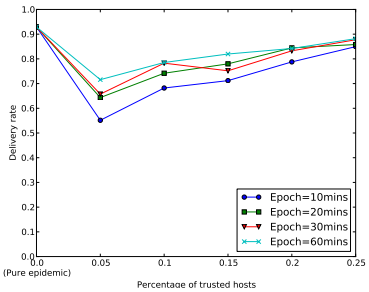
(a) Insecure protocol

(b) Secure protocol

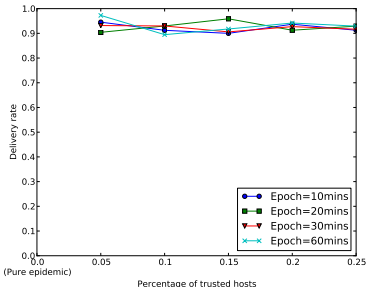
Figure 1: **Overall packet delivery rate.** Delivery rate of pure epidemic routing: 92.91%. Ephemeral ID is valid for 6 epochs. About 5% drop is observed in secure protocol.



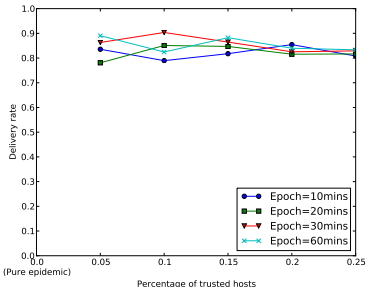
(a) Insecure protocol. In-group to In-group.



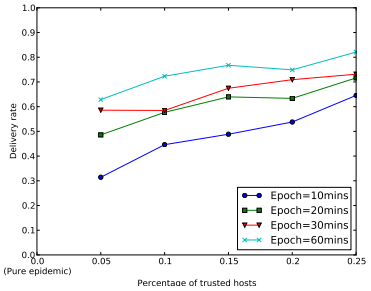
(b) Secure protocol. In-group to In-group.



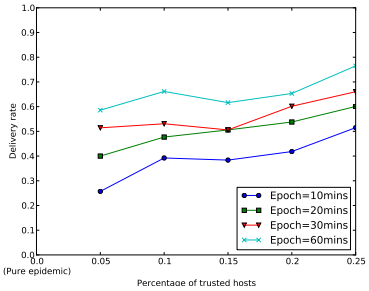
(c) Insecure protocol. In-group to Out-of-group.



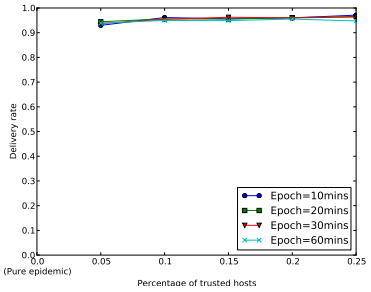
(d) Secure protocol. In-group to Out-of-group.



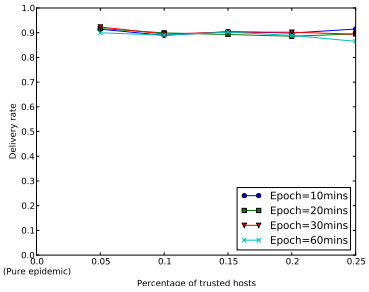
(e) Insecure protocol. Out-of-group to In-group.



(f) Secure protocol. Out-of-group to In-group.

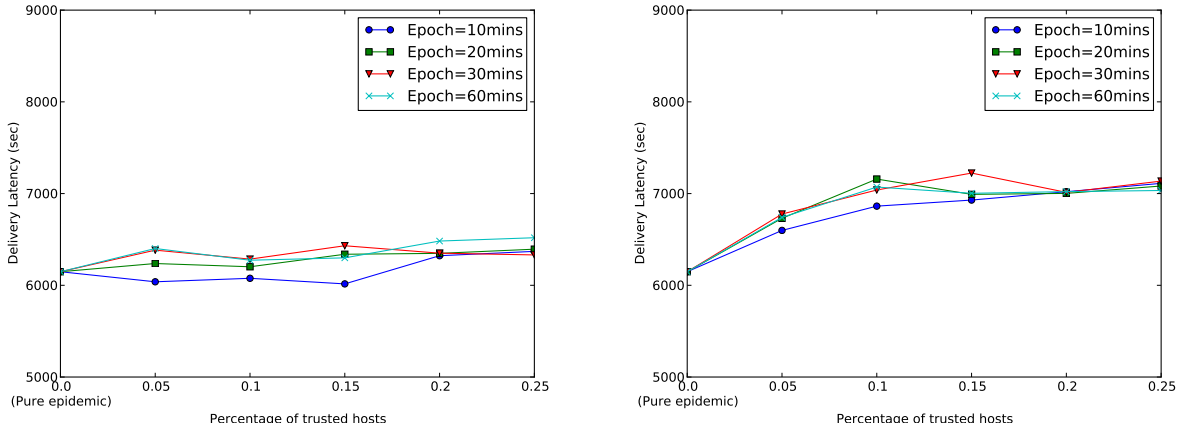


(g) Insecure protocol. Out-of-group to Out-of-group.



(h) Secure protocol. Out-group to Out-group.

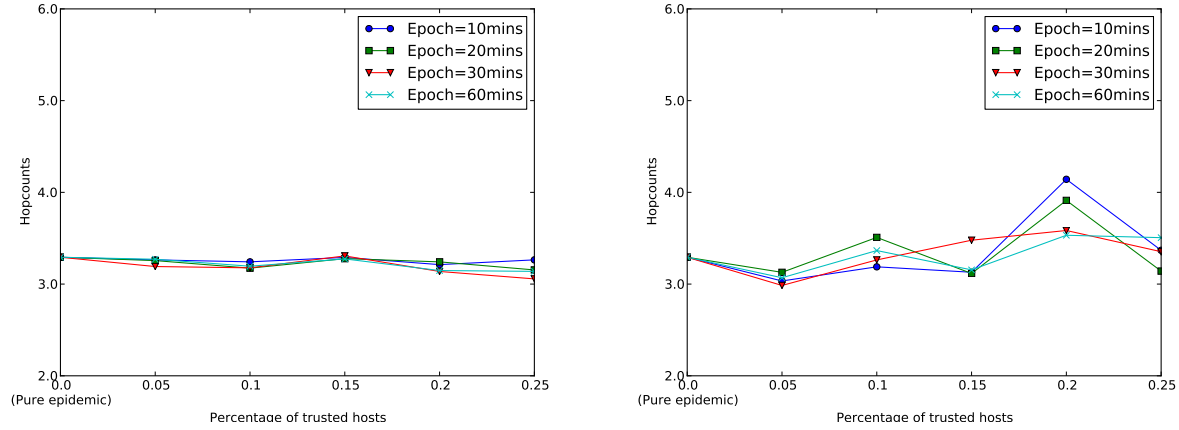
Figure 2: **Detailed packet delivery rate.** Delivery rate of 'In-group' to 'In-group' is decreased significantly in secure protocol.



(a) Insecure protocol.

(b) Secure protocol.

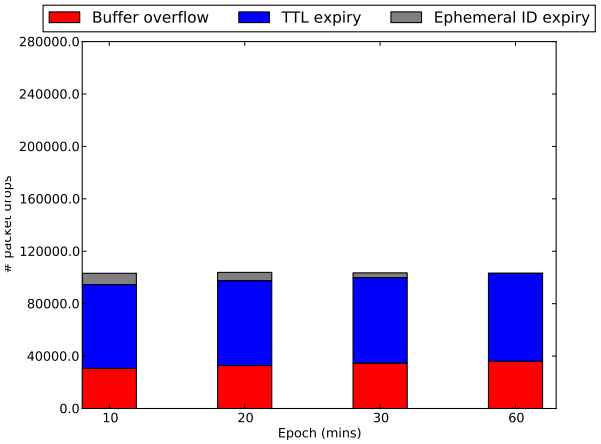
Figure 3: **Overall packet delivery latency.**



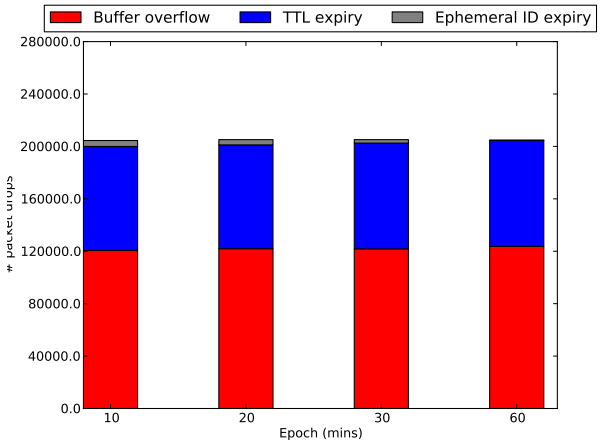
(a) Insecure protocol.

(b) Secure protocol.

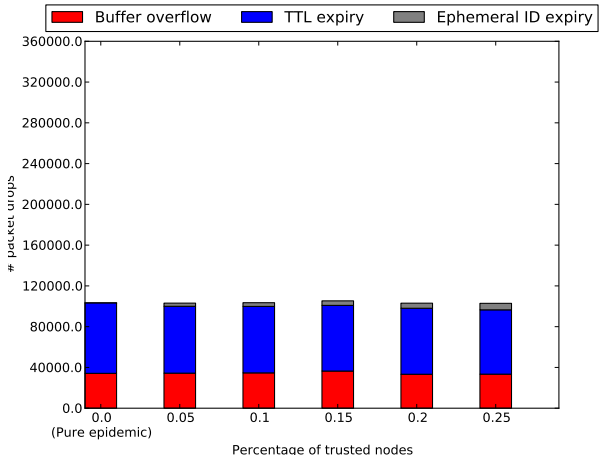
Figure 4: **Overall packet delivery hop count.**



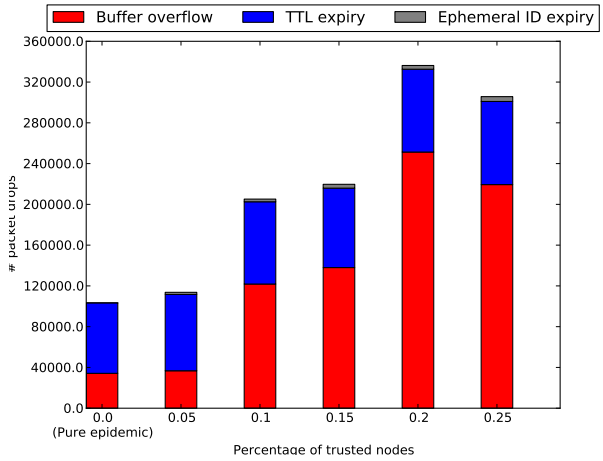
(a) Insecure protocol. Percentage of trusted nodes = 15%.



(b) Secure protocol. Percentage of trusted nodes = 15%.



(c) Insecure protocol. Epoch = 30 mins.



(d) Secure protocol. Epoch = 30 mins.

Figure 5: **Packet drop classification.** New packet ID assignment in secure protocol results in more packet relays and packet drops.