Preserving anonymity of data transfer in open wireless networks using network coding

Mikhail Sergeev
Moscow Institute of Physics and Technologies

September 12, 2014

Outline

Motivation

- Provide a method of preserving anonymity of data transfer without usage of keys

COPE Overview

Main goals:

- Increase throughput
- Achieve anonymity of transmission

Basic concepts:

- Opportunistic Listening
- Opportunistic Coding
- Learning Neighbour State

COPE Overview

Main goals:

- Increase throughput
- Achieve anonymity of transmission

Basic concepts:

- Opportunistic Listening
- Opportunistic Coding
- Learning Neighbour State

COPE Overview

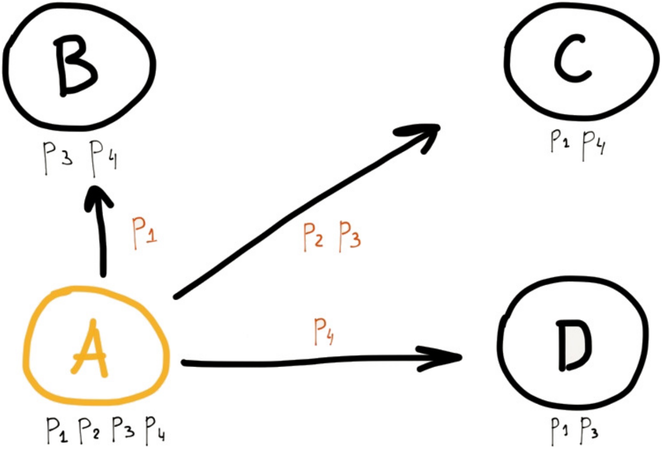


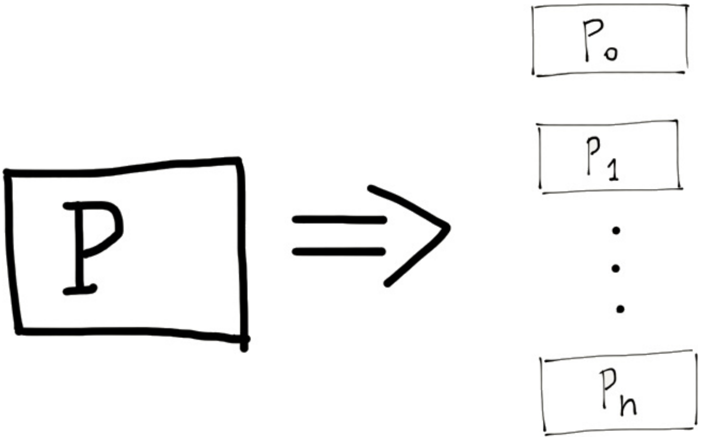
Figure: Example of packet transmission using COPE approach

Splitting the message

Let's take the message **P** and divide it into several parts:
$\mathbf{P} = p_0 \oplus p_1 \oplus \ldots \oplus p_n$, so that $p_0 = \varphi(\cdot)$
φ is a function whose nature will be described later

Each part is now treated as an independent packet except for the p_0, that will not be transferred.
Also each each new packet receives the *message_id* field that determines that all p_i belong to the one message.

Splitting the message



Parties

S - sender of the transmission (source)
R - receiver of the transmission (destination)
N_i - nodes of a considered network; $\exists i_1 : N_{i_1} = S; \quad \exists i_2 : N_{i_2} = R$
N_i and N_j are called neighbours if they can directly exchange packets

Malefactor:
M - we assume a read-only model of malefactor.

Transfer schema

1. S generates a batch of packets, hiding the addresses of receiver and sender inside original message
2. S starts sending COPE packets to neighbour nodes
3. N_i receives packet, stores it into internal storage and resends it to all its neighbours, ignoring *next_hop* parameter
4. if N_j already have the received packet in storage, it discards packet
5. when R has received n different packets (or n linearly independent combinations of packets) with the same message_id, it generates additional packet and produces original message $\varphi(\cdot) \oplus \sum_{i=1}^{n} p_i$

Basic idea

Original message:

$$\mathbf{P} = \varphi_R(\cdot) \oplus p_1 \oplus \ldots \oplus p_n$$

Sent message:

$$\mathbf{P_{sent}} = p_1 \oplus \ldots \oplus p_n$$

Legitimate decoding:

$$\mathbf{P_{leg}} = \mathbf{P_{sent}} \oplus \varphi_R(\cdot) = \mathbf{P}$$

Malicious decoding:

$$\mathbf{P_{mal}} = \mathbf{P_{sent}} \oplus \varphi_M(\cdot) \neq \mathbf{P}$$

Secret packet generation

$\varphi(\cdot) = F(message\_id, recv_{ip})$ Thus, R can recreate this packet on his own. S node knows the addressee and can create it too. Other legitimate nodes cannot recreate it.

In order to hinder to malefactor, F should have rather large time complexity - effectively, he need M times calculating power where M is the number of nodes in the network

Conclusion and possible improvements

- Achieved anonymity of transmission without usage of keys
- Weak level of anonymity - an opportunity for improvements

Thank you for your attention!