Fourteenth International Workshop on Algebraic and Combinatorial Coding Theory
September 7–13, 2014, Svetlogorsk (Kaliningrad region), Russia *pp. 295–299*

Preserving anonymity of data transfer in open wireless networks using network coding [1]

Sergeev Mikhail S.MementoMori@gmail.com
Moscow Institute of Physics and Technology

Abstract. A well-known COPE architecture is generally used to increase network throughput in multi-hop wireless networks by means of network coding. In this paper some modifications were proposed for the parties to maintain anonymity during transmissions. These modifications exploit the idea of inability of malefactor to extract receiver's and sender's address from the message, unless he has successfully intercepted all parts of the message.

1 Introduction

The main difference between wired and wireless networks is the medium: during the wired data transfer the medium - i.e. wire, cannot be arbitrarily accessed by third party, whereas the wireless data transfer exploits air medium which can be easily overheard by anyone. There are several methods to perform secure data transfer over an insecure medium, for example, WPA2-PSK protocol, allowing to secure data transfer over Wi-Fi network by using pre-shared password to prevent third party from accessing the network. But once the malefactor managed to obtain that password, he can get access to the messages.

The main point of this paper is not the security of the data transfer itself, but rather a more narrow task - the anonymity of such transfer, which at times can be the most important part of the secure data transfer.

There are several problems behind the common name of «anonymity». Usually they are separated into three groups: hiding receiver's and sender's identity, hiding sender-receiver relationship and hiding the transfer itself. [4]

Problem of the first type is the main goal of this paper. In order to archive this goal I utilize the approach that was introduced in the COPE - splitting the original message into parts and sending them separately via intermediate nodes. By XOR-ing two different messages and transferring the sum it is possible to get a large increase of network throughput, given that the receiving end will be able to separate the XOR-ed sum.

But it is also possible to use this approach in order to conceal the message contents, especially the header with the receiver's address, so that unrelated nodes (that do not act as receiver or sender) will not be able to identify it. This

[1] This research is partially supported by the grant RFBR 12-07-00122-a

is done by removing one of the parts from the data transfer so that no one, except the legitimate receiver, cannot get the full set of message parts.

This paper is organized as follows. In the next section I will make a short overview of the current COPE method. After that proposed changes and improvements will be introduced.

2 COPE Overview

COPE [1] is an architecture, that provides additional coding layer between MAC and IP layers. This layer utilize network coding and send several packets in one transmission upon finding an opportunity. Another main idea behind this architecture is exploiting the multicast nature of wireless networks.

There are three key methods in the COPE architecture:

- **Opportunistic Listening**, allowing nodes to store overheard packets in a special buffer for a limited amount of time. At the same time each node sends information about packets stored in that buffer.

- **Opportunistic Coding**, allowing nodes to mix different packets in one transmission to increase throughput. This mechanism is based on the list of the stored packets, obtained from the neighbour nodes.

- **Learning Neighbour State**, allowing nodes to estimate the probability of neighbour node's buffer state. This method basically use link weights, assigned to each pair of nodes by routing protocol, to guess whether specific node has specific packet. This is required because Opportunistic Listening can fail to deliver reception report about some packets, which may lead to inefficient coding.

In order to understand the aforementioned description, let's consider an example depicted on the Figure 1. Assuming that node A stores four packets - $P1$, $P2$, $P3$ and $P4$. By means of the Opportunistic Listening and Learning Neighbour State mechanisms, node A discovers the state of each adjacent node's buffer. In order to send packet $P1$ to the node B, $P2$ and $P3$ to the node C and packet $P4$ to the node D, node A have different ways to combine the packets. But some of the combinations are less efficient, as the will result in inability to decode the message on some nodes. For example, if node A will send a $P1 \oplus P3$ packet, nodes B and C will be able to comprehend the transmission by XOR-ing the message with either $P1$ or $P1$ packet, that is already stored in the buffer, but the node D will not be able to receive the packet. On the contrary, sending $P1 \oplus P3 \oplus P4$ packet will result in successful decoding of message on each node, provided that this packet will be successfully delivered to all recipients. Thus, it is possible to formulate a formal criteria for Opportunistic Coding possibility [2]

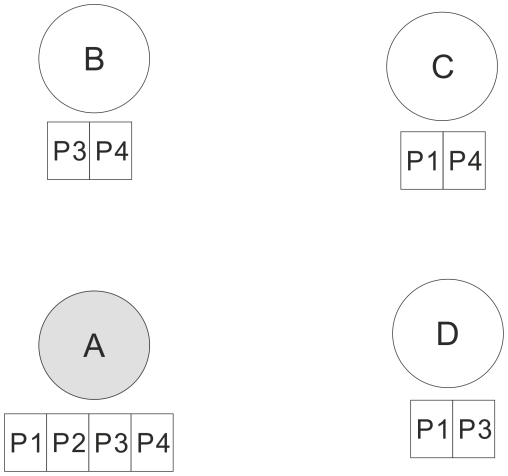

Figure 1: COPE Transmission Example

To transmit n packets, $p_1, ..., p_n$, to n next-hops, $r_1, ..., r_n$, a node can XOR the n packets together only if each next-hop r_i has all $n-1$ packets p_j for $j \neq i$.

3 Proposed changes

After inspection of COPE structure we can move to the proposed modifications. Let's say that sender wants to send message **P** to the node R1. In order to do so header, containing address of receiver, is included into the original message **P**, which is later split into a sum of packets $\mathbf{P} = \sum_n P_i$. According to the COPE approach each of these packets is transferred separately in some linear combination of packets. Receiver have to get at least n linearly independent combination in order to be able to unambiguously restore the **P**. But if sender will not use one of the packets P_i (e.g. P_1) in the transfer, receiver, as well as any other node in the network, including malefactor, will not be able to restore the original message, because $n-1$ packets cannot form n linearly independent combinations.

But now another problem is being raised - how can receiver decode original message if he does not have enough linear combinations of packets? The solution can be found if the receiver is able to generate or restore the missing packet on his own. To allow such restoration, the unused packet have to be specially generated. The ways to produce the secret packet are discussed in the next section. Now we assume that secret packet was generated using a function $P_1 = F(addr_s, addr_r, counter)$, where $addr_s$ - sender's address, $addr_d$ - receiver's

address and *counter* is a random sequence, that is transferred in the packet. So, after generating secret packet, sender XORs it with the original message and split resulting message into $n-1$ parts, that are transferred separately.

After receiving packets legitimate receiver saves them in the buffer and once collected $n-1$ independent linear combinations of packets, node generates secret packet and use it as an n^{th} linear combination, allowing to easily decode original message by solving a system of linear equations. At the same time all other nodes due to the ignorance of receiver's address are unable to generate the required packet and cannot create missing linear combination, that is required to restore the original message.

4 Secret packet generation

In the previous section the secret packet generation problem was described - there should be a way to create packet that can be independently recreated on both sides of transmission. There are several solutions for this problem, one of the simplest is to make it pre-shared, so that contacting parties need to use some other secure medium (e.g. personal meeting) to exchange the packet. Another simple way is to generate secure packet just before the transmission using Diffie-Hellman key exchange protocol.

Also, this paper proposes two alternative ways to generate secret packet. One of them exploits some considerable time-consuming function like calculation of MD5 hash of a large file or some proof-of-work algorithm [3]. Using this method sender generates random counter and calculates this function, using addresses and counter as parameters - in the case of MD5 hash sum calculation we can take specified large file, append addresses and counter to its end and perform hash sum calculation. The result can be used as a secret packet that can easily restored on the receiving end, but malefactor needs M times more calculating power in order to identify the receiver, where M is a number of nodes in network.

Another approach allows sender to use all parts of the message, but every part should be separately XOR-ed with secret packet, generated as described above with its own counter. This way when receiver gets new packet, he tries to decode it using his own address as a key and in case of successful decoding, stores packet in a separate buffer. When necessary number of packets is stored in that buffer, receiver can restore the original message. In case of malefactor, he is required to find n secret packets before decoding the original message and retrieving to destination address, which may take considerable amount of calculating power.

5 Conclusion

Method allowing to organize anonymous data transfer across insecure wireless network is introduced. Several options for maintaining anonymity were proposed by exploiting several algorithms of generating secret packets, that a used to hide message from adversaries.

References

[1] Katti Sachin, et al. *XORs in the air: practical wireless network coding,* IEEE/ACM Transactions on Networking (TON), 16.3 (2008): 497-510

[2] Zhu Xiaoyan et al. *A batched network coding scheme for wireless networks,* Wireless Networks, 15.8 (2009): 1152-1164.

[3] Coelho Fabien *Exponential Memory-Bound Functions for Proof of Work Protocols,* IACR Cryptology ePrint Archive 2005 (2005): 356.

[4] Kang Ruogu Stephanie Brown and Sara Kiesler. *Why do people seek anonymity on the internet?: informing policy and design.* Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.