Analysis of a Homomorphic MAC-based Scheme against Tag Pollution in RLNC-Enabled Wireless Networks

Alireza Esfahani*, Georgios Mantas*, Valdemar Monteiro*, Kostas Ramantas†, Eftychia Datsika†, and Jonathan Rodriguez*
*Instituto de Telecomunicações (IT)
Campus Universitário de Santiago, P 3810-193, Aveiro, Portugal
Email: {alireza, gimantas, vmonteiro, jonathan}@av.it.pt
†IQUADRAT
Passeig Sant Joan 89, 08009 Barcelona, Spain
Email:{kramantas, edatsika}@iquadrat.com

***Abstract*—Network Coding-enabled wireless networks are vulnerable to data pollution attacks where adversary nodes inject into the network polluted (i.e. corrupted) packets that prevent the destination nodes from decoding correctly. Even a small proportion of pollution can quickly propagate into other packets via re-coding, occurred at the intermediate nodes, and lead to resource waste. Therefore, during the past few years, several solutions have been proposed to provide resistance against data pollution attacks. One of the most well-known solutions is Homomorphic Message Authentication Code (HMAC). However, HMAC is susceptible to a new type of pollution attacks, called tag pollution attacks, in which a malicious node randomly modifies MAC tags appended at the end of the transmitted packets. To address this issue, we have recently proposed an HMAC-based scheme making use of two types of MAC tags to provide resistance against both data pollution attacks and tag pollution attacks. In this paper, we steer our focus on improving the resistance of our proposed scheme against tag pollution attacks by decreasing the number of MACs. Finally, we analyze the impact of the total number of MACs on the bandwidth overhead of the proposed scheme.**

***Index Terms*—Network coding, homomorphic message authentication code, data pollution attack, tag pollution attack.**

I. Introduction

Nowadays, Network Coding (NC) has emerged as a new communication paradigm that can provide significant benefits to networks in terms of bandwidth, robustness to packet losses, delay and energy consumption [1]–[3]. NC was introduced for the first time by Ahlswede et al. in [4] and its main principle is that the intermediate nodes not only store and forward the incoming packets but also employ coding operations to mix them. In [5], Li et al. further proposed Linear Network Coding (LNC) which is based on linear combinations of the incoming packets at the intermediate nodes. Then, Ho et al. in [6] proposed Random Linear Network Coding (RLNC) as a fully distributed method for performing NC. Each node in RLNC-enabled networks selects a set of coefficients randomly and uses them to make linear combinations of the incoming packets.

Due to the mixing and re-coding of packets, RLNC-enabled networks are vulnerable to a wide range of attacks. More precisely, RLNC-enabled networks are more susceptible to pollution attacks than the traditional store-and-forward networks. Even a small number of polluted (i.e., modified) packets can infect a large number of downstream nodes in the network. If a data pollution attack is not detected by the intermediate nodes, then the sink nodes will not be able to recover the original packets correctly. This has as a result network resources waste.

Therefore, during the past few years, a lot of research effort has been placed on schemes protecting RLNC-enabled networks from pollution attacks. There are three main categories of these schemes: *corrupted packets correction* [7], [8], *corrupted packets detection* [9]–[15], and *adversary nodes localization* [16], [17]. The corrupted packets correction schemes can only detect data pollution attacks at the sink side. On the other hand, the corrupted packets detection schemes make use of cryptographic schemes, such as homomorphic hash functions [9], homomorphic signatures [10] and homomorphic MACs [11]–[13] to enable the intermediate nodes to detect data pollution attacks. Moreover, the adversary nodes localization schemes target detecting the exact location of adversary nodes and make those nodes unable to propagate polluted packets [17].

Our research effort is focused on homomorphic MAC-based schemes since they are low-complexity solutions for data pollution detection. In these schemes, a MAC or tag is appended to the end of the transmitted packets, in order to protect their integrity. However, the homomorphic MAC-based schemes are susceptible to tag pollution attacks. They are more sophisticated pollution attacks where an adversary modifies packet's tags instead of the packet's content. Hence, a packet with polluted tags is possible to travel multiple nodes before it is detected.

In this sense, we have recently proposed in [18] a new homomorphic MAC-based scheme, which detects both data

pollution attacks and tag pollution attacks in RLNC-enabled wireless networks. To protect the integrity of each packet and corresponding tags (i.e., MACs), the source node generates multiple MACs for each packet.

In this paper, we intend to improve the resistance of our proposed scheme against tag pollution attacks by decreasing the number of MACs. Furthermore, we intend to analyze the impact of the total number of MACs on the bandwidth overhead of our proposed scheme in [18].

The rest of the paper is outlined as follows. In Section II, the background is given. In Section III the impact of the number of tags on the tag pollution attack probability is presented. Furthermore, in Section IV the impact of the number of tags on the bandwidth overhead is given. Finally, Section V concludes the paper.

II. BACKGROUND

A. RLNC-Enabled Network Model

We consider a RLNC-enabled network model which is defined by a triple (G, S, I), where G, S, and I represent a multigraph model of the network, a source node, and non-source (intermediate and destination) nodes, respectively:

- **Directed multigraph** G.
  Each graph G includes a set of nodes V and a set of links E. For instance, in Fig 1, the set of nodes is $V = \{S, I_1, I_2, I_3, I_4, I_5, I_6, I_7\}$, and the set of links is $E = \{e_1, ..., e_{11}\}$.
- **Source node** S.
  There is a source node S who wants to multicast its messages. Due to RLNC properties, each message is divided into a sequence of packets in each generation. Each packet consists of a number of symbols. Only the packets which belong to the same generation can be decoded by destination nodes.
- **A set of non-source nodes** I.
  Apart from the source node S which encodes the packets, we define intermediate and destination nodes in a set of nodes which recode and decode the packets.

Due to the above mentioned network model, we consider a traditional multicast scenario where the source S wants to send its messages to a number of destination nodes(I_6, I_7) through several intermediate nodes ($I_1, ..., I_5$) (See Fig 1). More precisely, the source splits each message into a sequence of packets and partitions them into generations. Our assumption is based on that each generation consists of m packets $\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, ..., \underline{\mathbf{u}}_m$. Each packet $\underline{\mathbf{u}}_i$, for $1 \leq i \leq m$, is represented by a vector $\underline{u}_1, \underline{u}_2, ..., \underline{u}_n$, where each $\underline{u}_i$ $(1 \leq i \leq n)$ is in the finite field $\mathbb{F}_p$. After that, the source S creates an augmented packet $\mathbf{u}_i$ for each packet by prefixing $\underline{\mathbf{u}}_i$ with the i^{th} unit vector of dimension m.

The augmented packet is illustrated in Fig 2 and represented as a row vector in the finite field $\mathbb{F}_p^{m+n}$ as follows:

$$\mathbf{u}_i = (\overbrace{\underbrace{0, \cdots, 0}_{i-1}, 1, 0, \cdots, 0}^{m}, \underline{u}_{i,1}, \cdots, \underline{u}_{i,n}) \in \mathbb{F}_p^{m+n} \quad (1)$$

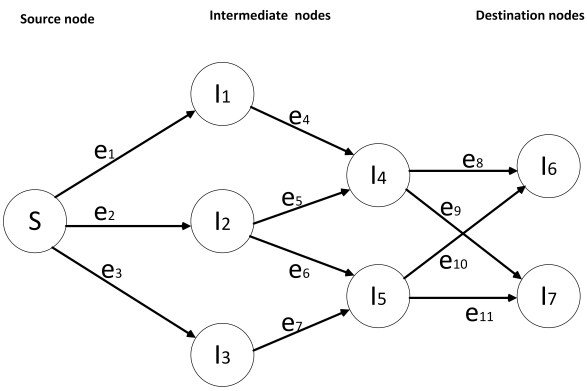

Fig. 1. A simple multicast scenario which includes a source, some intermediate, and two destination nodes.

Finally, the source S tries to multicast these augmented packets to its neighbour nodes. Due to RLNC scheme, an intermediate node makes random linear combinations of the incoming packets, buffers them temporarily and creates a new coded packet y , which is a linear combination of a number of h augmented packets $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_h$ belonging to the same generation. By using h random coefficients $c_i \in \mathbb{F}_p$ which are selected by each intermediate node, the new coded packet is represented as follows:

$$y = \sum_{i=1}^{h} c_i \mathbf{u}_i \quad (2)$$

Upon receiving m linear independent packets, a destination node can start decoding the original packets using Gaussian eliminations [6]. The received packets are linear independent when they belong to the linear subspace spanned by the original augmented packets. This is represented as follows::

$$y \in (\alpha_{\mathbf{1}} \mathbf{u}_1 + \cdots + \alpha_{\mathbf{m}} \mathbf{u}_m) \quad (3)$$

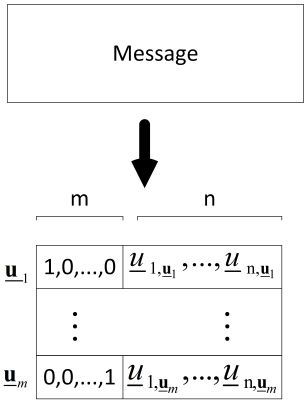

Fig. 2. A RLNC model where each message is divided into m packets in each generation. Each packet includes n symbols of finite field $\mathbb{F}_p$, and a unit vector of dimension m.

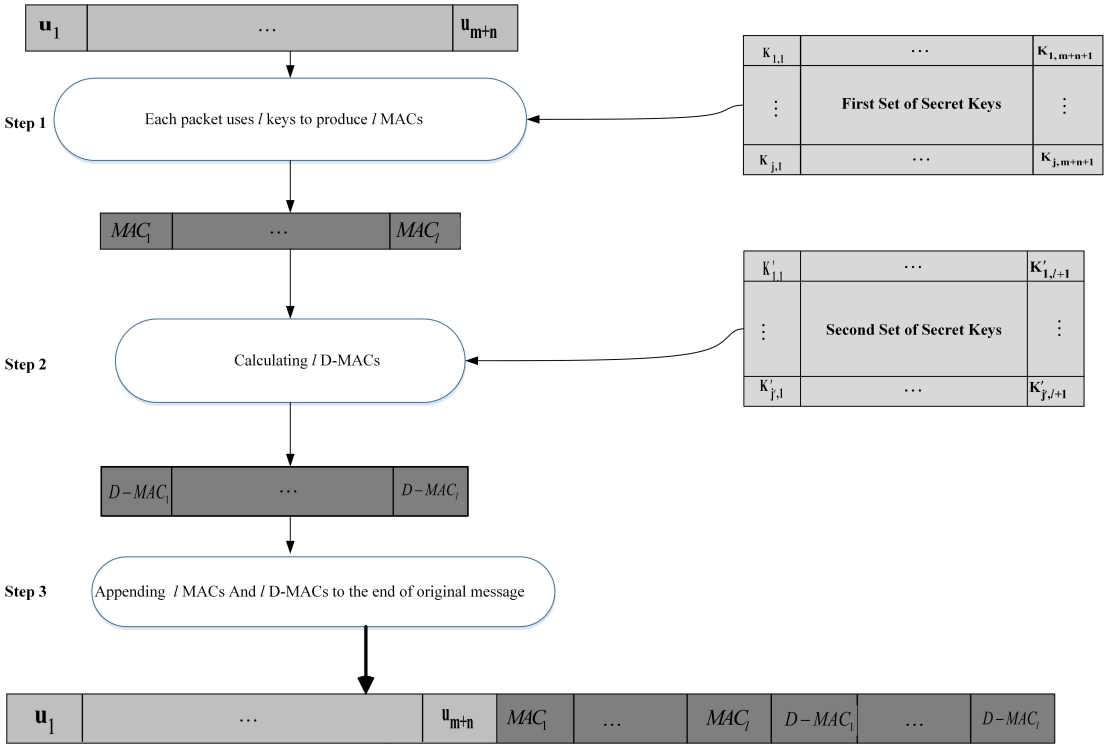

Fig. 3. An overview of our proposed scheme in [18]

where $\alpha_i \in \mathbb{F}_p$. Otherwise, by transmission errors or pollution attacks, y is discarded and it is denoted as:

$$y \notin (\alpha_{\mathbf{1}}\mathbf{u}_1 + \cdots + \alpha_{\mathbf{m}}\mathbf{u}_m) \tag{4}$$

B. Key Distribution Model

In our key distribution scheme, a Key Distribution Center (KDC) assigns two key pools to the source node in order to generate tags for each packet. Then, KDC assigns a set of keys from each key pool to each node (i.e. intermediate and destination) randomly. These keys are used by each node to verify the integrity of the received packets. According to the *Definition 1*, if the key assignment is done appropriately, no coalition of c nodes can fool another node.

Definition 1. A set system (A, F), where A is a finite set of elements and F is an ordered set of subsets of A, is called a c cover-free family (c-CFF) if, for any community of c sets $x_1, ..., x_c \in F$, there is other set $z \in F$, who has

$$z \nsubseteq \bigcup(x_1, ..., x_c) \tag{5}$$

In other words, our key distribution model is based on the fact that there is not any community including more than c compromised nodes. The details of our key distribution model is given in [18].

C. The Proposed Dual-Homomorphic MAC Scheme

In our proposed Dual-HMAC scheme [18], we make use of two types of tags in order to provide resistance against data pollution and tag pollution attacks. The first type of tags (i.e., MACs) is responsible to check the integrity of the packet, and the second type of tags (i.e., D-MACs) is responsible to check the integrity of the MACs. In addition, we use two sets of secret keys for generating the MACs and the D-MACs. Our scheme is based on orthogonality properties which have been presented in [13]. Thus, to calculate the MAC of each augmented packet, Step 1, in Fig 3, uses the following formula (formula (3) of [18]):

$$\frac{\sum_{p=1}^{m+n} \mathbf{u}_{i,p} K_{j,p}}{K_{j,m+n+1}} \tag{6}$$

As a result, we produce l MACs for each augmented packet. Then, each D-MAC is calculated according to

$$\frac{\sum_{p=1}^{l} MAC_{i,p} K'_{j,p}}{K'_{j,l+1}} \tag{7}$$

In other words, totally $L = 2 * l$ tags are appended to each augmented packet.

III. THE IMPACT OF THE NUMBER OF D-MACS ON THE TAG POLLUTION ATTACK PROBABILITY

The proposed scheme provides resistance against data pollution attack which can occur in the intermediate nodes, due to the l MACs that ensure integrity for the transmitted packets. However, our scheme can provide partially resistance against tag pollution attacks, since the l D-MACs are not protected

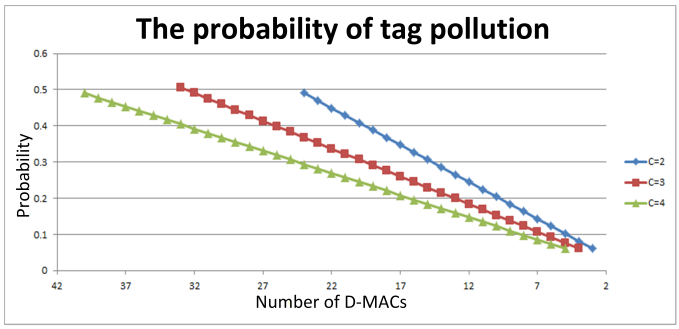

Fig. 4. The probability of tag pollution in terms of the number of D-MACs, where c is the number of compromised nodes.

by additional mechanisms ensuring their integrity. In order to address this issue, in this paper, we focus our work on analysing how the different number of MACs and D-MACs appended to each packet can reduce the probability of tag pollution attack. However, our target is to maintain the same bandwidth overhead for our scheme compared to the scheme presented in [13] where only MACs and a signature are used to provide resistance against tag pollution attacks. For this reason, in our scheme, the total number of appended tags (i.e., MACs and D-MACs) to each transmitted packet is calculated by the following formula which is based on [13], [19]:

$$L = 2 \cdot e \cdot (c+1) \cdot \ln q \tag{8}$$

where:

c : is the number of compromised nodes.

q : is a security parameter.

In other words, the L represents the total number of MACs and D-MACs appended to each packet. Therefore, the probability of an attacker to achieve tag pollution in our scheme is given by the following formula:

$$Pr_{tag\_pollution} = \frac{Number\ of\ D\text{-}MAC}{L} \tag{9}$$

According to the formula (9), the probability of tag pollution decreases as the number of D-MACs decreases and the L is fixed.

In order to assess the impact of the decreasing number of D-MACs on the probability of tag pollution, we run a Matlab simulation for calculating the probability of tag pollution based on the formulas (8) and (9). Specifically, we run the simulation for 3 different numbers of compromised nodes ($c = 2, 3$ and 4) and q equals to 1000. Fig 4 plots the probability of tag pollution in terms of the number of D-MACs for $c = 2, 3$ and 4 .

In Fig 4, the plot shows that the probability of tag pollution attack decreases as the number of D-MACs decreases. It is what we expected according to the formula (9) and the fact that the L has to be fixed. Specifically, we notice in Fig 4 that the maximum value is achieved when the number of D-MACs is equal to $\frac{L}{2}$, which is our initial assumption in the presentation of our proposed scheme in [18].

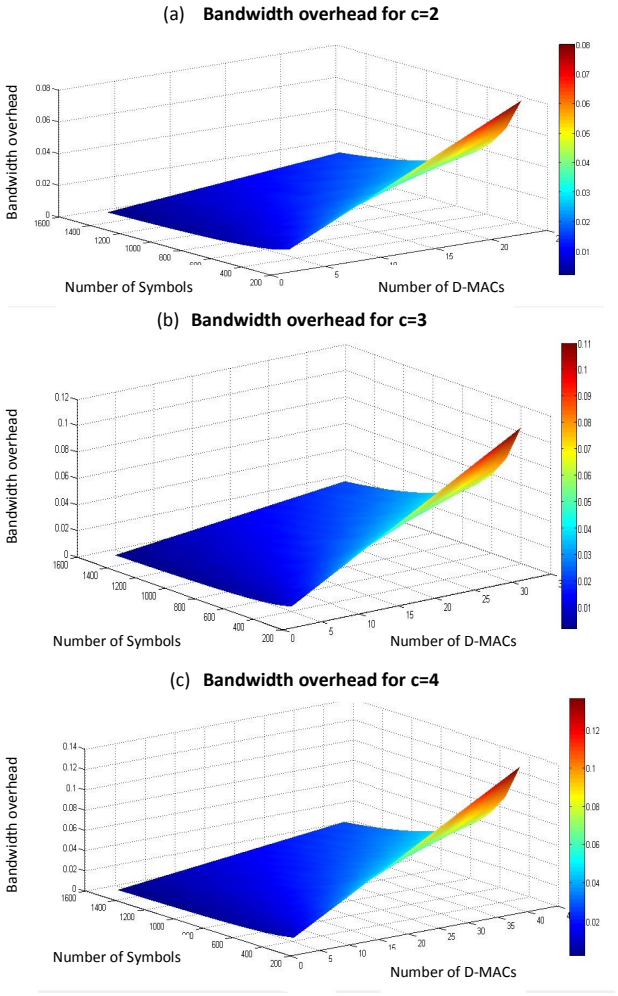

Fig. 5. The bandwidth overhead in terms of the number of D-MACs and the total number of symbols, where c is the number of compromised nodes.

On the other hand, the minimum value of the probability of tag pollution is placed when the number of D-MACs is equal to 3, 4, and 5 for $c = 2, 3$ and 4 respectively.

It is expected since each node has at least one secret key more than the number of compromised nodes, c, according to [18]. Our experimental result shows that not only we still maintain the same bandwidth overhead compared to the scheme presented in [13], but we also reduce the tag pollution probability.

IV. THE IMPACT OF THE NUMBER OF D-MACS ON THE BANDWIDTH OVERHEAD

The bandwidth overhead in terms of the number of the tags in our proposed scheme can be calculated by the following formula (10) which is based on [13]:

$$Overhead_{Bandwidth} = \frac{L}{m+n} \tag{10}$$

where,

L : is the total number of MACs and D-MACs and is given by the formula (8).

$m+n$: is the total number of symbols of each augmented packet.

According to the formula (10), the bandwidth overhead is related to the number of D-MACs and the total number of symbols of the augmented packets. In order to assess the impact of the number of D-MACs on the bandwidth overhead, we run a Matlab simulation for calculating the bandwidth overhead for three different values of the number of compromised nodes $(c = 2, 3, 4)$. As it is shown in Fig 5, by decreasing the number of D-MACs and increasing the total number of symbols, the bandwidth overhead is decreased.

V. Conclusion

In this paper, we have focused on the improvement of the Homomorphic MAC-based scheme that we have presented in our previous work [18] in terms of its resistance against tag pollution attacks. Our analysis showed that the probability of tag pollution can be reduced by decreasing the number of D-MACs appended to the transmitted packets. Finally, our analysis showed that by decreasing the number of D-MACs and increasing the total number of symbols, the bandwidth overhead is decreased.

As future work, we plan to further evaluate and improve our proposed scheme in order not only to achieve partially tag pollution attack detection but to be tag pollution immune.

Acknowledgment

The research leading to these results has received funding from the European Communitys Seventh Framework Program [FP7/2007-2013] under grant agreement n 285969 [CODELANCE]. The first author would like to acknowledge support of the Fundacão para a Ciência e a Tecnologia (FCT - Portugal), through Grant number: SFRH/BD/102029/2014.

References

[1] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, *Trading structure for randomness in wireless opportunistic routing*. ACM, 2007, vol. 37, no. 4.

[2] D. Petrović, K. Ramchandran, and J. Rabaey, "Overcoming untuned radios in wireless networks with network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2649–2657, 2006.

[3] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4. IEEE, 2005, pp. 2235–2245.

[4] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

[5] S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *Information Theory, IEEE Transactions on*, vol. 49, no. 2, pp. 371–381, 2003.

[6] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, 2006.

[7] T. Ho, B. Leong, R. Koetter, M. Médard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks with random network coding," *Information Theory, IEEE Transactions on*, vol. 54, no. 6, pp. 2798–2803, 2008.

[8] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of byzantine adversaries," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 616–624.

[9] C. Gkantsidis, P. Rodriguez *et al.*, "Cooperative security for network coding file distribution." in *INFOCOM*, vol. 3, 2006, p. 5.

[10] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*. IEEE, 2007, pp. 556–560.

[11] S. Agrawal and D. Boneh, "Homomorphic macs: Mac-based integrity for network coding," in *Applied Cryptography and Network Security*. Springer, 2009, pp. 292–305.

[12] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "Ripple authentication for network coding," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[13] P. Zhang, Y. Jiang, C. Lin, H. Yao, A. Wasef, and X. Shen, "Padding for orthogonality: Efficient subspace authentication for network coding," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1026–1034.

[14] M. N. Krohn, M. J. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*. IEEE, 2004, pp. 226–240.

[15] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[16] A. Le and A. Markopoulou, "Tesla-based defense against pollution attacks in p2p systems with network coding," in *Network Coding (NetCod), 2011 International Symposium on*. IEEE, 2011, pp. 1–7.

[17] ——, "Cooperative defense against pollution attacks in network coding using spacemac," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 2, pp. 442–449, 2012.

[18] A. Esfahani, D. Yang, G. Mantas, A. Nascimento, and J. Rodriguez, "Dual-homomorphic message authentication code scheme for network coding-enabled wireless sensor networks," *International Journal of Distributed Sensor Networks*, 2015.

[19] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 1999, pp. 708–716.