# On blockchain enhanced secure network coding for 5G deployments

Vipindev Adat, Ilias Politis, Christos Tselios, Panagiotis Galiotos, Stavros Kotsopoulos
Wireless Communications Laboratory
University of Patras, Greece
{vipindev, ipolitis, tselios, pgaliot, kotsop}@ece.upatras.gr

***Abstract*—The fifth generation of mobile networks is rapidly evolving, fueled by the ever-growing mobile traffic over recent years. Ultra-reliable and low latency provisioning of services with ranging contextual parameters mapped to highly fluctuating levels of quality of service and quality of user experience, render the optimum operation point of the future networks unsustainable. Network coding is emerging once more as a potential key enabler for optimizing bandwidth requirements and energy consumption in highly dense mobile network environments. Nonetheless, network coding deployments still need to consider security vulnerabilities and their countermeasures, before they can be adapted as part of the emerging mobile network deployments. This paper studies the performance of random linear network coding for wireless mobile networks with an emphasis on pollution attacks while proposing a novel blockchain based message authentication scheme in order to minimize delays and signalling costs.**

***Index Terms*—network coding; RLNC; pollution attacks; 5G small cells**

## I. Introduction

Emerging fifth generation (5G) networks and services are being characterized by strict requirements regarding throughput gain and consumed energy. This evolution of wireless mobile networking towards higher data rates and capacity, ultra low latency and increased resilience is fueled by the rapid penetration of advanced smart mobile applications and their diverse nature in terms of quality of experience, security and ubiquity. Moreover, the foreseen increase of the connected devices in the near future and the resulted data volumes stretches the capabilities of current network deployments to satisfy the future network requirements [1], [2]. The 5G paradigm as it emerges in recent studies and early pilots [3], [4], adopts a number of technological solutions [5] to form the building blocks of the next generation wireless mobile network architecture and address all these challenges. Specifically, the future networking environment will be characterized by highly dense heterogeneous cells. This heterogeneity is extended not only to the diverse radio access technologies that 5G networks will incorporate (i.e., 3G, 4G, 5G, etc.) but also to the different type of cells that future wireless networks will consider, including macro, pico and small cells [6]. Cloud is already anticipated to be the cornerstone of 5G deployments and as such Cloud-RAN is considered as a key enabler for efficient base band processing in the cloud [7]. As the future networking environment begins to materialize, it seems that the concept of cooperated small cells may provide a basis for the mobile cell architecture. Towards this end, Network Coding by incorporating either linear or non-linear coding schemes, is being considered as a key enabler for achieving the requirements for increased throughput and high resilience for wireless mobile networks [8].

The future networking environment will also be characterized by flows of confidential information being downloaded, uploaded and processed via trusted or untrusted mobile small cell networks. Hence, security is rapidly emerging as a paramount concern for both mobile operators and users. In an era where cyber crime is an occurring theme in the internet highways, national authorities and telecommunication stakeholders have invested heavily into preventive measures that include service denials, intrusion attacks, and false identity among others. Therefore, the 5G paradigm needs also to address a number of energy efficient secure network coding solutions. The problem of secure network coding was first studied in [9], where a scheme which converts any linear network codes to a form that an eavesdropper, who monitors a bounded number of network links is unable to obtain any information about the transmitted messages, was proposed. Furthermore, authors in [10] proposed distributed polynomial-time rate-optimal network codes that are resistant against Byzantine adversaries. Moreover, Microsoft launched a file swarming application with network coding called "Microsoft Secure Content Downloader" [11]. Network coding can further be used to protect wireless video streams by increasing the overall robustness to losses and failures and protect transmission from adversaries, as previously proposed in [12].

The aim of this paper is to study the performance of secure random linear network coding architecture against pollution attacks, utilizing homomorphic message authentication codes and signatures. The study goes one step further by introducing the concept of block chains as a novel solution for efficient key distribution mechanism. A comparison with well studied network coding schemes MacSig [13] and Dual HMAC [14] indicate that the proposed block chain enhanced secure network coding scheme reduces the computational complexity and the communication overhead significantly.

The rest of the paper is organized as follows. Section II introduces the network coding architecture considered in this

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 722424

study, while Section III discusses the security aspects, focusing on pollution attacks and blockchain-based key management. The performance analysis of the proposed scheme is described in Section IV and Section V summarizes the results and concludes the paper.

## II. Network Coding envision for 5G deployment

Network coding is a technology constantly relevant for wireless mobile network implementations, due to its ability to optimize the harness of stringent bandwidth resources and increase the resilience of such randomly fluctuating networks. Network coding is an excellent fit for multi-hop mobile wireless networks, since it ensures error-free links, hence increased throughput, among the communicating nodes. Special focus has been given lately on random linear network coding (RLNC) [15]. RLNC has emerged as a highly significant protocol for many applications, especially due to its independent and simple encoding mechanism, where distributed and independent nodes comprise the networks such as wireless networks, sensor networks, mobile ad-hoc networks, and peer to peer networks. The proposed secure network coding implementation is based on RLNC and its basic principles of operation are introduced in this section.

### A. Random Linear Network Coding

In the RLNC approach, the complete message is considered as a sequence of generations where each generation has a number of native packets within it. The source node encodes these original packets with locally generated random coefficients and transmits them. When an intermediate node receives these incoming packets, it re-encodes them with its locally generated random coefficients and forward the re-encoded packets. Finally, the destination node will decode the original packets of a generation from the incoming encoded packets, if it receives enough number of encoded packets from the same generation. Figure 1 showcases the RLNC-based butterfly network architecture that is described, analyzed and evaluated throughout this study.

In a practical multicast scenario where a source wants to send some message to a number of destinations, this RLNC based approach can be used as follows. The original message is divided into packets which will be coded. Each packet $P_i$ can be considered as a vector of $n$ elements such as, $P_{i1}, P_{i2}, \ldots P_{in}$ defined over the finite field $F_q^n$ where $q$ defines the finite field size. Chou et al. demonstrate in [16] that $q = 2^8$ can be considered as practically sufficient and convenient for applications. Further, instead of sending the message simply as packets, it is grouped into generations to reduce the inevitable overhead of coding coefficients. Each generation will consist of $m$ packets so that the coding coefficient for each packet will be a vector of dimension $m$. Thus a single native generation will be a matrix of size $m \times n$ and the augmented generation will be a matrix of $m \times (m+n)$. The initial construction at source is facilitated by augmenting the $i^{th}$ unit vector of dimension $m$ to the original packet $P_i$. Then, it will be multiplied by a random coefficient matrix of

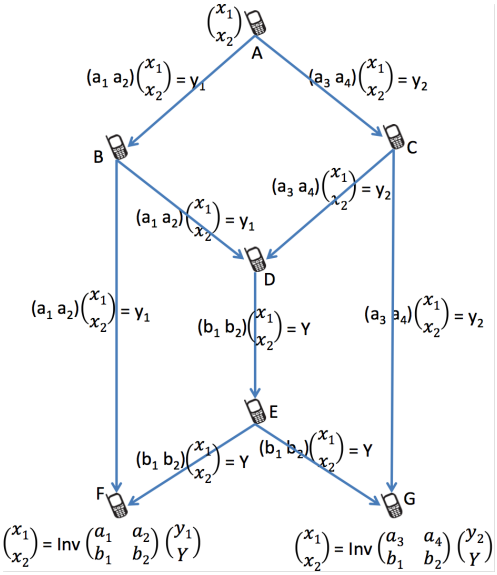


Fig. 1: RLNC based butterfly network

$m$ vectors where each vector will be an element of $F_q^n$ prior to sending.

## III. Secure Network Coding

Despite the tremendous advantages of RLNC in terms of bandwidth, energy consumption and robustness to packet losses, a wide spectrum of security attacks should be addressed before network coding reaches its full potential in real-world communication systems. Particularly, in a dense heterogeneous small cells based on network coding implementation, pollution attacks, which result into severe network performance degradation and energy waste, needs to be addressed effectively and efficiently. To this end, appropriate security schemes are required, that provide resistance against pollution attacks. Furthermore, the appropriate key management schemes to support the proposed security schemes are essential. A use case study that follows, firstly aims to address the vulnerabilities of network coding to pollution attacks, where malicious nodes are injecting infected packets/blocks in the mobile small cell network. Such attacks threaten the correct reconstruction of the original information and deteriorate the overall network performance. The proposed approach utilizes homomorphic hash function and homomorphic signatures in order to secure the network coded mobile small cells and enable intermediate nodes to verify the signature (or the hash digest) of each block and discard the polluted ones with the help of a blockchain, a distributed ledger of immutable encrypted values [17]. Secondly, the studied use case aims to secure network coded small cells by building a block chain based tag sharing between the communicating nodes.

### A. Security system architecture

This section discusses security aspects that the proposed scheme explores, with focus on data pollution and tag pollution

attacks. As in Figure 1 all network nodes are categorized as follows:

(i) *Source nodes ($S$):* source nodes generate the message and are considered trustworthy and non-compromised in all cases. Whenever a source node generates a message, it is split into generations to facilitate the RLNC-based multicast. Each generation will have $m$ number of packets. This native packet is then augmented with a coefficient vector to create the coded packet at the source. The source then transmits a generation of such coded packets at each transmission opportunity. To facilitate our security scheme, it also computes a number of tags using the pre-distributed symmetric keys over each packet and attaches it to the packet. Further, the tags in a particular generation are signed with a private key by the source and added as an entry into a blockchain. Further details regarding the security scheme are explained in the following subsection.

(ii) *Intermediate nodes ($I$):* Intermediate nodes are capable of re-encoding the packets. Further, they are considered as the vulnerable points in the network. Whenever the intermediate node receives a packet, it will be added to a buffer identified and sorted by the generation number. Once it has enough number of innovative packets in the buffer for a generation, it will re-encode the packets and transmit them at the next opportunity. The tags are verified as per the security scheme to ensure the authenticity of the messages. However, in case of a compromised node, the intermediate node can show anomaly behaviour. This will be further explained in the adversary model.

(iii) *Destination nodes ($D$):* The destination nodes activity is very much similar to an intermediate node. It follows the same procedure as the intermediate node when it receives a packet and waits for enough number of packets to decode a generation. Once it has enough number of packets in the buffer, it decodes the packet and once all the generations are decoded, it has the complete message. The tags are verified as per the security scheme to ensure the authenticity of the messages. Throughout this paper, we will not address the scenario which includes a compromised destination since it wont have any significance other than the particular node wont be able to receive the message.

In order to facilitate the proposed scheme, a blockchain-like distributed ledger is deployed over the network. Each block in the chain consists of the tags created by the source for a particular generation of native packets. These entries to the chain will be signed by a private key owned by the source to verify the authenticity. Whenever a new generation of packets have to be encoded, the corresponding tags are calculated and added as an entry in the chain with the generation number. Whenever, an intermediate or receiver node has to verify a generation it received, it will fetch the corresponding tags stored in the blockchain with the particular generation number and repeat the verification process. By design, blockchain is

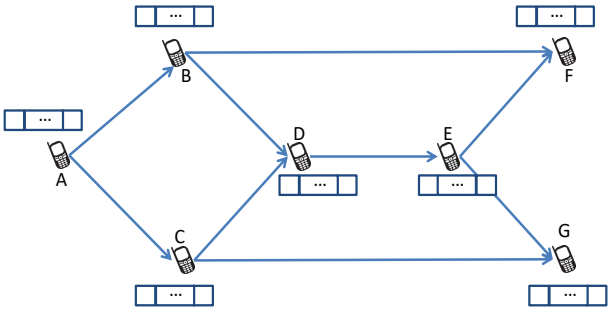


Fig. 2: Benign Scenario

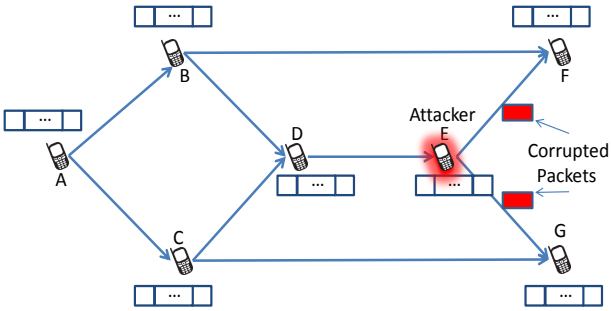


Fig. 3: Malicious Scenario

considered to be immune to alteration [18]. Thus chain entries cant be altered without considerable resource utilisation, which consequently ensures that intermediate nodes can fetch the original tags created and added by the source corresponding to the generation it received via the channel.

The following analysis is based on two scenarios, namely the benign scenario, shown in Fig. 2 and the malicious scenario, depicted in Fig. 3. In the former, an RLNC-based multi-hop network consisting from the source node $(A)$, a few intermediate nodes $(B, C, D)$ and $(E)$, and two destination nodes $(F)$ and $(G)$, allows packets to traverse multiple devices in order to reach the destination nodes. Each node in the architecture randomly selects a set of coefficients to make linear combinations of the incoming packets. Following the RLNC principle, as $(A)$ wants to deliver packets to both $(E)$ and $(F)$, the intermediate nodes create their own linear combinations of the received packets and transmit the output to the appropriate neighbours. The packets are then re-encoded until they reach the destination where the decoding occurs.

In the latter malicious scenario, the RLNC-enabled multi-hop network is vulnerable to pollution attacks. According to the above mentioned definitions, the adversary node (i.e., compromised node) injects into the network corrupted (i.e., polluted) packets, which if not detected, can reach the destination and interfere with the decoding procedure, rendering it redundant and wasting valuable resources. Tag pollution attacks are also studied, in which case the adversary tries to corrupt the tags attached to the packets instead of corrupting the packet itself. As in Fig. 3, one adversary node $(E)$ is injecting corrupted packet in the stream. The scenario assumes no mitigation mechanisms against pollution attacks in the

network; hence, the corrupted packet will reach the destination nodes $(F)$ and $(G)$ and will affect their decoding process.

### *B. Proposed security scheme against pollution attacks*

The simple RLNC based communication is prone to pollution attacks. As we mentioned different mechanisms are proposed to prevent pollution attacks using cryptographic techniques. The homomorphic message authentication codes [19] and signatures are considered as a suitable solution to prevent pollution attacks. Schemes like MacSig [13] and Dual HMAC [14] are proved to be secure in preventing both data pollution and tag pollution attacks on the expenses of an additional communication and computational overhead. We base our proposed blockchain enhanced secure network coding scheme on these MAC and signature based schemes. With the introduction of a blockchain to securely store the tags for each packet, we are improving these schemes in terms of the computational complexity and communication overhead. Further, we also achieve this with a lesser number of keys and a much simpler key distribution system, so that the key storage overhead at each node is also reduced considerably.

Our security scheme can be explained in the following steps:

1) Setup: The Key Distribution Centre (KDC) distributes a set of $L$ keys to all nodes in the network. These keys are used to create the tags at the source and to verify the tags at intermediate and destination nodes. Thus each node will have to save this key set $K_s = [K_1, K_2, ..K_L]$ with $K_i \in F_q^{n+1}$, $i.e$, each key $K_i$ will be defined over the finite field of field size $q$ with $n+1$ symbols. Further, each source node will have its own private key to sign the tags created at the source. The corresponding public key is available to all other nodes in the network. All these key distribution can be done offline in the pre-processing stage itself.
2) Tag generation: We will consider the case of a single generation for simplicity. Thus a source will have $m$ native packets where each packet $P_i = P_{i,j}$, where $i \in [1, m]$ and $j \in [1, n]$, as explained in the basic RLNC description before. For each packet $P_i$, the source will generate $L$ tags using the $MAC(P_i, K_s)$ algorithm. Then these $l \times m$ tags, along with the generation number, are added as an entry to the blockchain by the source. This step is authenticated by the private key of the source. Further, it should be noted that the tags are created before augmenting the native packet with the locally generated random coefficients. The MAC algorithm takes a native packet $P_i$ and the key set $K_s$ as its inputs and give $L$ number of tags as its output. Each tag $Tag_l$ is created according the following equation:

$$Tag_l = \frac{\left(\sum_{j=1}^{n} P_{i,j} \times K_{l,j}\right)}{K_{l,j+1}}, \quad l \in (1, L) \qquad (1)$$

3) Verification: Whenever an intermediate or destination node receives enough innovative packets to decode or re-encode the packets from a single generation, it will first verify the tags of the received packets. The verification process has two steps. The receiving node will check the generation number of the incoming message and check for the corresponding entry in the blockchain from the source. It can verify the authenticity of the entry in blockchain using the public key corresponding to the secret key used by the source to sign the entry. Now the tags stored in the particular blocks are compared with the received packets using the verification algorithm presented in Algorithm 1 at receiving node. Once the verification is successful, the packet will be re-encoded or decoded.
4) Re-encoding: The intermediate nodes have to re-encode the packets before forwarding it. However they dont create new tags at this level. In our proposed scheme, the tags are created only at the source nodes and added to the native packets at the source and further at any receiving node it is considered just as portion of the packet. Thus the re-encoding process becomes significantly simple compared to the existing schemes. The intermediate node can create the forward coding packet with the locally generated random coefficients as explained in the system model.

**Algorithm 1:** Verification Algorithm

**Data:** Received packet $\mathbf{C}_i$, $\mathbf{L}$ tags corresponding to $\mathbf{C}_i$ retrieved from the blockchain, Key set $\mathbf{K}_s$

**Result:** $\mathbf{1}$ if verification is successful and $\mathbf{0}$ if verification is failed. In case of a failed verification, the type of the attack is also reported.

**Step 1**:
Retrieve the coefficient matrix from the received packet
**Step 2**:
Multiply the tags retrieved from the blockchain with the corresponding coefficients
**Step 3**:
Compare the tags with those appear in the received codeword.
**if** *they dont match* **then**
    Report Warning and Proceed
**else**
    Proceed
**Step 4**:
Create tags for the received packet using MAC algorithm (without considering the coefficient part)
**Step 5**:
**if** *MAC algorithm output matches with the tags retrieved from blockchain* **then**
    $1 \longleftarrow$ Return
**else**
    $0 \longleftarrow$ Return

### C. Security Analysis

The security analysis describes how the proposed scheme achieves security against data and tag pollution attacks. Towards this end, the model of the adversary node considered in this paper, is defined. In details, the vulnerable points in the network are the intermediate nodes. An adversary can have control over a compromised intermediate node. We expect the adversary can completely utilize the abilities of a compromised node to create pollution attacks. In that case, a compromised node will have knowledge of the key set $K_s$, a received packet $C_i$, and is also in a position to view the original tags of the native message. Given the fact it receives enough number of codewords from a generation before transmitting, it is also possible to decode and access the content of the original message. In other words, the adversary has a strong knowledge of the content of the overall messaging scheme. This provides the adversary with enough information to virtually corrupt the benign scenario.

*1) Data pollution attacks:* In data pollution attacks, the adversary will try to corrupt the message and still try to pass the message. In our scheme, there are two processes in verification. The tags in the received codeword should match with the tags in the block chain and it should be matched with the tags created from corresponding message part in the received packet. Since adversary have the keys used for creating the tags locally, it can easily make the tags for its own message without any difficulties. However, here the adversary has to find out such a message which will also create the tags same as the one with original message since the tags in the received message will be checked with the original tags created at the source. Thus the probability of adversary producing a corrupted packet $P_i'$ satisfying this condition, is the probability of a data pollution attack to succeed. If the adversary has to create such a message, then it has to change at least one symbol of the packets content which will again satisfy the MAC algorithm with the modified packet, ie, $MAC(P_i, K_s) = MAC(P_i', K_s)$. Thus the probability of succeeding in data pollution attack is equal to the probability of finding a symbol from the field of Packet $P_i$ which has the size $q = 2^8$. So, the probability of finding another symbol from the field of $P_i$ has a probability of $1/q$. However, we have $L$ tags, thus the probability of succeeding the complete security check becomes $1/q^L$ which is negligible. Thus we can say that the proposed scheme is highly secure against the data pollution attack with a probability of $1/q^L$ to be vulnerable against the data pollution attack. It shows we are achieving a much better security compared to [13] and Dual HMAC [14] at the expense of same number of keys, but each key of our scheme is smaller than the other two resulting in a smaller storage overhead at the nodes. It should be further noted that even if an adversary tries to forge only the coefficient part of the packet, then also our scheme will be able to detect it. This is facilitated by considering the coefficients received while comparing the received tags with the tags retrieved from the blockchain.

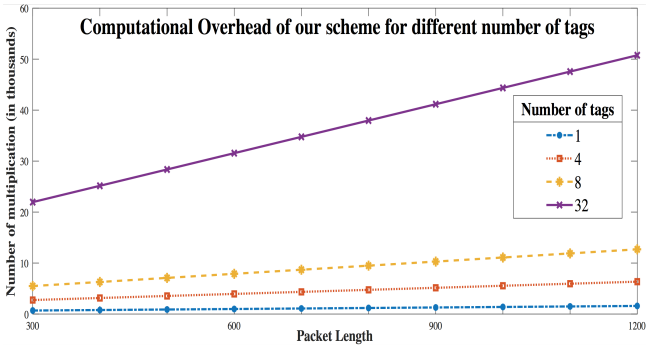

Fig. 4: Computational Overhead for different number of Tags

*2) Tag pollution attacks:* The security of our scheme against tag pollution attack is dependent on the concept of blockchain to securely transfer the original tags generated at the source to all other nodes along with enabling the receiving nodes to create and test the tags for received message by itself. Since all the nodes already have the key set available locally, they can verify whether the tags attached to each coded packet is valid or not. Further if they check it with the original tags retrieved from blockchain, it can decide whether there is a tag pollution or not. If the tags in the received packet do not match with the tags received, but match with the original tags retrieved from the blockchain, a tag pollution attack is detected. However, in such cases, a benign node can create warning in the system against the malicious node and continue with the re-encoding, since the message matched with the original tag. Further it should be noted that even a combination of adversary intermediate nodes will not have any advantage in polluting the system since the immediate innocent node will again do the verification with the original tags stored in the blockchain. This property of our scheme is very significant and give us an advantage over the practical scenario of mobile networks where a number of nodes in a particular area is compromised by the attacker.

## IV. Performance Evaluation

The proposed security scheme is closely comparable with [13] and Dual HMAC schemes [14], [20] which already are well known and secure. However, we provide an advantage over the computational and communication overhead of these schemes at the expense of a blockchain facilitated over the network. Further, we considerably reduce the key storage space required by these schemes. To test our proposed scheme we simulated a butterfly network using MATLAB R2017b on an Intel Core i7-4770K CPU operating at 3.4GHz, equipped with 8GB RAM. Furthermore, in the packet settings we considered $q = 2^8$, $n = 1024$ and $m = 32$.

### A. Computational Overhead

The computational overhead of our system over the simple RLNC based network is the overhead due to computing the tags and the signature for the tags. At the source node, we have to create $L$ tags and for each tag we need $n+1$ multiplications.

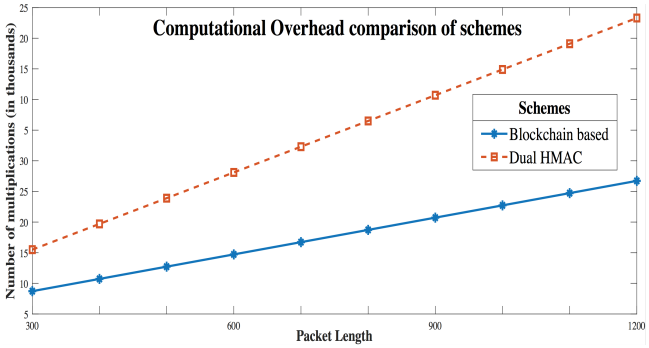


Fig. 5: Computational Overhead for different schemes

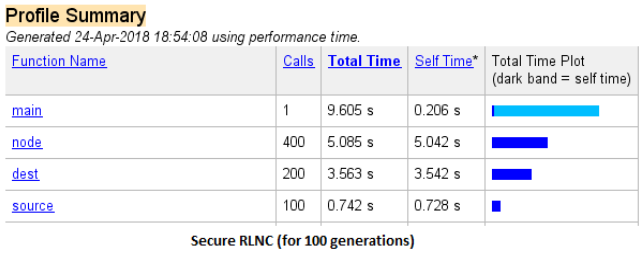


Fig. 6: MATLAB Profiler Summary for Secure RLNC

Further for the signature we need to do $L+1$ multiplications. Thus the total overhead at the source nodes counts to be $L\times(n+2)+1$ multiplications. At every other node, there will be an extra overhead of $L\times(n+1)$ multiplications for MAC verification and $L+1$ exponentiation for signature verification. Each exponentiation can be compared to $\frac{3}{2}|q|$ multiplications as shown in [13]. Figure 4 illustrate the computational overhead due to our security scheme against the number of tags used.

The computation overhead of our scheme can be directly compared to the Dual HMAC [14] scheme, which have an overhead of the second order of $L$, and we can claim a significant advantage. Figure 5 shows the computational overhead comparison of our scheme with the Dual HMAC scheme when same number of tags ($L$=8) are considered for each scheme. Further the overhead due to the security scheme over the simple RLNC implementation of same network is tested using MATLAB simulation. The results of the experiment shows that for the same network and conditions, a secure RLNC scheme takes almost double time compared to simple RLNC for simulation. However, this relation is not varying with the number of generations and the MATLAB simulation for 100 generations took less than 10 seconds to complete, as showcased in Fig. 6.

### B. Communication overhead

The communication overhead of the security scheme is directly proportional to the number of tags associated with the packet. Our scheme has $L$ tags with each tag we achieve a security level of $1/q$ probability. As explained in [19] we can use multiple tags to improve the security. For $q=2^8$, each tag is 8 bit long. However on per packet rate, we have $L$ symbols additional to provide the security. Thus we have an $L/(m+n)$ communication overhead per packet which is equal to the overhead of [19] along with an additional overhead of maintaining a blockchain. Each block in the chain consists of $L$ tags along with a signature. However this is again a part of the message and hence negligible comparatively to the total overhead. Even if we consider the same number of tags for the Dual HMAC scheme [20], it needs additional $L'$ tags to protect against the tag pollution attack making our scheme approximately $L/(L+L')$ times efficient in terms of communication overhead. Fig. 7 and Fig. 8 show the comparison of communication overhead against number of tags and the comparison of communication overhead of different security schemes, respectively. While comparing the overhead for different schemes we considered 8 number of tags for all the security schemes.

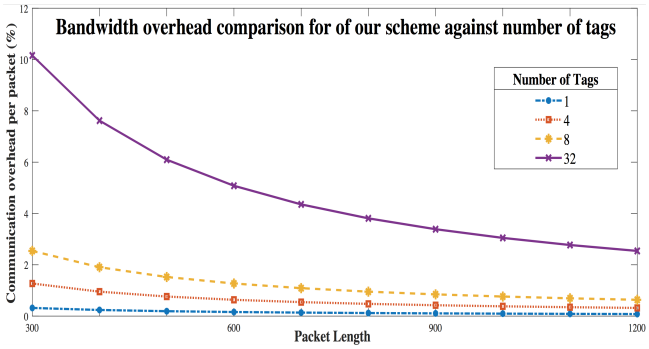


Fig. 7: Communication Overhead for different number of Tags

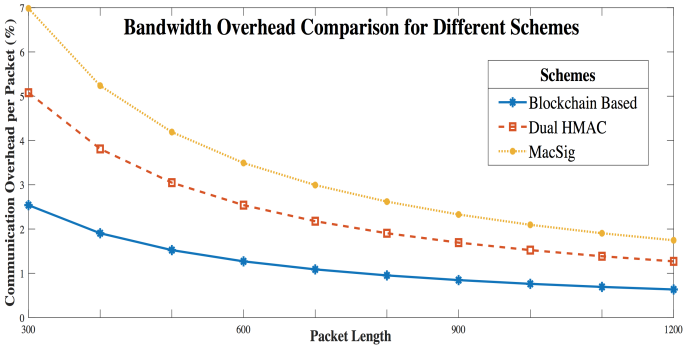


Fig. 8: Communication Overhead for different schemes

### C. Storage Overhead

Since each key $K_i$ in the key set $K_s$, will have $n+1$ symbols as per the definition, each node in the network has to store the $L\times(n+1)$ symbols to create tags. Further, for the signature verification of the tags in the blockchain $L+1$ sized private key will be stored at the source and a public key of same size will be stored at the receiving nodes. Thus the total overhead due to key storage at each node is $L\times(n+2)+1$ bytes. Further a small storage overhead due to the blockchain will be suffered at the nodes.

## V. Conclusions

5G wireless mobile networks are emerging rapidly and new standards are being drafted and accepted in a regular basis. Although a strict set of networking and service requirements have already been defined, the 5G paradigm is still open to novel proposals for optimized networking architecture, that will address these requirements for high throughput, high resilience and low latency over heterogeneous wireless small cells. In the scope of 5G deployment, this work proposes a secure network coding architecture based on the RLNC scheme in order to optimally harness available bandwidth over wireless links. Specific focus has also been given to addressing security aspects of network coding deployment. The study concentrated on a very common security concern, which is pollution attacks. The proposed secure network coding incorporates homomorphic message authentication codes and block chains in order to support a secure and energy aware key management scheme. The performance analysis of the proposed scheme using simulation tools and its comparison against well known homomorphic based security schemes indicate a significant improvement in terms of computational and communication overhead.

This work marks the initial step to build an RLNC based secure 5G network. The scheme is yet to be tested in the complete real world system to check for the problems due to channel distortions. It is also planned to extend the work to address more security challenges in the network coded systems [21], [22]. Further improvements in the underlying blockchain concept to reduce the complexity and overhead due to mining, are also planned as an extension to the work.

## References


[1] C. Tselios and G. Tsolis, "On QoE-awareness through virtualized probes in 5G networks," in *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, Oct 2016, pp. 159–164.

[2] I. Politis, C. Tselios, A. Lykourgiotis, and S. Kotsopoulos, "On optimizing scalable video delivery over media aware mobile clouds," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[3] G. Bianchi, E. Biton, N. Blefari-Melazzi, I. Borges, L. Chiaraviglio, P. Cruz Ramos, P. Eardley, F. Fontes, M. J. McGrath, L. Natarianni *et al.*, "Superfluidity: a flexible functional architecture for 5G networks," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1178–1186, 2016.

[4] L. Bolivar, C. Tselios, D. Mellado, and G. Tsolis, "On the deployment of an open-source, 5G-aware evaluation testbed," in *IEEE International Conference on Mobile Cloud Computing, Services and Engineering (Mobile Cloud)*, March 2018, pp. 1–6.

[5] C. Tselios and G. Tsolis, "A Survey on Software Tools and Architectures for Deploying Multimedia-Aware Cloud Applications," in *Lecture Notes in Computer Science: Algorithmic Aspects of Cloud Computing*. Springer International Publishing, 2016, vol. 9511, pp. 168–180.

[6] S. Mumtaz, K. M. S. Huq, J. Rodriguez, S. Ghosh, E. E. Ugwuanyi, M. Iqbal, T. Dagiuklas, S. Stavrou, L. Kanaris, I. D. Politis, A. Lykourgiotis, T. Chrysikos, P. Nakou, and P. Georgakopoulos, "Self-organization towards reduced cost and energy per bit for future emerging radio technologies - sonnet," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Dec 2017, pp. 1–6.

[7] E. Hossain and M. Hasan, "5G cellular: key enabling technologies and research challenges," *IEEE Instrumentation Measurement Magazine*, vol. 18, no. 3, pp. 11–21, June 2015.

[8] J. Rodriguez, A. Radwan, C. Barbosa, F. H. P. Fitzek, R. A. Abd-Alhameed, J. M. Noras, S. M. R. Jones, I. Politis, P. Galiotos, G. Schulte, A. Rayit, M. Sousa, R. Alheiro, X. Gelabert, and G. P. Koudouridis, "SECRET - Secure network coding for reduced energy next generation mobile small cells: A European Training Network in wireless communications and networking for 5G," in *2017 Internet Technologies and Applications (ITA)*, Sept 2017, pp. 329–333.

[9] N. Cai and R. W. Yeung, "Secure network coding," in *Proceedings IEEE International Symposium on Information Theory,*, 2002, pp. 323–.

[10] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2596–2603, June 2008.

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

[12] L. Lima, S. Gheorghiu, J. Barros, M. Medard, and A. L. Toledo, "Secure network coding for multi-resolution wireless video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 377–388, April 2010.

[13] 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*, April 2011, pp. 1026–1034.

[14] A. Esfahani, G. Mantas, J. Rodriguez, and J. C. Neves, "An efficient homomorphic mac-based scheme against data and tag pollution attacks in network coding-enabled wireless networks," *International Journal of Information Security*, vol. 16, no. 6, pp. 627–639, 2017.

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

[16] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Allerton Conference on Communication, Control, and Computing*, October 2003. [Online]. Available: https://www.microsoft.com/en-us/research/publication/practical-network-coding-2/

[17] C. Tselios, I. Politis, and S. Kotsopoulos, "Enhancing SDN security for IoT-related deployments through blockchain," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2017, pp. 303–308.

[18] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[19] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-Based Integrity for Network Coding," in *Applied Cryptography and Network Security*, M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 292–305.

[20] 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*, vol. 11, no. 7, p. 510251, 2015.

[21] V. N. Talooki, R. Bassoli, D. E. Lucani, J. Rodriguez, F. H. Fitzek, H. Marques, and R. Tafazolli, "Security concerns and countermeasures in network coding based communication systems: A survey," *Computer Networks*, vol. 83, pp. 422–445, 2015.

[22] A. Antonopoulos and C. Verikoukis, "Cops: Cooperative statistical misbehavior mitigation in network-coding-aided wireless networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1436–1446, 2018.