On the Performance Analysis of IDLP and SpaceMac for Network Coding-enabled Mobile Small Cells

Reza Parsamehr, *Member, IEEE*, Georgios Mantas, *Member, IEEE*, Jonathan Rodriguez, *Senior Member, IEEE*, and José-Fernán Martínez-Ortega, *Senior Member, IEEE*

***Abstract*—Network coding (NC)-enabled mobile small cells are observed as a promising technology for 5G networks in a cost-effective and energy-efficient manner. The NC-enabled environment suffers from pollution attacks where malicious intermediate nodes manipulate packets in transition. Detecting the polluted packets as well as identifying the exact location of malicious users are equally important tasks for these networks. SpaceMac [1] is one of the most competitive mechanisms in the literature for detecting pollution attacks and identifying the exact location of attackers in RLNC. In this paper, we compare SpaceMac with the IDLP mechanism presented in [2]. Both mechanisms have been implemented in KODO and they are compared in terms of computational complexity, computational overhead, communication overhead and decoding probability. The performance evaluation results demonstrated that IDLP is more efficient than SpaceMac while at the same time is more secure as shown through the security analysis part in this paper.**

***Index Terms*—Network coding, pollution attacks, intrusion detection, location-aware prevention, 5G.**

I. Introduction

Network coding (NC) is a promising solution for increasing the throughput and improving the performance of the wireless network in Mobile Small Cells (MSC). However, despite the outstanding benefits of NC technology, this technology is susceptible to pollution attacks which are one of the most severe security threats for NC-enable MSC [3]. Thus, detection of the pollution attacks and the exact location of attackers are important for intrusion detection and location-aware mechanisms in NC-enabled MSC. Although there are many schemes against pollution attacks [4]–[10], there are only few focusing on identifying the location of malicious users [1], [2], [11]–[13].

This work is partly supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant H2020-MSCA-ITN-2016-SECRET-722424.

R. Parsamehr is with the Instituto de Telecomunicações, 3810-193 Aveiro, Portugal, and also with the Departamento de Ingeniería Telemática y Electrónica, Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, Universidad Politécnica de Madrid, 28031 Madrid, Spain (e-mail: parsamehr.r@av.it.pt).

G. Mantas is with the Instituto de Telecomunicações, 3810-193 Aveiro, Portugal, and also with the Faculty of Engineering and Science, University of Greenwich, London SE10 9LS, U.K. (e-mail: gimantas@av.it.pt).

J. Rodriguez is with the Instituto de Telecomunicações, 3810-193 Aveiro, Portugal, and also with the Mobile and Satellite Communications Research Group, School of Engineering, University of South Wales, Pontypridd CF37 1DL, U.K. (e-mail: jonathan@av.it.pt).

J.-F. Martínez-Ortega is with the Departamento de Ingeniería Telemática y Electrónica, Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, Universidad Politécnica de Madrid, 28031 Madrid, Spain (e-mail: jf.martinez@upm.es).

SpaceMac mechanism, presented by Le et al. in [1], is one of the most competitive mechanisms in the literature for detecting pollution attacks and identifying the exact location of attackers in RLNC. In this paper, we compare the IDLP mechanism presented by Parsamehr et al. in [2] with SpaceMac in terms of computational complexity, computational overhead, communication overhead and decoding probability. In order to evaluate and compare IDLP with SpaceMac, we implemented both mechanisms on KODO. The performance evaluation and the security analysis show that IDLP is more efficient and secure than SpaceMac.

The rest of this paper is organized as follows. Section II outlines the IDLP mechanism [2] and SpaceMace mechanism [1]. In section III, the two main security weaknesses of SpaceMac are presented. In Section IV, the performance evaluation and comparison of IDLP and SpaceMac are given. Finally, Section V concludes this paper.

II. Related Work

In the following section, "IDLP: An Efficient Intrusion Detection and Location-aware Prevention Mechanism for Network Coding-enabled Mobile Small Cells" [2], and "Cooperative Defense Against Pollution Attacks in Network Coding Using SpaceMac" [1] will be briefly summarized.

A. IDLP: An Efficient Intrusion Detection and Location-aware Prevention Mechanism for NC-enabled Mobile Small Cells

The IDLP involves a locating scheme and detection scheme both being founded on the basis of null space homomorphic MAC scheme [5] and they are described in following sections. This mechanism is divided into two phases for improving its effectiveness with regard to the consumption of resources;

- ***Phase 1: identifying the MSC where pollution attack happend.*** In the first step, all Relay Destination Nodes (DNs) and Nodes (RNs) are given a detection scheme belonging to IDLP mechanism. When a pollution attack is detected by an RN or DN, they drop the polluted packet, and send a report to the Hotspots of the Mobile Small Cells (MSCs) where the reporter belongs to and the Hotspot will forward the report to the SDN Controller, being in charge of MSC identification where in the location which a pollution attack ensued according to the reports that were received.
- ***Phase 2: Identifying the location of adversary node in the MSC that has been polluted.*** All mobile devices

in the spotted MSC that has been polluted in phase1 are given the locating and detection scheme. Once a mobile device in MSC that has been polluted detects any pollution, the mobile device will drop the polluted packet and will send a report according to locating scheme towards the Hotspot. Then the report will be sent to the SDN controller by the Hotspot to select the best suitable preventive measure (e.g., preventing adversary mobile device(s) from gaining access to the network). Otherwise, an expanded coded packet will be created which is established on the basis of the received coded packet and the key that is shared between them and SDN controller. They will send this expanded coded packet to the next node and Hotspot as well.

1) Detection Scheme: According to [7] and [5], in the detection scheme of IDLP, the message is separated into a generation of native packet represented as $\underline{\mathbf{b}}_1, \underline{\mathbf{b}}_2, ..., \underline{\mathbf{b}}_m$ by the SN, in which m is the size of the generation and each of the packets $\underline{\mathbf{b}}_i$ is composed of n symbols (i.e., $\underline{b}_{i,1}, \underline{b}_{i,2}, ..., \underline{b}_{i,n}$) which are located in the finite field $\mathbb{F}_p^n$. Therefore, a coded packet $\mathbf{b}_i$ will be created by the source node based on (1) and will direct it to the next mobile devices.

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

For simplicity, (1) can also be written as follows:

$$\mathbf{b}_i = (b_{i,1}, \cdots, b_{i,m+n}) \in \mathbb{F}_p^{m+n} \quad (2)$$

As shown in (3), each intermediate node generates a fresh coded packet x being a linear combination of h received coded packets ($\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_h$) and refers it to its neighbors. β_i is coding coefficient which is chosen randomly from $\mathbb{F}_p$ and all arithmetic operations are performed within the finite field $\mathbb{F}_p$.

$$x = \sum_{i=1}^{h} \beta_i \mathbf{b}_i \quad (3)$$

The L tags are created according to null space properties [14] by the SN, in order to detect pollution attacks. The five steps below are applied in order to create the tags in addition to verification of the orthogonality of the received coded packets:

1) Distribution of the keys to the source node: A set of keys ($\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_L$) are send out by key distribution center in the finite field $\mathbb{F}_p^{m+n+L}$ and which are distributed in the source node.
2) The L tags (i.e., $t_1, t_2, ..., t_L$) are created by L keys for each coded packet by the SN, as stated by (4). Each coded packet is comprises $m+n$ symbols, as well as L created tags (i.e., t_{SN}).

$$\begin{bmatrix} \mathcal{C}_{1,1} & \ldots & \mathcal{C}_{1,m+n} \\ \vdots & \vdots & \vdots \\ \mathcal{C}_{L,1} & \ldots & \mathcal{C}_{L,m+n} \end{bmatrix}_{L*(m+n)} * \begin{bmatrix} \mathbf{b}_{i,1} \\ \mathbf{b}_{i,2} \\ \vdots \\ \mathbf{b}_{i,m+n} \end{bmatrix}_{(m+n)*1} + \begin{bmatrix} \mathcal{C}_{1,m+n+1} & \ldots & \mathcal{C}_{1,m+n+L} \\ \vdots & \vdots & \vdots \\ \mathcal{C}_{L,m+n+1} & \ldots & \mathcal{C}_{L,m+n+L} \end{bmatrix}_{L*L} * \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_L \end{bmatrix}_{L*1} = 0 \quad (4)$$

3) With the aim of avoiding tag pollution attacks, the L tags are swapped on the basis of the secret key (SV) that is shared between the DNs and SNs, as stated by (5).

$$\overline{\mathbf{b}}_i = Swap(\mathbf{b}_i)_{SV} \quad (5)$$

4) A set of new keys are prodused by KDC center through the use of the swapping vector SV and according to the key set that was sent towards the SN during step 1 as stated in (6). Next, the keys are forwarded to the DNs the intermediate nodes for verifying the received coded packets.

$$\mathcal{C}'_i = Swap(\mathcal{C}_i)_{SV} \quad (6)$$

5) Lastly, the received coded packet is verified by each DN and intermediate node according to following equation:

$$\delta = Swap(\mathcal{C}_i)_{SV} * Swap(\mathbf{b}_i)_{SV} = \sum_{j=1}^{m+n+L} \mathcal{C}'_{i,j} * \overline{\mathbf{b}}_{i,j} \quad (7)$$

If $\delta = 0$, then the received coded packet is confirmed and suitable for transmission to the next nodes. If not, it is dropped.

2) Locating Scheme: The adversary mobile node's precise location within the MSC that has been polluted is recognized by locating scheme. In this step, each of the mobile nodes is accountable for: a) generating a coded packet which is expanded and on the basis of the received coded packet, in addition to sending it to the Hotspot and next node, then b) reporting Hotspot as soon as a polluted packet is detected by the detection scheme within polluted MSC. Both the report and the expanded coded packet are forwarded to the SDN Controller being in charge of recognizing the adversary mobile nide's precise location.

- ***Expanded Coded Packet:*** An additional tag is added to each coded packet by each of the intermediate node with the aim of verifying itself to the SDN Controller. This tag is calculated on the basis of the pre-distributed share key between the SDN Controller and each node. This tag is calculated based on the following Equation:

$$\begin{bmatrix} \mathcal{C}''_{1,1} & \ldots & \mathcal{C}''_{1,m+n} & \ldots & \mathcal{C}''_{1,m+n+L} \end{bmatrix}_{1*(m+n+L)} * \begin{bmatrix} b_{i,1} \\ b_{i,2} \\ \vdots \\ b_{i,m+n} \\ t_1 \\ \vdots \\ t_L \end{bmatrix}_{(m+n+L)*1} + \mathcal{C}''_{1,m+n+L+1} * \mathbf{s}_i = 0 \quad (8)$$

where $\begin{bmatrix} \mathcal{C}''_{1,1} & \ldots & \mathcal{C}''_{1,m+n} \cdots & \mathcal{C}''_{1,m+n+L+1} \end{bmatrix}_{1*(m+n+L+1)}$ is the pre-shared key sent out by the KDC, and s_i is the properly calculated tag.

The received expanded coded packet $\{\mathbf{b}_i \| t_{SN} \| s_i\}$ is confirmed by the SDN controller based on the following formula. Where $\mathbf{b}_i$ represents the coded packet, t_{SN} is the set of tags which appeded by SN, and s_i is the tag which appended by the given intermediate node. If $\delta = 0$

then the received expanded coded packet is confirmed.

$$\delta = \sum_{j=1}^{m+n+L+1} \mathcal{C}''_{i,j} * \overline{\{\mathbf{b}_{i,j} \parallel t_{SN} \parallel s_i\}} \quad (9)$$

- ***report***

  When a signed polluted packet (e) by the previous mobile device's key ($\{e \| s_{i-1}\}$) is detected, a report is generated by the intermediate node or a DN, who detects pollution. The generated report is the received polluted packet ($\{e \| s_{i-1}\}$) signed by the given node and is represented as $\{e \| s_{i-1} \| s_i\}$.

  In the following equation if $\delta = 0$, then the SDN controller will verify the sender.

$$\delta = \sum_{j=1}^{m+n+L+2} \mathcal{C}''_{i,j} * \overline{\{e \parallel s_{i-1} \parallel s_i\}} \quad (10)$$

  Therefore, the signature of the adversary node (s_{i-1}) is verified if $\delta = 0$.

$$\delta = \sum_{j=1}^{m+n+L+1} \mathcal{C}''_{i,j} * \overline{\{e \parallel s_{i-1}\}} \quad (11)$$

B. Cooperative Defense Against Pollution Attacks in Network Coding Using SpaceMac

In [1] the authors plan a cooperative defense system including both locating and detection schemes, with SpaceMac being their building block inspired from [4] and [11]. In this scenario, there is S which is a source node, some receivers R and some intermediate nodes denoted by I. A generation is composed of m packets, $\bar{v}_1, \bar{v}_2, ..., \bar{v}_m$ in finite field $\mathbb{F}_p^n$. Then the source node generates a coded packet v_i according to following equation and refers it to the intermediate nodes.

$$v_i = (-\bar{v}_1 -, \overbrace{\underbrace{0, \cdots, 0}_{i-1}, 1, 0, \cdots, 0}^{m}) \in \mathbb{F}_p^{m+n} \quad (12)$$

The packet y stand as a linear combination of the packets v_i's. Represent the subspace spanned by v_i's by $\Pi^S \subseteq \mathbb{F}_p^{n+m}$. Here, Π^S is denoted as the source space.

The HomMac scheme suggested in [4] is described with three algorithms: Mac, Combine, and Verify.

- A tag is produced by the Mac algorithm for a given vector getting a secret key k, the identifier id of the source space Π^S, as well as a vector $y \in \mathbb{F}_q^{m+n}$ as input and generates tag t for y.
- The Combine algorithm gets p vectors $y_1, \ldots, y_p$, their tags $t_1, \ldots, t_p$ under key k, and their coefficients $\alpha_1, \ldots, \alpha_p \in \mathbb{F}_q$ as inputs and calculates a tag t for a linear combination of some given vectors $y \stackrel{def}{=} \Sigma_{i=1}^{p} \alpha_i y_i$.
- The tag is confirmed by Verify algorithm through receiving a secret key k, the identifier id of the source space Π^S, a vector $y \in \mathbb{F}_q^{m+n}$, and its tag t as input and giving 0 (reject) or 1 (accept) as an output.

1) Detection Scheme: In the detection scheme, an intermediate node N receives some coded packet from its parents and sends out a recoded packet y to its children and it must be $y \in \Pi_N(t)$ or else, y is corrupted ($\Pi_N(t)$ represents the space covered by all the packets that node N receives from all its parents until time (t)). Hence, in the SpaceMac the parents of N are allowed to sign Π_N and N's children are allowed to verify if the received packet from N belongs to Π_N. This cooperation of N's parents and its children, makes it possible for children to detect any polluted packet sent by N.

There is a controller in this scheme. The controller knows the whole topology of the graph and each node N shares a pair of secret keys (k_N^1, k_N^2) with the controller which can be recognized with a public key infrastructure. The controller defines the key $k_{\bar{N}}$ for every intermediate node N. When using SpaceMac, the parents and children of N utilize this key and it is not known to N itself. Each node necessarily should know a dissimilar set of keys in order to contribute in the detection scheme according to its position in the network, since it can serve as either a parent or a child. Next, all the receivers and the source must share k^*, an end-to-end key. The key is used to guarantee detection in the company of colluding adversaries, where a node N colludes with its parent to gain $k_{\bar{N}}$ and therefore will be able to bypass its children's verification.

Firstly, an end-to-end tag $t_{v_i}^{K^*}$ is calculated by source S, through the use of SpaceMac's Mac algorithm with key k^* : $t_{v_i}^{K^*} = Mac(v_i, k^*)$. Source node attaches the created end to end tag to every source packet and sends $w_i \stackrel{def}{=} \{t_{v_i}^{K^*} \| v_i\}$ instead of v_i. The MAC tag, $t_y^{k_{\bar{N}}}$, is calculated by P by using Mac, under key $k_{\bar{N}}$: $t_y^{k_{\bar{N}}} = Mac(y, k_{\bar{N}})$ if P as a parent of N wants to send a packet y to its child N. In addition, the helper tag, P must also pass along a verification tag of y, which is used by N to confirm the integrity of y. The node uses $k_{\bar{P}}$ to verify the integrity of the packet, when a node N receives a packet $\{t_y^{k_{\bar{N}}} \| t_y^{k_{\bar{P}}} \| y\}$ from its parent P. As can be seen in the Verify algorithm, if Verify $(k_{\bar{P}}, y, t_y^{k_{\bar{P}}}) = 1$ the packet is assumed to be non-corrupted. So, if N receives a corrupted packet from P, the attack is instantly detected by N. If N is a receiver, then it will additionaly verifies the end-to-end tag using key k^*.

2) Locating Scheme: Each intermediate node cooperates with the controller through this scheme, by signing the space that is spanned by the packets which it refers to the next node using SpaceMac. Therefore, when the attacker as the next node gives a report a fake space to the controller, it will not have the appropriate signature of the fake space to convince the controller.

According to locating scheme proposed in [14] and [20], node N reports a arbitrarily chosen packet, y_r, of the space Π_N^P which is a received subspace from a parent P. The controller though checking whether $y_r \in \Pi_S$ can recognize if the edges are polluted or not (e.g., $\Pi_N^P \subseteq \Pi_S$).

Based on Mac algorithm in SpaceMac, P generates a tag t_{y_i} with a sectert key shared by controller, when it wants needs to send packet y_i to N. When N reports y_r to the controller, the generated tag using the Combine algorithm for y_r is valid on the condition y_r is a linear combination of packets that it received from P. If y_r is not a linear combination of y_i's then N can forge a valid tag for y_r with only an insignificant probability of $\frac{1}{q}$.

They utilize a non-repudiation transmission protocol proposed by Wang et al. [12] with the aim of not allowing a malicious nodes to send invalid tags to their children to stop the children from giving a report about polluted space. This mechanism can accurately locate the attacker after the

TABLE I
COMPUTATIONAL COMPLEXITY

Scheme	No Attack	One attacker			Two or more attacker in a row		
	Tag creation complexity in each intermediate node	Attack Detection	Identify the location of attacker	Tag creation complexity in each intermediate node	Attack Detection	Identify the location of attacker	Tag creation complexity in each intermediate node
IDLP	0	in the next node	Yes	O(c)	in the next node	Yes	O(c)
SpaceMac	$O(f^2)$	in the next node	Yes	$O(f^2)$	in the destination node	No	$O(f^2)$

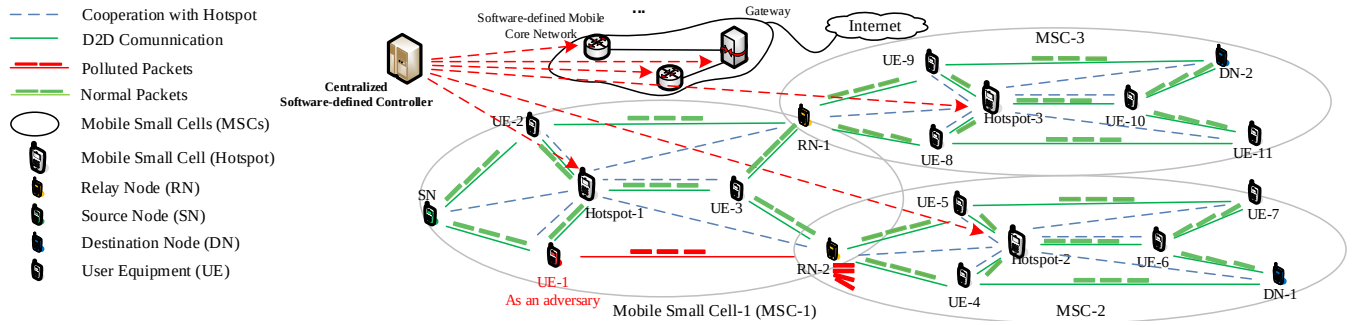

Fig. 1. Scenario Architecture

controller gathers the true subspaces from all nodes, and stops nodes from lying.

III. SECURITY ANALYSIS

The two main security weaknesses of SpaceMac, compared to IDLP, are presented in this section. We consider the following two scenarios:

- ***Scenario A – One attacker:*** Both mechanisms detect the polluted packets in each intermediate node and identify the exact location of the compromised node. However, in SpaceMac, if an external attacker injects a polluted packet to the nodes in the first level of intermediate nodes (i.e., the children of the source node), the intermediate node could not detect it because there is no verification tag for intermediate nodes in the first level. This pollution is detectable in the next level. In the contrary, IDLP detects all the polluted packets in the first level.
- ***Scenario B – Two or more attackers in a row:*** In SpaceMac, when there are two or more attackers in a row and node N colludes with its parent to obtain $k_{\tilde{N}}$, the polluted packet can bypass the verification at the children of node N. However, there is the key, k^*, to use for ensuring pollution detection in destination nodes, but SpaceMac cannot identify the exact location of the attackers in a row and therefore, the pollution distribute through the network. Thus, the destination nodes receive the polluted packets and drop them. While IDLP does not only detect and drop the polluted packet but also identifies the exact location of the attackers in a row.

IV. PERFORMANCE EVALUATION

In this section, the IDLP and the SpaceMac mechanisms are compared. First of all, as shown in Fig. 1, three butterfly topologies, including 18 normal nodes and one adversary node are implemented. The adversary node is considered at a fixed position, the probability that the adversary node pollutes a relayed packet is 1 and the pollution scheme is continuous. The implementation is based on RLNC approach of the Network Coding library of KODO [15], [16]. In this implementation, the packets, the required keys, and their proper tags at the intermediate and nodes source node are generated by Matlab. Then, the generated packets, keys and tags are included manually in KODO so as to achieve the desired functionality of the implemented scenario on KODO.

The packet generation size is 64 symbols and the symbol size is set between $1,000$ to $10,000$ bytes. For IDLP, the value of L (i.e., number of tags) can be 27, 42, or 54 [14]. However, for SpaceMac the value of L is always 1. The Galois field in use is $GF(2^8)$. In the whole implementation, the machine used for running has the following characteristics: a 2.7GHz Core i7 CPU with 8GB of physical memory.

In this section, the performance evaluation and the comparison of the proposed IDLP, and SpaceMac are provided in terms of computational complexity and overhead, communication overhead, and decoding probability when there is one attacker.

1) Computational Overhead: The total time elapsed from the time that the packet is generated to the time that the destination nodes verify and decode the received packet, and it is given by the following equation:

$$T_{total} = T_{enc} + T_{rec} + T_{dec} + T_{ver} \quad (13)$$

In this equation, T_{enc} is the encoding time at the source node, T_{rec} is the recoding time at each intermediate node, T_{dec} is the decoding time at the destination node, and T_{ver} is the verifying time at the intermediate and destination nodes.

The T_{total}, for IDLP and SpaceMac are illustrated in Fig. 2. As shown, by increasing the number of tags in IDLP, the T_{total} increases as well. However, the T_{total} of IDLP is still less than the T_{total} of SpaceMac because in IDLP the detection and locating schemes are not applied to all mobile devices in order to protect the NC-enabled mobile small cells from the depletion of their resources.

Additionally, regarding the verification time (see Fig. 3), as mentioned before in SpaceMac, in each round, each parent P of N creates a new tag for each child of N and appends it to the coded packet to facilitate the detection of any pollution that might be created by node N. These operations are repeated for all the intermediate nodes. Therefore, if each parent P has

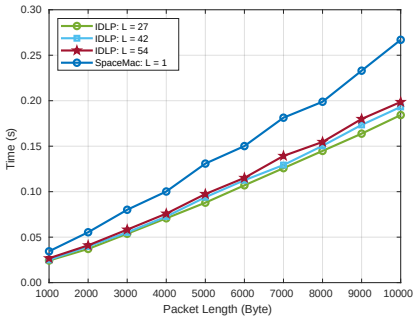

Fig. 2. The T_{total} for different number of tags in IDLP and SpaceMac

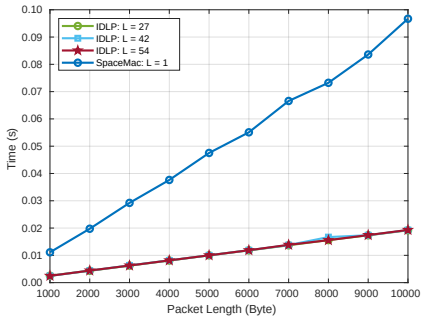

Fig. 3. The T_{ver} for different number of tags in IDLP and SpaceMac

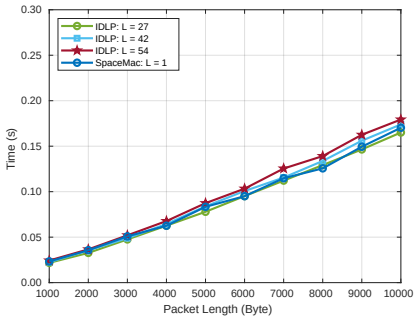

Fig. 4. The T_{comm} for different number of tags in IDLP and SpaceMac

f children in average (e.g., $N_1, \ldots, N_f$), node P must create f^2 tags for each coded packets. In the case of an extensive network that has h intermediate nodes, the complexity of SpaceMac reaches to $O(h \times f^2)$. It is worthwhile to mention that these operations are happening even if there is no attack in the network. While in IDLP, if there is no attack in the network, IDLP does not create any extra tag in intermediate nodes. As it is described in section II, if the RNs or DNs detect any pollution, IDLP creates an extra tag in each intermediate node. This tag is only created within the area of a polluted MSC, which is already detected by the SDN controller. Since it is assumed that the number of mobile devices in each MSC is constant, the complexity of IDLP is $O(c)$ (See table I).

2) communicational Overhead: The communication overhead (T_{comm}) of IDLP and SpaceMac is defined as follows:

$$T_{comm} = T_{total} - T_{ver} \tag{14}$$

Fig. 4 shows that the T_{comm} of SpaceMac and IDLP for 27 tags is almost the same. However, the T_{comm} of IDLP for 42 and 54 tags is more than the T_{comm} of SpaceMac. The difference is due to the fact that the length of the coded packets, sent through the network based on IDLP, is larger than the length of the coded packets sent through the network based on the SpaceMac.

3) Decoding Probability: The P_r is defined as the probability that a corrupted packet is not detected in the verification phase. According to our implementation results, when there is one attacker the P_r is almost 0 for both IDLP and SpaceMac. This is because when there is one attacker and IDLP or SpaceMac are applied, the adversary does not have any chance to distribute the corrupted packet in the network due to the fact that the detected adversaries are blocked from access to the network.

However, when there are two or more adversaries in a row, the P_r of IDLP is still 0, since IDLP blocks the attackers after detection, but the P_r of SpaceMac is 1 because SpaceMac cannot identify the exact location of the attackers and block them even after detection at the destination nodes.

V. CONCLUSION

In this paper, we compared IDLP with SpaceMac which is one of the most competitive mechanisms in the literature for detecting pollution attacks and identifying the exact location of attackers in RLNC. Both mechanisms have been implemented in KODO and they are compared in terms of computational complexity, computational overhead, communication overhead and decoding probability. The performance evaluation results demonstrated that IDLP is more efficient than SpaceMac while at the same time is more secure as shown through the security analysis in Section III.

REFERENCES

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

[2] R. Parsamehr, G. Mantas, J. Rodriguez, and J. Martínez-Ortega, "Idlp: An efficient intrusion detection and location-aware prevention mechanism for network coding-enabled mobile small cells," *IEEE Access*, vol. 8, pp. 43 863–43 875, 2020.

[3] R. Parsamehr, G. Mantas, A. Radwan, J. Rodriguez, and J.-F. Martínez, "Security threats in network coding-enabled mobile small cells," in *International Conference on Broadband Communications, Networks and Systems*. Springer, 2018, pp. 337–346.

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

[5] A. Esfahani, G. Mantas, and J. Rodriguez, "An efficient null space-based homomorphic mac scheme against tag pollution attacks in rlnc," *IEEE Communications Letters*, vol. 20, no. 5, pp. 918–921, 2016.

[6] A. Fiandrotti, R. Gaeta, and M. Grangetto, "Securing network coding architectures against pollution attacks with band codes," *IEEE Transactions on information forensics and security*, vol. 14, no. 3, pp. 730–742, 2018.

[7] R. Parsamehr, A. Esfahani, G. Mantas, A. Radwan, S. Mumtaz, J. Rodriguez, and J.-F. Martíez-Ortega, "A novel intrusion detection and prevention scheme for network coding-enabled mobile small cells," *IEEE Transactions on Computational Social Systems*, 2019.

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

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

[10] M. Kim, M. Médard, and J. Barros, "Algebraic watchdog: mitigating misbehavior in wireless network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 10, pp. 1916–1925, 2011.

[11] M. J. Siavoshani, C. Fragouli, and S. Diggavi, "On locating byzantine attackers," in *2008 Fourth Workshop on Network Coding, Theory and Applications*. IEEE, 2008, pp. 1–6.

[12] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, "Identifying malicious nodes in network-coding-based peer-to-peer streaming networks," Tech. Rep., 2009.

[13] R. Parsamehr, A. Esfahani, G. Mantas, J. Rodriguez, and J.-F. Martínez-Ortega, "A location-aware idps scheme for network coding-enabled mobile small cells," in *2019 IEEE 2nd 5G World Forum (5GWF)*. IEEE, pp. 91–96.

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

[15] M. V. Pedersen, J. Heide, and F. H. Fitzek, "Kodo: An open and research oriented network coding library," in *International Conference on Research in Networking*. Springer, 2011, pp. 145–152.

[16] P. Pahlevani, H. Khamfroush, D. E. Lucani, M. V. Pedersen, and F. H. Fitzek, "Network coding for hop-by-hop communication enhancement in multi-hop networks," *Computer Networks*, vol. 105, pp. 138–149, 2016.