A Null Space-based MAC Scheme against Pollution Attacks to Random Linear Network Coding

Alireza Esfahani, Georgios Mantas, Jonathan Rodriguez, Alberto Nascimento, and Jose Carlos Neves
Instituto de Telecomunicações (IT), Aveiro, Portugal
alireza, gimantas, jonathan, jneves@av.it.pt, ajn@uma.pt

***Abstract*— Network Coding has significantly shown the achievable throughput and robustness in wireless Networks. However, network coding-enabled networks are susceptible to pollution attacks where a small number of polluted messages will propagate due to recoding and corrupt bunches of legitimate messages. Several lightweight Homomorphic Message Authentication Code (HMAC) schemes have been proposed for protecting the transmitted data against pollution attacks; however, most of them are not appropriate for wireless networks or cannot resist tag pollution attacks. In this paper, we present a computationally efficient null space-based homomorphic MAC scheme, for network coding-enabled wireless networks. The proposed scheme makes use of two types of tags (i.e., MACs and D-MACs) to provide resistance against data pollution and tag pollution attacks. Furthermore, we demonstrate that due to its lightweight nature, our proposed scheme incurs a minimal complexity compared to other related schemes.**

Keywords*— *Network coding, security, homomorphic message authentication code, data pollution attack, tag pollution attack.

I. INTRODUCTION

Network Coding (NC) is a promising technology which is used nowadays in various applications over a wide spectrum of networks, such as wireless mesh networks [1], wireless sensor networks [2] and peer-to-peer networks [3]. NC was introduced for the first time by Ahlswede et al [4]. In contrast to the classical commodity flow, in which the information is only routed or replicated, in a NC-enabled network the information flow can also employ coding operations at the nodes. Linear Network Coding (LNC) which is based on linear combinations of the incoming packets was appeared further [5]. Random linear Network Coding (RLNC) was studied by Ho et al. in [6] as a fully distributed method for performing network coding. There is a possibility that each node in the network independently and randomly selects a set of coefficients and uses them to make linear combinations of the data symbols. In particular, RLNC was introduced by showing all properties of network coding with achieving the maximum capacity.

Despite its benefits, RLNC-enabled networks are more susceptible to pollution attacks than the traditional store-and-forward ones. Even a small number of polluted (i.e., modified) messages can infect a large number of downstream nodes because the pollution propagates via recoding. If a data pollution attack is not detected at the forwarders (i.e., intermediate nodes), then the sink nodes will not be able to recover the source messages correctly.

So far, several information-theoretic schemes [7, 8] and cryptographic schemes [9-18] have been proposed to secure network coding against data pollution attacks. However, information-theoretic schemes can only detect data pollution attacks at the sink side. On the other hand, cryptographic schemes, such as homomorphic hash functions [10], signatures [19, 20] and homomorphic MACs [15] enable the intermediate nodes to detect data pollution attacks. Among the proposed cryptographic schemes, MAC can be used as a low-complexity solution for data pollution detection. More explicitly, a MAC or tag is a small piece of information appended to the end of the message packet. This piece of information is the output of a MAC function taking as inputs the message packet and a secret key. However, MAC is vulnerable to tag pollution attacks. These attacks are more sophisticated pollution attacks where attackers pollute (i.e., modify) tags instead of the messages' content. Hence, a message packet with polluted tags is possible to travel multiple hops before it is detected and cause to a waste of bandwidth. Therefore, a number of MAC-based schemes have been also proposed in order to address this issue.

The RIPPLE, TESLA and TESLA-Based schemes are resistant against tag pollution attacks [16, 21, 22]. However, in these schemes, each node is required to be synchronized with the other nodes. Moreover, the authors in [18] presented a hybrid authentication scheme which is based on homomorphic MACs and homomorphic signatures. Their scheme can resist against data and tag pollution. However, the verification phase increases the computational complexity and delay of the scheme.

Hence, to detect both data pollution and tag pollution attacks in an efficient way, we propose an efficient null space-based homomorphic MAC scheme, for RLNC-enabled wireless networks. According to our scheme, the source generates multiple MACs and D-MACs for each message packet. The former one ensures the integrity of the packet and the latter one ensures the integrity of the MACs. We

show that our scheme can resist against data and tag pollution attacks.

The rest of the paper is outlined as follows. In Section II, the background is given. In Section III, the proposed scheme is discussed. Furthermore, in Section IV the correctness of the proposed scheme is given. In Section V, we provide a discussion. Our performance evaluation takes place in Section VI. Finally, Section VII concludes the paper.

II. BACKGROUND

A. Random Linear Network Coding-Enabled Network

To study secure network coding, we consider a triple (G, S, I) which consists of the following components:

- Directed multigraph G: We consider a pair of (V, E) as a directed acyclic graph where V and E are the node and edge set of G, respectively.
- Source node S: In our network model, we have a source S which wants to multicast its messages. To achieve that, each message is divided into a sequence of packets and these packets are sent to the destinations. Each packet consists of a number of symbols.
- A set of Intermediate and Sink nodes I: We define relay and sink nodes in a set of nodes which is defined as: $I = \{\forall x \in V - \{S\}\}$.

In this sense, a traditional multicast scenario is used where the source S wants to send its native data packets to multiple destinations. We consider a basic scenario based on a line network in which there is a RLNC based communication via three nodes: A source node encodes the native data packets and floods them into the network; an intermediate node recodes the incoming encoded packets and forwards them toward the sink nodes, and the sink nodes decode the incoming coded packets to extract the native packets. Prior to transmission, the source divides each native data packets into a sequence of packets and partitions them into generations. Thus, we consider that each generation consists of m packets denoted as $\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, ..., \underline{\mathbf{u}}_m$. Each packet $\underline{\mathbf{u}}_i$, for $1 \le i \le m$, is represented as a vector $\underline{u}_1, ..., \underline{u}_n$ in the finite field $\mathbb{F}_p^n$. Then, the source S generates an augmented packet $\mathbf{u}_i$ for each packet $\underline{\mathbf{u}}_i$ by prefixing $\underline{\mathbf{u}}_i$ with the i^{th} unit vector of dimension m. A simple description of RLNC is depicted in Figure 1.

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

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

After that, the source S transmits these augmented packets to its neighbour nodes.

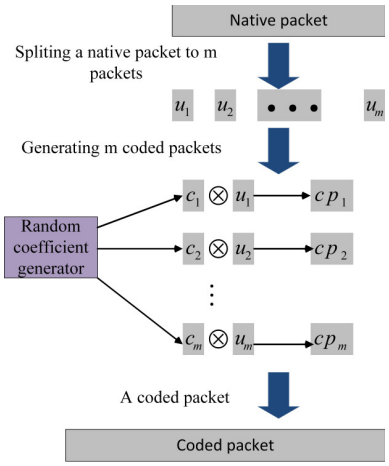

Figure 1. A simple description of RLNC scenario.

During transmission, an intermediate node buffers its received packets $\mathbf{u}_i$ temporarily and creates a 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. The coded packet is represented as follows:

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

where each $c_i \in \mathbb{F}_p$ is a random coefficient chosen by each intermediate node. A coded packet y is considered to be valid if it is in the linear subspace spanned by the original augmented packets. This is denoted as $y \in Span\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$. In fact, when y is valid, these linear combination coefficients are the first m symbols of the packet y. Otherwise, y is invalid and it is denoted as $y \notin Span\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$, which may be caused by transmission errors or pollution attacks. After transmission, when a sink node has obtained m linearly independent coded packets, it can decode those using Gaussian eliminations, so as to recover the original packets that belong to the same generation [23].

B. Null Space Properties

Typically, Null space is the set of solutions to the equation $A.x = 0$, where 0 is understood as the zero vector. Consider a linear map represented as a $m \times n$ matrix A with coefficients in a finite field $\mathbb{F}_p$. The main idea in Null space is based on the randomization and the subspace properties of random network coding. In the other hand, in RLNC, the source native packets form a subspace and any linear combination of these native packets belongs to that same subspace. For more simplicity, we use Null keys terminology instead of Null space. However, these null keys are not randomly generated but calculated at the source node

(e.g. via Singular Value Decomposition (SVD)), to ensure their orthogonality to the subspace spanned by the data vectors in a generation ($Span\{\mathbf{u}_1,\cdots,\mathbf{u}_m\}$). Similar to the source native packets, the null keys go through random linear combinations, which makes it hard for a malicious node to identify them at its neighbors.

C. Adversary Models

For the above mentioned network coding-enabled model, we consider that the source is always trusted and there is no possibility to forge it, but the intermediate and sink nodes can be compromised. The adversary is able to wiretap all the data packets that are transmitted over a network. The adversary's goal is to achieve pollution attack. There are two types of pollution attack:

Data pollution attack: an adversary can inject fake data packets into the network. The objective of data pollution attack is to pass the verification of other innocent nodes, and to cause incorrect decoding at the sink node, as well as wasting of bandwidth.

Tag pollution attack: an adversary injects a corrupted packet consisting of correct data but modified tags to the network. The objective of tag pollution is to discard the correct data packets due to the corresponding corrupted tags. This results in the waste of bandwidth.

III. THE PROPOSED SCHEME

A. Outline

To generate the required MACs and D-MACs, our scheme consists of three steps. In step 1, it computes the l MACs (i.e., tags) for each message according to the keys which are chosen randomly from a set of secret keys, $\mathcal{K}_S$. Then, in step 2, our scheme computes a number of D-MACs (i.e., tags of tags), which should satisfy an orthogonality property of the initial MACs calculated in step 1, and it is calculated by using the properties of Null keys which was defined in the previous section. Finally, in step 3, the computed MACs and D-MACs are appended to the message. These three steps are depicted in Figure 2.

B. Construction

This scheme includes four steps: *Setup, Tag Generation, Verification* and *Encoding* detailed as follows.

1. *Setup:*
   a. Key Distribution Centre (KDC) distributes the following set of secret keys (i.e., key vectors) to the source node S:

$$\mathcal{K}_S = \{\mathbf{k}_1,\mathbf{k}_2,\cdots,\mathbf{k}_l\}, \mathbf{k}_i \in \mathbb{F}_p^{m+n+1} \qquad (3)$$

   b. KDC also distributes the following subsets of secret vectors to all intermediate nodes and sink nodes:

$$\mathcal{K}_{n_i} = \{\mathbf{k}_1,\cdots,\mathbf{k}_R\},\ \mathbf{k}_r \in \mathbb{F}_p^{m+n+1} \qquad (4)$$

2. *Tag generation:* For every data packet $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$, of each generation consisting of m data packets, the source uses the equations *(5)* and (*6*) to generate the l MACs and l' D-MACs, respectively:

$$t_{\mathbf{u}_i,l} = -\frac{\sum_{j=1}^{m+n} \mathbf{u}_{i,j} \times k_{l,j}}{k_{l,m+n+1}} \qquad (5)$$

$$\begin{pmatrix} k_{1,1} & \cdots & k_{1,l} \\ \vdots & \ddots & \vdots \\ k_{l',1} & \cdots & k_{l',l} \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_l \end{pmatrix} + \begin{pmatrix} k_{1,(l+1)} & \cdots & k_{1,(l+l')} \\ \vdots & \ddots & \vdots \\ k_{l',(l+1)} & \cdots & k_{l,(l+l')} \end{pmatrix} \begin{pmatrix} t'_1 \\ \vdots \\ t'_{l'} \end{pmatrix} = \mathbf{o}_{(l'\times 1)} \qquad (6)$$

3. *Verification:* When a relay or sink node receives a coded packet $y \in \mathbb{F}_p^{m+n}$ with its tags, this node checks the correctness of packet y using the algorithm *Verify* via its pre-distributed keys $\mathcal{K}_{n_i}$. If the results of following equations are 0, the received coded packet y is correct and the output is 1; otherwise the output is 0 and then this packet is discarded.

$$\delta_r = \left(\sum_{j=1}^{m+n} y_j\, k_{r,j} \right) + t_{y,r}\, k_{r,m+n+1} = 0,\ \forall \mathbf{k}_r \in \{\mathbf{k}_1,\cdots,\mathbf{k}_R\} \qquad (7)$$

$$\delta'_r = \left(\sum_{j=1}^{l} t_{y,j}\, k_{r,j} \right) + \left(\sum_{j=L+1}^{l'} t'_{y,j}\, k_{r,j} \right) = 0,\ \forall \mathbf{k}_r \in \{\mathbf{k}_1,\cdots,\mathbf{k}_R\} \qquad (8)$$

4. *Encoding:* When an intermediate node receives h encoded packets $\mathbf{u}_i \in \mathbb{F}_p^{m+n}\ (1 \le i \le h)$, and they all are checked or considered to be correct, a forward coded packet along with new tags is generated using the *Combine*

$$\left(\left(\mathbf{u}_i, \mathrm{t}_{\mathbf{u}_i,1}, \cdots \mathrm{t}_{\mathbf{u}_i,L}, \mathrm{t}'_{\mathbf{u}_i,1}, \cdots, \mathrm{t}'_{\mathbf{u}_i,L'} \right)_{i=1}^{h}, (c_i)_{i=1}^{h} \right)$$

algorithm with locally randomly generated coefficients c_i.

IV. THE CORRECTNESS OF PROPOSED SCHEME

Our proposed scheme is correct if the *Verify* algorithm passed the verification by getting 1 for the output of the two following algorithms. Our first algorithm (i.e., Algorithm 1) provides resistant against pollution attack; and the second one (i.e., Algorithm 2) takes care of tag pollution attacks.

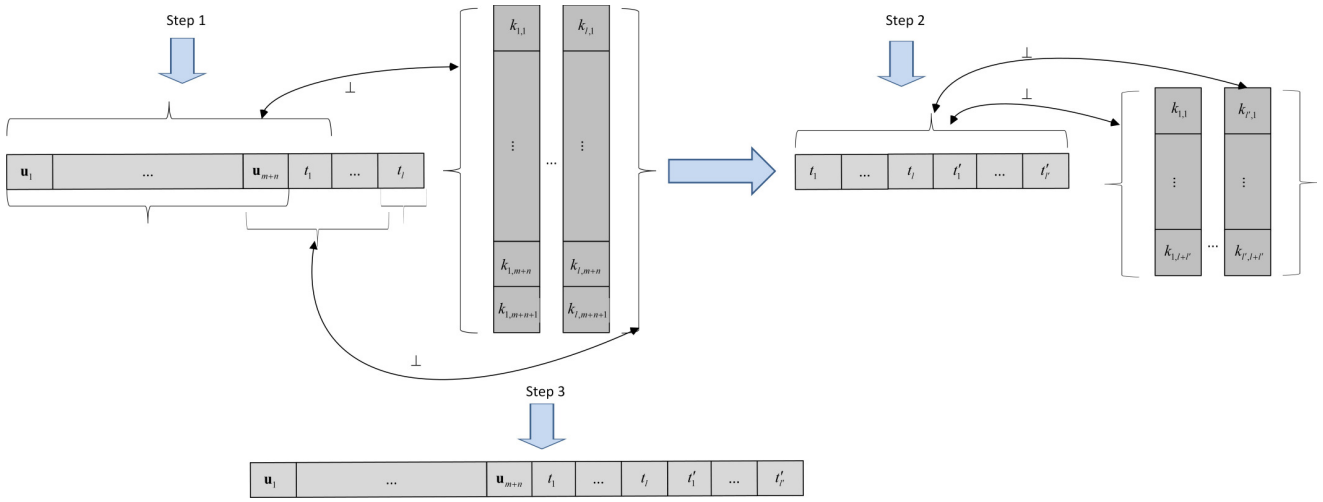

Figure 2. Our proposed scheme description. In the first step, we make l MACs (i.e., tags) by choosing l sets of keys from pool keys. Next, in step 2, we use the same keys to generate l' D-MACs (we use l' of l keys which is used for the first step). Finally, in step 3, we append these l MACS and l' D-MACs at the end of message packet.

The Algorithm 2 is run whenever the Algorithm 1 is given 1 as the result; otherwise the native packet is discarded.

Algorithm 1.

1. *Begin*
2. *Count=0;*
3. *For all the keys which each node has*
   If $Verify\left(\mathcal{K}_{n_i},\ (\mathbf{u}_i, MAC(\mathcal{K}_S,\mathbf{u}_i))\right)=1$
   then count=count++
   else continue
4. *If count>0*
   Then Call Algorithm 2;//it provides pollution attack resistance
   Else Discard $\mathbf{u}_i$
5. *End*

Theorem 1: The Algorithm 1 is correct.
Proof: According to the description of "*Setup*" and the PKD method [24] that we use, there is a key $\mathbf{k}_l$=$\mathbf{k}_r$ for l=r. As a result, by letting $y=\mathbf{u}_i$, $\mathbf{k}_l=\mathbf{k}_r$, and calculating $t_{y,r}$ from Equation (*5*), then for sure, according to Equation (*7*), we have that: $\delta_r=0, \forall \mathbf{k}_r \in \mathcal{K}_{n_i}$.
As a result, the Algorithm 1 is correct:
$Verify\left(\mathcal{K}_{n_i},\ (\mathbf{u}_i, MAC(\mathcal{K}_S,\mathbf{u}_i))\right)=1$ ■

Theorem 2: The Algorithm 2 is correct.
Proof: According to Equation (*6*), the D-MACs are created in a way that $\mathbf{k}_l$ is orthogonal to the concatenation of $t_{\mathbf{u}_i,j}$ and $t'_{\mathbf{u}_i,\mathrm{r}}$ where $(j=1,...,l)$, $(r=1,...,l')$. It can be represented as: $k_{\mathbf{u}_i,i} \perp (t_{\mathbf{u}_i,j} \,||\, t'_{\mathbf{u}_i,\mathrm{r}})$. This relationship is true for all packets in the same generation.

We assume that an adversary modifies d tags (i.e., t_i or t'_j), so regarding the Equation (*6*), If we assigned only one key vector to each node, the possibility of getting the result 1 is equal to $\frac{1}{p^d}$ and it is very small[1]. However, according to the key distribution model, we assign more than one vector keys, and it means that modifying any tags would be impossible and it will be detected immediately. Thus, the probability of a packet with modified tags passing the verification at two nodes is small and the second Algorithm is also correct. ■

Algorithm 2.

1. *Begin*
2. *Count=0;*
3. *For all the keys which each node has*
   If
   $Verify\left(\mathcal{K}_{n_i},\ (MAC(\mathcal{K}_S,\mathbf{u}_i),\mathrm{D\text{-}MAC}(\mathcal{K}_S,\mathbf{u}_i))\right)=1$
   then count=count++
   else continue
4. *If count>0*
   then Accept $\mathbf{u}_i$
   Else Discard $\mathbf{u}_i$
5. *End*

V. DISCUSSION

As outline in Section II, we assume the source node is trustworthy and the process of key pre-distribution is secure. Hence, the secret key sets k_S, assigned to the source node are considered secure.

[1] If $\mathbb{F}_p$=$\mathbb{F}_{2^8}$ and $d=2$, this probability should be $\frac{1}{2^{16}}\approx 0.001\%$.

Table I. Computational Complexity. L and N are the total number of tags which is used in [18] and [25], respectively.

	At the source node	At each node (intermediate or sink node)
MacSig [18]	$(m+n+1)L+(m+L+1)$	$\frac{3}{2}\lvert p\rvert(m+L+1)+(m+n+1)L$
KEPTE [25]	$N(m+n)$	$N(m+n)$
Our Proposed Scheme	$l(m+n+1)+l'(l'+l)$	$l(m+n+1)+(l'+l)$

However, an adversary can wiretap all the data communication in a network and may compromise several relay or sink nodes. Hence, the adversary can get access to the received packets from the previous hops, the key information distributed to him by the KDC and the key information stored at the compromised nodes. We consider three types of attacks as the following:

A. *Data Pollution attack*

If an adversary makes any change in the native packet, it is detected immediately; however, there is a possibility to travel more hops if these hops don't have the key. This probability is negligible.

B. *Tag Pollution attack (i.e., MACs)*

By checking the result of Algorithm 2, if an adversary makes any change in any tags, it is detected immediately.

C. *Tag Pollution attack (i.e., D-MACs)*

Regarding the tags (i.e., D-MACs) which are based on Null space properties and calculated by source node, there is no possibility to alter these tags.

VI. PERFORMANCE EVALUATION

In this section we provide a communication and computational complexity comparison of our proposed scheme and the two related works which can resist against tag pollution attacks [18], [25].

A. *Communication Overhead*

In [18], the MacSig scheme generates L MACs and a signature. This scheme uses a variable number of MACs which is calculated by $L=\frac{1}{\delta-1}e(c+1)\ln\frac{1}{\varepsilon}$, where δ and ε are security parameters, and c is the number of compromised nodes. Their idea relies on calculating L MACs and a signature by a source node. Totally, $L+1$ tags are appended at the end of each native packet. An intermediate node or sink node should verify each received packet by using the verification algorithm. We set the number MACs equal to l where this value is less than the number of MACs which is used in [18] (i.e., L). Moreover, we consider the value of l' as the number of D-MACs, where this value can be less or equal to l. In other word, the total number of tags which be used in our proposed scheme is $l+l' \ll L$.

However, the idea of [25] is based on generating N tags and appending these tags to each native packet. According to their consideration, this value is somehow equal to L. To end this section, our proposed scheme provides a less communication overhead in comparing to the related works.

B. *Computational Complexity*

For providing L MACs and a signature at a source node, MacSig [18] needs to use $L(m+n+1)$ and $(m+L+1)$ multiplications, respectively. However, for verification phase, it needs to do multiplications and exponentiations, where according to their setup parameters; it should be equal to $\frac{3}{2}\lvert p\rvert(m+L+1)+(m+n+1)L$ multiplications totally.

However, KEPTE [25] needs to calculate $N(m+n)$ multiplications[2].

In our proposed scheme, we need to generate $l+l'$ tags. So, a source node needs to do $l(m+n+1)+l'(l'+l)$ multiplications. However, each node needs to verify a received packet by calculating $l(m+n+1)+(l'+l)$ multiplications. For more detail, see Table I.

VII. CONCLUSION

This paper studied the problem of pollution and tag pollution attacks in wireless networks based on the technique of network coding. Previous studies demonstrated that network coding can provide an achievable throughput and robustness compared to traditional store-and-forward transmission paradigm. In our proposed scheme, a lightweight encryption scheme on top of network coding, to further to provide resistance against data pollution and tag pollution attacks by using two types of tags (i.e., MACs and D-MACs) was presented. We showed that our proposed scheme is efficient in computation, and incurs less communication overhead and keys storage for encryptions/decryptions.

[2] Similarly to the MacSig scheme and considering to the time is needed for addition which is negligible in comparing to multiplication, we don't count the number of additions which are needed in all schemes.

Our future work includes extending the application of proposed scheme to other communication networks, e.g., vehicular ad hoc networks. Moreover, we may aim to simulate our proposed scheme in future.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Program [FP7/2007-2013] under grant agreement n° 285969 [CODELANCE], and the Fundação para a Ciência e Tecnologia (PTDC/EEATEL/119228/2010 - SMARTVISION) and the Fundação para a Ciência e Tecnologia and the ARTEMIS JU (ACCUS – ARTEMIS - 005 - 2012 / GA number 333020). 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," *SIGCOMM Comput. Commun. Rev.,* vol. 37, pp. 169-180, 2007.

[2] D. Petrovic, K. Ramchandran, and J. Rabaey, "Overcoming untuned radios in wireless networks with network coding," *Information Theory, IEEE Transactions on,* vol. 52, pp. 2649-2657, 2006.

[3] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale File Distribution," in *IEEE INFOCOM*, 2005.

[4] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 46, pp. 1204-1216, Jul 2000.

[5] S. Y. R. Li, R. W. Yeung, and C. Ning, "Linear network coding," *Information Theory, IEEE Transactions on,* vol. 49, pp. 371-381, 2003.

[6] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, S. Jun, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *Information Theory, IEEE Transactions on,* vol. 52, pp. 4413-4430, 2006.

[7] T. Ho, L. Ben, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine Modification Detection in Multicast Networks With Random Network Coding," *Information Theory, IEEE Transactions on,* vol. 54, pp. 2798-2803, 2008.

[8] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," *Information Theory, IEEE Transactions on,* vol. 54, pp. 2596-2603, 2008.

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

[10] C. Gkantsidis and P. R. Rodriguez, "Cooperative Security for Network Coding File Distribution," in *INFOCOM, 25th IEEE International Conference on Computer Communications*, 2006.

[11] Y. Zhen, W. Yawen, B. Ramkumar, and G. Yong, "An Efficient Signature-Based Scheme for Securing Network Coding Against Pollution Attacks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008.

[12] Z. Fang, T. Kalker, M. Medard, and K. J. Han, "Signatures for Content Distribution with Network Coding," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, 2007, pp. 556-560.

[13] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a Linear Subspace: Signature Schemes for Network Coding," presented at the Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09, CA, 2009.

[14] y. Zhen, W. Yawen, B. Ramkumar, and G. Yong, "An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks," in *INFOCOM 2009, IEEE*, 2009, pp. 406-414.

[15] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-Based Integrity for Network Coding," presented at the Proceedings of the 7th International Conference on Applied Cryptography and Network Security, Paris-Rocquencourt, France, 2009.

[16] L. Yaping, Y. Hongyi, C. Minghua, S. Jaggi, and A. Rosen, "RIPPLE Authentication for Network Coding," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1-9.

[17] E. Kehdi and L. Baochun, "Null Keys: Limiting Malicious Attacks Via Null Space Properties of Network Coding," in *INFOCOM 2009, IEEE*, 2009, pp. 1224-1232.

[18] Z. Peng, J. Yixin, L. Chuang, Y. Hongyi, A. Wasef, and S. Xuemin, "Padding for orthogonality: Efficient subspace authentication for network coding," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1026-1034.

[19] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," *International Journal of Information and Coding Theory,* vol. 1, pp. 3-14, 2009.

[20] E. Porat and E. Waisbard, "Efficient signature scheme for network coding," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, 2012, pp. 1987-1991.

[21] A. Perrig, D. Song, R. Canetti, J. Tygar, and B. Briscoe, "Timed efficient stream loss-tolerant authentication (TESLA): Multicast source authentication transform introduction," *Request For Comments,* vol. 4082, 2005.

[22] L. Anh and A. Markopoulou, "TESLA-Based Defense against Pollution Attacks in P2P Systems with Network Coding," in *Network Coding (NetCod), 2011 International Symposium on*, 2011, pp. 1-7.

[23] T. HO, M. Medard, R. Koetter, D. Karger, M. Effros, S. Jun, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory 52,* pp. 4413–4430, 2004.

[24] 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*, 1999, pp. 708-716 vol.2.

[25] W. Xiaohu, X. Yinlong, Y. Chau, and X. Liping, "A Tag Encoding Scheme against Pollution Attack to Linear Network Coding," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 25, pp. 33-42, 2014.