IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 19, 2019, accepted January 29, 2020, date of publication February 21, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975761

On Security Against Pollution Attacks in Network Coding Enabled 5G Networks

VIPINDEV ADAT VASUDEVAN, (Member, IEEE), CHRISTOS TSELIOS, AND ILIAS POLITIS, (Member, IEEE)
Wireless Telecommunication Laboratory, University of Patras, 26504 Patras, Greece

Corresponding author: Vipindev Adat Vasudevan (vipindevadat@gmail.com)

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme through the Marie Sklodowska-Curie under Grant 722424.

ABSTRACT Future communication networks need to harness the available spectrum more efficiently to cater the requirements of the ever-increasing digital devices. Higher data rate with low latency is a major requirement of future communication networks. Network coding is arising once more as an enabler for satisfying the bandwidth requirements of future multimedia and resource hungry services. However, network coding techniques suffer from security vulnerabilities that will eliminate any bandwidth profits. Specific attacks in network coding like pollution attacks are extremely dangerous due to the nature of encoding and spreading inside the whole network. They deteriorate the bandwidth efficiency and even disrupt proper decoding of any message at the receiving end. Further, in a wireless environment, the authenticity of intermediate nodes is not easy to ensure, making it easier for an attacker to be part of the network. Thus counteracting pollution attacks in network coding becomes very important for practical applications of network coding enabled networks in the future generations of mobile communication. There has been a lot of research interest in this direction resulting in a few interesting approaches for secure network coding. However, most of the schemes fail to meet the expected standards or incur high overheads to the system. Schemes addressing the dense heterogeneous networks efficiently are yet to be proposed. This study surveys the security vulnerabilities of network coding, particularly those imposed by pollution attacks, as well as, the corresponding countermeasures. The survey goes a step further and includes a potential secure implementation of network coding enabled 5G networks, based on cooperating small cells.

INDEX TERMS 5G mobile communication, cryptography, network coding, network security.

I. INTRODUCTION

The Internet of Things (IoT) and the fifth generation of wireless technology are restructuring the digital world. With the current trend of exponential growth in the number of connected devices and the amount of traffic generated by them [1], current communication technologies need a major upgrade to serve the requirements put up by these devices. The next generation of wireless communication is expected to serve these large number of devices with high data rate and a low end to end latency. The researchers in the domain point to the requirements of not just an upgrade but an almost new system needs to be rolled out as the next generation communication technology [2] varying from the radio access technologies to the cell structure. The advancements in different domains such as radio access networks, software-defined networks, network virtualization etc. play a major role in the next-generation networks. Further, the enormous number of connected devices and the requirements arising from the user side also motivates the coverage areas to be extended almost everywhere, supporting higher network density, and providing a high quality of service. It expects to provide uninterrupted connectivity virtually anywhere, anytime. 5G networks are targeting high standards in different aspects such as data rates up to 1Gbps, end to end latency as less as 1ms and 90 percent energy efficient compared to 4G along with the high quality of service provisioning and higher coverage [3]. These expectations are specified in different 5G use cases as Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC) [4].

The pilots of the 5G paradigm in recent years [5], [6] address these challenges with the support of several technological solutions. Also, many of the standard structures

The associate editor coordinating the review of this manuscript and approving it for publication was Kanapathippillai Cumanan.

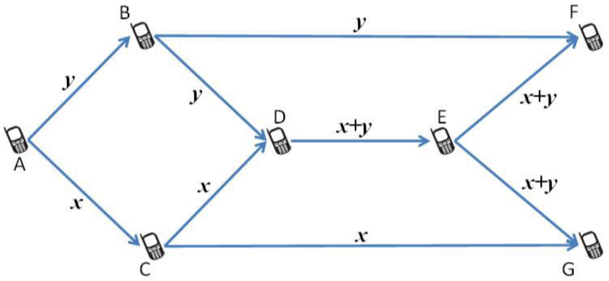

FIGURE 1. Butterfly network representing network coding.

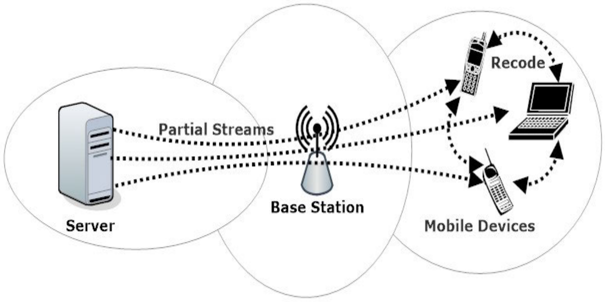

FIGURE 2. Network coded cooperation scenario.

that are being used in the current network technologies are having drastic changes in the 5G. The Cloud Radio Access Network (C-RAN) [7] is expected to be an integral part of the 5G architecture. The C-RAN acts as a pool of baseband processing units (BBU) operating at the cloud and separates from respective radio heads. This concept provides different possibilities for future networks. C-RAN reduces the processing requirements at the radio access points as well as reduces the latency and processing required during handovers in the same BBU pool. More importantly, it also supports the on-demand mobile small cells without extensive computational requirements. The whole paradigm is shifting from a base station centric approach to the user-centric approach gradually [2]. Small cells and device to device communication support this paradigm shift. Highly dense heterogeneous cells characterize such a future network environment more realistic [8]. This heterogeneity applies to different layers; the radio access technologies that will be used (5G, LTE, Wi-Fi, blue tooth etc), device capabilities being varied from simple sensors to complex portable devices, cell structures like macro, pico and small cells and obviously serving different applications, all at the same time. Further, these heterogeneous networks of devices also run a large variety of applications that come with different requirements. The live streaming of high-quality videos that ask for high data rate and critical applications related to vehicular networks and other industrial applications which require ultra-low latency are just a few examples showing the extent of 5G networks. This also provides a platform for many new technologies to be incorporated into the studies related to 5G networks to enhance the quality of experience. The network coded cooperation of small cells is one of the major research directions which also satisfies the different use cases of 5G. Recently, the European Union funded project SEcure network Coding for Reduced Energy nexT generation mobile small cells (SECRET) [9] has started investigating this possibility and is proposing promising advancements towards a secure network coding enabled mobile small cell environment for 5G and beyond networks.

Network coding is a relatively young branch of study in the field of network theory, introduced in the seminal paper of Ahlswede *et al.* [10] in 2000. Even though the concept of network coding was existing way before that [11], from then on, the advantages of network coding are well investigated. The basic idea of network coding is to allow intermediate nodes to perform linear operations on the packets it receives in its incoming edges and send these encoded packets in their outgoing edges instead of simply forwarding the packets. This reduces the number of packets to be sent over a channel and helps to achieve the maximum efficiency promised by the max flow min cut theorem [12]. The initial studies and discussions of network coding were explained with the basic example of a butterfly network performing X-OR additions on the packets, as shown in fig.1. In the given example, each edge or channel has an upper bound of one bit at a time and combining information at node D helps to achieve the optimal efficiency in sending the packets from source A to destinations E and F. Further, it has evolved from simple modulo additions of two packets to encoding multiple packets and sending the digital evidence instead of whole packets. This significantly improves the bandwidth efficiency of the network. Linear network coding also incites robustness and adaptability of the environment [13]. Since network coding tries to achieve optimum efficiency in terms of bandwidth usage by sending combined packets over different channels, it also enables some erasure corrections and imparts some resistance against man in the middle attacks. However, network coding by itself provides only a weak security [14]. Network coding [10] proves to be a worthy candidate to enable cooperation between the small cells to provide higher throughput over the network. Further with random linear network coding [15], it also becomes very suitable for the wireless environment with unstable topology. In a cooperative environment, it can produce the upper bound efficiency in multicasting. The future networks with small cell environment featuring device to device communication and cooperative environment of devices, try to ensure that every user in the network will be fairly provided with the required services [16]–[18]. In such a cooperative environment, as shown in fig 2, the receivers would receive partial streams from the channel and cooperate with the neighbouring nodes to create the complete information. This improves the bandwidth efficiency of the system compared to the current LTE based network without cooperation as illustrated in fig 3.

Even though network coding provides a highly resilient and bandwidth efficient communication, it also introduces new vulnerabilities in the security system. Network coding inherently provides weak security against some known attacks like eavesdropping, due to the coding and recoding at

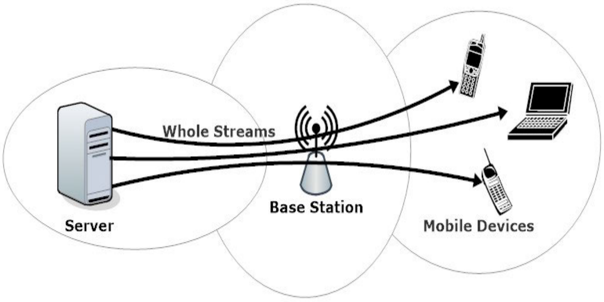

FIGURE 3. No cooperation scenario.

the nodes, it also suffers from specific security threats. Pollution attacks could be termed as one of the biggest and specific challenges to be addressed by network coding environment. Since every participating node can recode the packets it received, a malicious node in the network can send a packet that doesn't belong to the original content and it can completely mislead the other benign nodes from retrieving the original content. The pollution attack becomes very dangerous in case of multiple intermediate nodes being connected between the source and destinations since it spreads across the network as the malicious packet moves on. Pollution attack and its effects are discussed in a more detailed manner in the subsequent sections of the paper.

This paper analyses the security challenges of network coded cooperation for mobile small cells with emphasis on pollution attacks. Even though there have been some interesting surveys on the security of network coding implementations [19]–[21], an extensive study focused on pollution attacks and countermeasures on the background of upcoming 5G and beyond network requirements is still valid. This paper focuses on different types of pollution attacks in the network coding environment and discusses some of the major countermeasures against pollution attacks in the literature, in detail with comparisons. Further, the paper stresses the cryptographic based integrity schemes to cop up with the requirements in the future generation networks and work towards a secure network coding enabled mobile small cells for 5G networks. The paper also discusses some proposed integrity schemes for secure mobile small cells and the future research directions and challenges existing in this domain. This paper differs from most of the state of the art with the specific focus on the analysis of different integrity schemes with the backdrop of future network requirements. The remaining sections of the paper are as follows. Section II discuss some preliminary concepts of network coding followed by III discussing the additional requirements and KPIs for security schemes in 5G environment and provides a base for the topics of further discussion. Section IV discuss the security concerns of network coding with an emphasis on pollution attacks. It is followed by a section discussing the pollution attacks and countermeasures existing in literature against these attacks. Section VI compares the different important research directions in the literature. Section VII discusses the specific security requirements for a secure implementation of network coding enabled cooperative small cells. The paper concludes by discussing the future directions towards a secure and energy efficient network coding enabled small cells for next-generation wireless technologies.

II. NETWORK CODING PRELIMINARIES

Before discussing further regarding the security challenges of network coding aware protocols, this section introduces some preliminary concepts about different network coding protocols. Starting from the basic concept of modulo additions of two information flows, network coding has evolved to different detailed protocols suitable for different network conditions and different application types. In general, we can divide them into two types based on the network state awareness of the nodes, as State aware NC protocols and Stateless NC protocols [22]. In state aware network coding protocols, like COPE [23], the participating nodes have some idea regarding the network topology and utilize this information for improved performances in terms of throughput and robustness. The state aware NC protocols are very efficient since the nodes have local state awareness about the neighbouring nodes and network topology, but this dependency on the network topology also makes the system more vulnerable to attacks such as wormhole, blackhole and eavesdropping. COPE is one of the well-known state aware NC protocols, based on an opportunistic coding approach, and few other such protocols are discussed in [24]–[26].

State aware NC protocols are suitable if the network conditions are almost stable. However, in the wireless networks including a lot of mobile devices, this situation is very hard to achieve. Moreover, the dependence of the communication protocols on network state and topology also introduce security vulnerabilities. Stateless NC protocols, like randomized network coding [15] are more suitable for the wireless networks. In stateless NC protocols, the mixing and coding of packets don't depend on the network topology. These protocols do not expect any specific network conditions and code the packets such as if a node receives enough number of individual packets from whatever incoming links, independent of the network state, the receiver will be able to decode the packets. Stateless NC protocols are very much suitable to dynamic topologies, such as in the case of mobile ad-hoc networks. It also makes stateless NC protocols better suited for future wireless networks. Since these protocols don't depend on the network conditions, they also provides better immunity to many security challenges. However, it also needs more sophisticated coding operations compared to state aware protocols. Further, these network coding operations need to be performed in a relatively larger finite field to ensure that the coding and decoding operations are performed securely and efficiently. This introduces a little extra overhead in terms of computation and communication requirements for the systems employing stateless NC protocols.

Random Linear Network Coding (RLNC) [27] is the most common and popular stateless NC protocol. RLNC is emerging as one of the most efficient and suitable systems for network coding enabled wireless networks. It enables the nodes to encode the packets with the help of locally generated random coefficients and broadcast the packets. This also reduces the requirement of predefined paths making it very much suitable for the wireless environment. In RLNC based networks, original packets will be appended with some random coefficients. These random coefficients act as a key to decode the original packets. Thus decoding of an RLNC packet becomes a problem of solving a set of equations with a fixed number of variables. It also improves the erasure correction since a receiver can decode the packets if it receives a particular number of coded packets. These advantages of RLNC make it the most suitable candidate for future mobile communication networks.

Another classification of network coding systems is based on the mixing of packets in the intermediate nodes. When there are more than one source nodes in the network, there is a chance multiple information flows passing through a particular node. Based on how the network coding operations are performed over multiple flows, there is another classification of network coding protocols as inter-flow network coding and intra-flow network coding. In intra-flow network coding, the intermediate nodes perform network coding only on individual flows [28]–[30]. That means, only the packets from the same source will be considered for coding at the intermediate nodes. However, in the inter-flow network coding schemes packets from multiple flows are mixed together during recoding at the intermediate nodes as shown in different schemes like [31]–[33]. This could enhance the efficiency of coding, however, makes the system very complex and extremely difficult to identify the security threats. IN inter-flow NC protocols, packets from different sources are coded together, which requires a highly secure and authenticated environment. Also, this makes even homomorphic signature schemes [34] which are commonly used with inter-flow network coding to ensure the integrity of packets invalid because of the multiple sources generating packets in a single encoded packet.

The future wireless networks involve a lot more mobile devices and dynamic topologies. The stand-alone small cells and highly mobile devices make the network topology unpredictable. Further, a large number of devices with digital identity also makes it difficult to authenticate and ensure a preemptive trusted environment. Impersonation and identity management in the IoT environment [35] also leads to several challenges in ensuring a secure and authenticated environment as preferred by inter-flow coding schemes. Thus further in this survey, we concentrate mainly on intra-flow, RLNC based approaches, unless it is mentioned otherwise. This also helps to create a balance between the security of the system with the complexity of coding operations.

III. REQUIREMENTS OF INTEGRITY SCHEME IN 5G ENIVROMENT

Future communication networks are expected to handle a dense network of mobile devices with vivid capabilities and requirements. The heterogeneity of the network will be multi-dimensional; varies in terms of computational capability, memory availability, radio access technologies, application requirements, mobility, and resource requirements. Further, with the small cell environment, the network architecture itself can vary during the communication. In such a vivid environment, the security challenges also manifolds. Any security scheme to address a mobile small cell environment has to meet some additional requirements other than the security concern [36].

A. SCALABILITY

The wireless environment undergoes frequent changes. The participating devices will be moving around and the network dynamics reshape very frequently. Further, devices moving from one small cell to another or joining or leaving the network make the network unpredictable. Also, it rules out any pre-defined or rigid initial conditions for a security scheme. Any security or integrity scheme for a mobile small cell environment needs to address the challenges of mobile nodes and also varying network topology. For example, it can not depend on a strict key pre-distribution for ensuring security. Also it should be able to accommodate new nodes to join the network during the communication and works without any disruption if some node leaves the network during the process. The system should work flawlessly with a few participants and should also be capable of handling a dense network without any issues.

B. OVERHEADS

Security schemes always incur some overhead on the system. Keeping these overheads to a minimum is a primary criterion during the design of any integrity scheme. There are different overheads to be considered during the process, mainly communication, computational and storage overheads. Storage overheads are mainly due to the additional keys that every node will have to store to perform the verification process. The communication or bandwidth overhead occurs due to the additional information that is needed to pass through the communication channel. Higher the number of bits used to ensure the security, lower the bandwidth efficiency of the system. It is also to be noted that if any other fixed channel is being used for any security-related information, this may not be considered in the overhead calculation because another fixed channel is being used to send those signals. Computational overheads due to the integrity schemes are generally because of the additional finite field operations required for the integrity schemes. Generally, in MAC based schemes the computational costs are measured in terms of finite field multiplications which create considerable overhead and additions are considered as negligible overheads. In homomorphic signature based schemes, finite field exponentiations are required to perform the verifications and these have higher computational complexity than multiplications. During the performance evaluations, these overheads are either measured using such mathematical relationships or as processing time consumed by the additional operations. Dependency of these overheads on the number of users, if any, also leads to scalability issues which are discussed in some of the schemes explained in the following sections.

C. LATENCY

Low latency communication is one of the major targets of 5G and future networks. The end to end latency in communication is expected to be as low as a few milliseconds in the upcoming generations. Any significant computational delay or communication delay as part of the integrity scheme can have an adverse effect on the end to end latency of the network. Further, schemes depending on time asymmetry also introduce some delay in completing the communication. Along with latency, delay in initialization of any node with the network is also considered during the studies, for example, pre-distribution of keys. If the initialization process can be done without considering the network topology during the device manufacturing, then it is omitted while calculating initialization delay.

From the design perspective of any security scheme, these three requirements and their interdependency are very important. The overhead can also depend on the topology and varies with the number of users in the network. Thus the size of the network does affect the overhead constraints as well. In this study, we focus on these constraints of the state of the art integrity schemes and discuss how well each scheme suit for future wireless networks.

IV. SECURE NETWORK CODING AND POLLUTION ATTACKS

Security challenges in network coding were first studied in [37] in 2002. A secure network coding scheme for a communication system over a wiretape network (CSWN). They analyzed the security of linear coding over a wiretap network where one of the links are compromised by a wiretapper and proposed the sufficient condition for a secure and decodable linear network coding system. This was one of the initial works combining network coding with information security. From then onwards there has been a lot of research in analysing the security of network coding schemes. As shown in [37] linear coding itself gives some security against wiretapping but it also suffers from other security challenges. Some of these challenges like pollution attacks are extremely severe in network coding scenario compared to normal switching networks since the packets in transition are being coded at intermediate nodes. In this section, we analyse some commonly known security challenges from the perspective of a network coded environment summerizing some of the existing works on the security of network coding [19]–[21] and also discuss the pollution attack as a special case of security challenges.

One of the major security challenges in a wireless environment is eavesdropping or wiretapping. It corresponds to an attacker who has compromised a link and listens to the packets being transmitted over that link. This passive attack over the information that is passed over the compromised links is very common in the wireless environment. Depending on the capabilities of the compromised links or nodes, the severity of attacks can vary from gathering of partial content to capturing important messages, keys and other valuable information. References [14], [38] are a few initial studies on how network coding systems react to eavesdropping. In network coding environment, simple wiretapping becomes difficult due to the coded packet transmission. Especially in stateless NC protocols, wiretapping a link or a particular node may not help the attacker to gain any useful information. For example in RLNC, the packets will be coded with random coefficients and sent over random channels so only capturing the packets in a single channel will not help the attacker to decode any useful information [37], [39]. Thus, network coding inherently provides some weak security against Eavesdropping. Further improvements in preventing eavesdropping in network coding are studied in different works like [39]–[41].

Another common passive attack over wireless networks is traffic monitoring and analysis where an adversary node will try to analyze the data traffic to find traffic trends and some information about the participating nodes and network topology. The authorization of intermediate nodes to recode the incoming packets make it a bit difficult to prevent traffic analysis in a network coded environment. However, a proper encoding scheme will help to protect the privacy of nodes. Further, the stateless NC schemes do not provide much information about the network topology making the attack insignificant. A few schemes have studied the impact of traffic analysis in network coding environments and proposed schemes which are efficient in preventing such attacks [42]–[44].

When network coding provides better security than legacy routing schemes to passive attacks, the effect of active attacks like Byzantine modification can be very dangerous with NC based networks. The network coding schemes allows the intermediate nodes to recode and mix the packets on the fly which makes it difficult to identify if any activity by a node is legit or not. In traditional routing networks, any modification to the packets on the fly can be considered as a malicious activity wherein network coding the modification of packets on the fly makes the essence of the scheme. Thus active attacks that directly disrupt the network operation or packets being transmitted become dangerous in network coding environments [20]. Different types of denial of service (DoS) attacks can target network coding aware schemes and be destructive. Denying one or a small set of nodes from participating in the communication protocol may not have a significant impact on a stateless NC protocol but as the volume of attack increases, the number of packets being received at the sink will reduce and it can result in a state of not enough packets to properly decode the information. The DoS attacks in network coding environment are studied in detail by [45] and some of the common denial of service prevention schemes for wireless networks are discussed in [46]–[48]. Further, jamming also accounts for DoS in the wireless networks. In network coding environment, jamming a node can be done easily by sending a lot of packets to it from different adversary nodes. It will also be difficult for an intermediate node to find out which packets are genuine and should be used for generating its own coded packet. However, simply not forwarding a packet to prevent packet flow in a network coding environment is not that vital since packets are received in different incoming channels and most of the time no particular channel will be specifically important. However, this sending of bogus packets to jam a node can also lead to resource depletion problem and prevent the node from participating in the communication [49]–[51].

Another major attack in network coding environment is related to the entropy of packets being sent. Especially with linear network coding schemes, the receiving nodes require enough number of innovative packets received at the sink to properly decode the packets. As the number of non-innovative packets received increases, the efficiency of the scheme is dropping and resources are wasted. The entropy attack is when an attacker node, mostly an insider node, repeatedly sends valid but non-innovative packets down its outgoing links. This leads to the resource exhaustion at the receiving nodes and degrades the efficiency of the scheme. It can be more serious if these packets may be innovative for the immediate neighbour but of no use for a node down the communication line, known as global entropy attack. Reference [52] simulates entropy attacks and differentiate between local entropy where the packets are non-innovative for the immediate downstream node and global entropy where the packets are innovative for the immediate node but not for a distant downstream node. A few other studies are also done on entropy attacks over network coding [53], [54].

Byzantine fabrication and modification are the major and most popular active attacks in network coding environments. In byzantine fabrication attacks, the attacker creates packets with invalid content and transmit over the network. It can also be a fabrication of invalid headers like routing table overflow, route poisoning or ACK pollution [55]–[58]. Byzantine modification or pollution attacks are the most popular and dangerous attack among the different network coding security challenges [19]. Also, it is one of the most studied security challenges specific to network coding environment. In pollution attacks, the malicious insider nodes will perform incorrect coding operations and send invalid packets over the downstream links. This will lead to the decoding of incorrect packets at the sink and negate whatever throughput efficiency being achieved by network coding. However, the intermediate nodes cannot be prevented from coding the packets it received, thus preventing pollution attacks difficult. Pollution attacks are epidemic if unchecked at the earliest possible node. Once a polluted packet is introduced in the network, it could spread over all the paths it travels and multiplies the degradation of the throughput efficiency. Further, with inter-flow networks, it becomes more complex because no node cannot be completely trusted and mixing different flows make it very complicated to detect polluted packets. However, we restrict our studies to only intra-flow NC protocols, as mentioned before. Allowing intermediate nodes to mix and code the packets result in a completely new packet being in transit with some information about the original packets. This completely contradicts the basics of most of the commonly used integrity schemes. Further, the authenticity and genuineness of the intermediate nodes come to question. If an intermediate node inserts a malicious packet into the coding process, the receiving nodes won't be able to decode the original packets even if it receives enough number of innovative packets. Increasing the devastation of a malicious packet being inserted, it can travel in the network, getting mixed and coded with more packets and completely dismantle the information being transmitted. This is one of the major challenges in the network coding paradigm, known as pollution attack. Figure 4 shows the conditions of RLNC without a malicious node in the network and Fig. 5 shows an RLNC network including a malicious node. Pollution attacks are one of the most dreaded attacks in a network coded environment because it negates all the advantages imparted by network coding by exploiting the basic concept of network coding. Commonly used encryption schemes and cryptographic solutions for the integrity of packets won't work with network coding enabled environment since the packets in transit are frequently modified. Thus homomorphic cryptographic schemes emerge as the cryptographic solution for network coding enabled networks. Homomorphic schemes allow some degree of computations over the encrypted packets and still be able to decrypt them. This suits the requirement of network coding paradigm for integrity schemes. However, there are few other approaches including information theoretic schemes which ensure the security of network coding. A detailed analysis of many interesting approaches for secure network coding against pollution attacks is discussed in the next section.

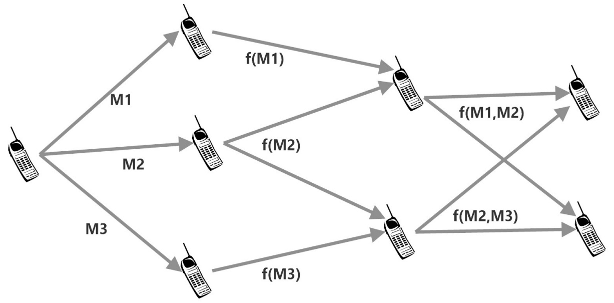

FIGURE 4. Benign scenario.

V. COUNTERMEASURES TO POLLUTION ATTACKS IN NETWORK CODING

The pollution attack in network coding is catastrophic due to its nature of spreading to the whole communication channels after the first injection of a polluted packet. Since the packets are mixed or coded and then forwarded at every node, a polluted packet if not detected, will also get involved in the

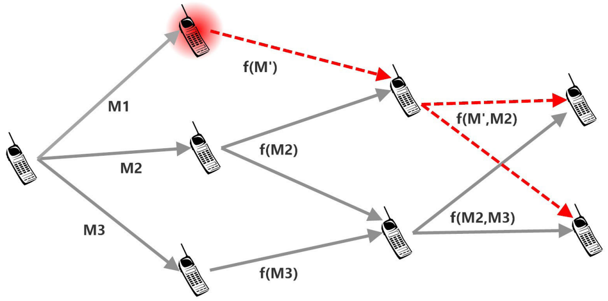

FIGURE 5. Malicious scenario.

process and pollutes all the packets with it. Thus detecting the pollution attack at the earliest point is of utmost importance. There are different integrity schemes against pollution attacks in network coding. Since mobility and heterogeneity are among the basic requirements of future networks, stateless network coding should be considered. Further, Random linear network coding (RLNC) has been identified and proposed for wireless networks [27] so we will analyze the techniques which are most suitable for RLNC based networks. Broadly, all these can be divided into the following two types.

1) Cryptographic Approaches
2) Information Theoretic (Non-Cryptographic) Approaches

A. CRYPTOGRAPHIC APPROACHES

Existing cryptographic approaches do not work for network coding environments due to the mixing of packets on the go. In legacy networks, once a packet is generated at the sender, it will never change while in transition, unless it is modified by a malicious user. This enables the generally used cryptographic schemes such as signature schemes to verify the integrity of packets in transit. However, in network coding enabled networks, intermediate nodes are capable of coding the packets which make such approaches out of scope. The integrity schemes to be used in network coding should be homomorphic in nature. The basic requirement for such schemes is that it should be able to verify the integrity of packets even if linear mathematical computations are performed over the packets. In other words, the integrity schemes for network coding enabled networks should be able to verify the integrity of individual packets from linear combinations of these packets. Homomorphic cryptographic schemes are again classified into homomorphic signature schemes and homomorphic message authentication codes (MACs). Homomorphic signatures work by signing a linear subspace of the original packets so that any combination of them can be identified with the signature attached to it. On the other hand, in homomorphic MAC based schemes, a tag with homomorphic properties (i.e, if two vector-tag pairs (v_1, t_1) and (v_2, t_2) are given, anyone can create the tag t for vector $y = a_1v_1 + a_2v_2$) is attached to each packet. Any cryptographic scheme requires some keys to be distributed between the participating nodes in order to do the integrity check. Signature-based schemes employ public key cryptography where MAC based schemes employ private key cryptography. In both cases, most of the times, the keys are pre-distributed by a secure key distribution center, as shown in Fig. 6. Some of the cryptographic schemes are discussed below. Few of these schemes use both signatures and MACs to provide better security while some use specific key distribution scenarios to prevent situations of more than one compromised nodes in the network.

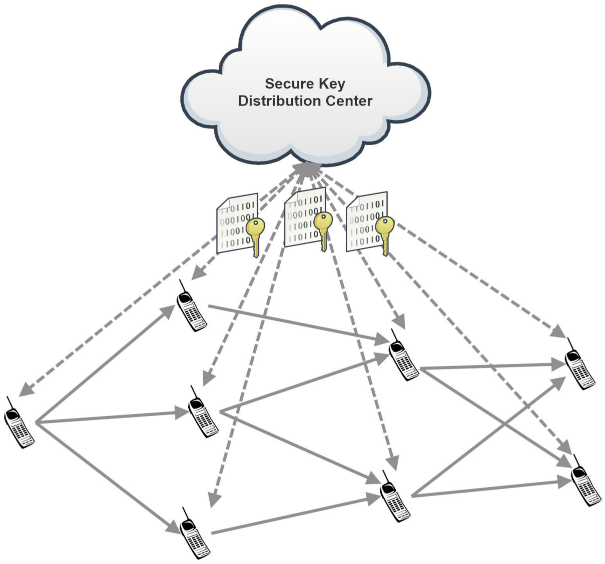

FIGURE 6. Generalised key distribution scenario for cryptographic approaches.

In 2006, the homomorphic signatures for network coding are proposed by Charles, Jain, and Lauter [34]. This work proposed a homomorphic signature scheme based on elliptic curves to ensure the security of network coding. The proposed scheme was capable of signing a linear space with signatures so that the network coded packets can also be verified. The signature scheme claims to have hardness levels of discrete log problem and co-Diffie-Hellman problem on elliptic curves. Even though this work manages to avoid the requirement of the secure channel for the hash value exchange, the complexity of the proposed scheme was very high. Each node will have a signature creation overhead of $O(d_{in} \log p \log^{1+\varepsilon} q)$ bit operations, where d_{in} is the in-degree of the vertex in(e and verification overhead of $O((d + k) \log^{2+\varepsilon} q)$ bit operations per signature $h(e)$. However, this scheme has a fixed bandwidth overhead of 2 log q bits per signature, which depends only on the chosen prime numbers. It was suitable for a linear network coded environment but imposes a high computational complexity of Weil pairing for internal nodes and encoding-decoding processes and initial requirements of finding a suitable elliptic curve and torsion points.

Microsoft was one of the initial beneficiaries of network coding. They launched a file swarming application called Microsoft Secure Content Downloader, based on network coding [59]. It explains a cooperative secure network coding for large-scale content distribution with mobile clients in a heterogeneous network. Further, they extended this work to explain the security aspects of the system in [60]. This work addresses the entropy and jamming attacks (pollution attack) and proposes a basic homomorphic hash based security mechanism. However, they further emphasize a cooperative security scheme to reduce the computational complexity due to the hash verification process. Even though the idea was proposed for content distribution, it can be well connected with the requirements of wireless mobile networks. However, this scheme expects there is a secure channel for communicating the hash values between the nodes. It is discussed in this paper by scaling this network from a finite set of users and fixed percentage of malicious users to a theoretically infinite number of users. However, the hash based schemes need a private key being shared between each pair of the nodes and this creates challenges in a practical network to ensure a completely connected mesh network. The computational overheads and running time depend on the multiplications for hash calculation and exponentiations for hash verifications which in turn depends on the number of malicious users, packet size, number of codewords. As we already discussed, homomorphic hashes are computationally complex than homomorphic signatures or MACs.

An efficient homomorphic signature scheme against pollution attacks [61] was proposed by Zhen Yu et.al. in 2008. It was one of the initial proposals to use RSA type asymmetric key cryptography to create a homomorphic signature scheme against pollution attacks. The scheme introduces a linearly homomorphic signature scheme which enables the forward nodes to sign over their outputs using the signatures they received as inputs and used for encoding, without knowing the private key of the source node. It enables the intermediate nodes to verify the incoming packets using the public key of the source node and coding coefficients attached to the packets. Further, the scheme enables batch verification of messages which will help to reduce the computations required at the intermediate nodes. Most of the security schemes against pollution attacks based on homomorphic signature follow the same pattern of signature generation. The security of this scheme depends on the hard problem of integer factorization to find a valid signature for a polluted packet or finding a hash collision for a genuine packet which corresponds to a discrete logarithm problem. Finding a hash collision by brute forcing depends on the finite field used for the network coding scheme. Authors also considered the situation of resource-constrained nodes such as wireless sensor networks where computation and verification of signatures may create an unacceptable overhead and proposed a lightweight security scheme by compromising the security level of the scheme. This lightweight signature scheme could be compromised if the malicious node could listen to a number of packets and signatures. Still, the scheme was much faster than the previous one in terms of computation and verification of signatures. This work also presents a comparative study of the overheads and running time for the proposed security scheme and its lightweight alternative against the [34], [60]. The computational overhead of the original scheme is slightly higher than [60] but reduced in the lightweight scheme. However, it is to be noted that the verification of the signatures still requires $(2 + m + n)$ exponentiation and this overhead is not reduced in the alternative scheme as well. The running time for signature verification on a Pentium IV, 3GHz Linux machine was 1.43s per message during the simulations. However, Yun *et al.* [62] identified the flows in this work and proposes minor changes to make it properly homomorphic. However, the scheme is still vulnerable to trivial no-message attacks or if the adversary is able to eavesdrop some messages, the scheme can be completely compromised.

In 2009, NCS1 and NCS2 signature schemes [63] were proposed by Dan Boneh et.al. which further explained the algorithms for homomorphic signatures for network coding systems. NCS1 is a homomorphic signature scheme based on the random oracle model and its security is based on the co-CDH assumption in the random oracle model. The second scheme, NCS2 is based on a weaker discrete logarithm assumption and it is an extension of the work by Krohn *et al.* [64]. In both the proposals, the authors try to ensure secure signatures over a linear subspace. It enables the verification procedures even if genuine packets are mixed up by the intermediate nodes. The tuple of polynomial time algorithms (Gen, Sign, Combine, and Verify) explained in the context of NCS1 is followed in most of the homomorphic signature schemes, with modifications in the algorithm, but the method of explanation has become somewhat standard. These schemes enable secure signing of a linear subspace and ensure that knowledge of a signature doesn't allow any adversary to forge a valid signature for an element out of the subspace without solving the mathematically hard problems. It also facilitates the signing of the stream of data, without knowing the whole file from the starting. The NCS schemes also define a lower bound on the signature size as $mlog_2\ p$ where m is the size of the generation and p is the field size. In NCS, the signature creation requires $m+n$ exponentiations in the bilinear group. These signature based schemes do not depend on the number of participants because a public key is used for verification and it can be easily distributed to all the participants. There is no need for a secure channel or private key pairs to be shared with all the participant nodes.

Homomac [65] was introduced in 2009 as an integrity scheme based on Message Authentication Codes to prevent pollution attacks in network coding. A triple (Sign, Combine, Verify) of polynomial time algorithms is proposed to explain the homomac. Authors propose a homomorphic MAC scheme based on the classic MAC of Carter and Wagman [66] to satisfy the requirements of network coding. It ensures that if an intermediate node has two vector-tag pairs, it can create a tag for the combination of the known vectors from the known tags. Thus only the source node will create the tags from original packets and the intermediate nodes combine the vectors and corresponding tags to create new encoded packets. The verification process using the shared secret key ensures that any received packets with genuine tags are not modified. However, in this case, only the sink node which has secret keys can verify the tags. In order to facilitate verification at the intermediate nodes, they use a key distribution mechanism similar to Canetti *et al.* [67]. This broadcast MAC is the useful version of homomac and it is c-collision resistant secure against pollution attack as per the definition. It means that the scheme is secure unless c neighbouring nodes are simultaneously compromised or malicious. The security aspect of homomac depends on two cases: the attacker could forge a tag for a nonexistent vector or the attacker could find another vector that matches one of the existing tags. The first case is a computationally hard problem and the second case has a very less probability based on the field size (usually 2^8) of the generation. Further, it is possible to enhance this security by adding multiple tags to a packet, at the expense of more computational and communicational overhead. The MAC creation and verification overhead depend on the number of multiplications being performed. In Homomac, $n + m$ multiplications are needed to create one tag. However, to prevent colluding attackers in the broadcast MAC integrity scheme, the sender attaches l tags and each receiving node will verify only a tuple of them depending on the keys available to that particular node. A c-cover based key distribution is performed to ensure that the scheme is c-collision resistant. However, this creates a scalability issue. Homomac and most of the MAC based integrity schemes that followed it depended on some kind of specific key distribution schemes (mostly c-collision resistant). Such systems need to append more tags to ensure colluding attackers can not deceive the integrity scheme. This increases the bandwidth overhead but does not improve the detection probability proportionally. Further, in a dense network of mobile users, deciding the number of possible colluding users is a challenge.

Another signature scheme against pollution attacks is explained in [68] by MinJi Kim et. al. This paper emphasizes the destructive and exponentially increasing contaminating nature of pollution attacks in a peer-to-peer network. Further, they propose a signature scheme which enables the detection of contaminated packets on the fly. Authors propose a signature based on an orthogonal vector to the original packet vectors. This orthogonal vector is signed by the source using it's private key and distributed. Any node can verify the signature and get this orthogonal vector. Then, the verification of any linear combination of original vectors is possible using this scheme. Finding another vector which will be capable of breaking this scheme is as hard as a discrete logarithm problem by definition. This will also help to prevent multiple contaminations at a single benign node and thus ensure the maximum bandwidth efficiency for the transmission. This scheme has an overhead of approximately 6% if the signature is applied per file. The actual overhead is $6(m + l)/ml$ times the file size where m packets of l-dimensional vector space over the finite field are considered in a generation.

An identity-based signature scheme for network coding was proposed by Jiang *et al.* [69]. The proposed scheme features a dynamic identity-based authentication and signature scheme which also enables batch verification of packets. A multi-level binary authentication tree (M-BAT) is introduced to properly mitigate the corrupted packets. The signature scheme is based on the bilinear map and pseudo-identity. Finding a hash collision for the signature is as hard as computing discrete logarithm problem and the pseudo identity PID will be changed periodically which prevents signature forging attacks as well. This paper also performs an overhead comparison with [34], [61]. The proposed approach has similar overhead for signing the packets but has a reduced overhead for verification since the identity based signature scheme has eliminated the requirement of modular exponentiation and reduced the number of pairing operations required. With the batch verification the overhead is reduced further. They also analyzed the communicational cost and explained the fixed computational costs the scheme introduces. It is mentioned that the identity based cryptography does not require any certificates (125 bytes as per IEEE 1609.2 standard [70]) to be associated with the signature, but only a smaller identity information (44 bytes) along with a fixed 22 bytes of ECDSA signature similar to [34].

RIPPLE [71] was the first integrity scheme based on symmetric key cryptography to address tag pollution attacks and an arbitrary number of collusion between adversaries. The tag pollution attacks significantly reduce the network efficiency by altering the tags which will be checked in later levels of authentication. This will result in discarding of many packets when the altered tags are found which in turn reduces the network efficiency. RIPPLE utilizes MACs to provide security and use nested MACs to prevent tag pollution attack. Further, it uses the RIPPLE transmission protocol to provide time asymmetry for secure key transmission. In this protocol, the whole network is considered as a tree structure and the packets are transmitted in a wavelike fashion to each level. Further, the keys used to create MACs are transmitted by the source in a different time interval, for level by level. This ensures the authenticity of the packets and tags associated with them. The computational complexity of RIPPLE is lesser compared to most of the previous schemes, however, the new tags being created at each node and nested tags impose a significant communicational overhead. The total number of modular multiplications needed to verify L tags is $L \times (n+m+(L-1)/2)$ where n is the size of original packets and m is the number of packets in the generation. Further, to compute the tags for outgoing packets, each node has to perform $(L-1)w/2$ multiplications on average for a network with every node having same number of parents and w is the number of incoming edges. However, the communication overhead of RIPPLE scheme increases as it travel number of hopes. Further, the key sharing based on time asymmetry introduce accumulative delay in the network and increases with the levels (number of hopes) making the RIPPLE transmission protocol may not be suitable for practical applications requiring low latency wireless connections wireless networks with D2D communications over multiple hops.

Nuttapong [72] et.al. explains an extension of the NCS scheme proposed in [68] and presents a homomorphic signature scheme for network coding in the standard model. The authors utilize a dual encryption model for security than the random oracle model. They extend the NCS scheme by adding a compatibility check algorithm to ensure the randomness and uses a pseudorandom function to partially generalize the system so that the source doesn't have to wait for the complete file to start encoding. Instead of signing all the span of the subspace of vectors here signatures are defined over the vectors that are being transmitted at that particular moment. However, this additional computation to make the scheme homomorphic also imposes a computational overhead on the previous proposed scheme.

In 2011, MacSig [73] was proposed as an integrity scheme combining both symmetric key based MAC and public key based signatures to provide security against both data pollution and tag pollution attacks. The signature and MAC schemes proposed for MacSig are based on the concept of padding for orthogonality, where the source pads every packet with an extra symbol/tag such that the subspace spanned by this extra symbols/tags is orthogonal to a specific vector. This helps to verify the integrity of the packets received at a node by verifying whether the padded packets map it to zero. Authors explain the constructions of homomorphic subspace MACs (HSM) and homomorphic subspace Signatures (HSS) using the concept of padding for orthogonality. Further, they explain the double random key distribution to prevent the adversary from predetermining a combination of compromised nodes to pass the verification. In the proposed MacSig protocol, each receiver nodes are distributed with a random subset of keys by the source and the source itself choose random keys to create a number of tags for each generation (double random key distribution). A number of tags are created over the packets and a signature is created over these tags to prevent pollution attacks. While signing, the augmented coefficients are also considered to improve the security of the signature. Each node on the network could verify the signature using the public key of the source and then verify at least one MAC using the subset of keys it holds. This scheme has a better performance against previous cryptographic solutions in terms of computational and communicational overheads as well as prevents tag pollution attacks as well. The bandwidth overhead of MacSig was defined as $(l+1)/(m+n)+32l/|p|(m+n)$ where $|p|$ is the field size and l is number of tags and to verify a packet the node has to perform $(m+l+1)$ exponentiations and $(m+n+1)l$ multiplications. However, with the double random key distribution, the number of tags to be verified (proportionally number of multiplications) is reduced. Further, successful implementation of double random key distribution is challenging in the dense environment.

Catalano *et al.* [74] proposed two network coding signatures in the standard model in 2012. The first proposal was based on the q-Strong Diffie Hellman assumption proposed by Boneh and Boyen [75] and another one as an extension to their own previous work in [76], based on RSA assumption. The authors propose both their schemes without depending on the random oracle still achieving similar performances. The first proposal based on q-SDH achieves most efficiency among the already existing standard model integrity schemes. The computational overheads for signing and verifying the signature remains same as the overhead of [68], [72] but reduces the signature size to a function of the security parameter λ for eg.: 512 bits if the security parameter k = 128 bits of security and asymmetric pairings. However, the key size is not fixed as the previous schemes, but varies proportionally with the packet size and number of packets in a generation, $m + n$. The RSA assumption based signatures is an optimization of [76] by allowing lower exponents and computing over $mode(e)$ to restrict the vector coordinates from growing beyond limits. However, the schemes based on the standard model underperform slightly compared to the integrity schemes based on random oracle heuristic.

A TESLA-based homomorphic MAC scheme [77] was proposed for authentication in a P2P live streaming environment. This scheme uses the idea of loose time synchronization and delayed key sharing from the source to other nodes as proposed in the TESLA protocol for multicast authentication [78]. The homomorphic MAC forms an integral part of the scheme and used to verify the integrity of the packets. However, they have modified it to PMAC using a pseudorandom generator and a pseudorandom function to reduce the key size and computational overhead. Further, they use a test tag along with the MAC tag to ensure that the network coding processing is done properly with the help of the delayed key distribution from the server. This requires that every node has to buffer the received packet for an interval, but the high throughput can still be achieved by simultaneously transmitting multiple generations of packets. The computational overhead due to the proposed scheme includes a one time computation of $m + l$ PRF calls and one PRG call and $(n + 2m + l)(|P_N| + 1)$ multiplication over F_q per node per packet where $|P_N|$ is the size of the set of parent nodes for node N and l is the number of MAC tags. The communication overhead due to MAC tags and test tags combines to be $|P_N| \cdot (3l + 1)log_2 q$ bits.

Key Predistribution-based Tag Encoding (KEPTE) [79] was proposed in 2014. It is a hybrid cryptographic-based integrity scheme against pollution attacks in network coding. KEPTE utilizes different keys for creating tags at the source and to verify the tags on intermediate and receiving nodes. However, it differs from a signature scheme since it is not a public key cryptographic approach. KEPTE is a private predistributed key based scheme which encodes the packets with tags. It provides a number of keys to the source (to create tags) and a unique pair of the key to all other receiving nodes. These keys hold the mathematical relationship to verify the tags created using the keys distributed to the source by using the unique pair of keys held by other nodes. Thus it reduces the key storage overhead at the intermediate nodes as well provides better computational efficiency as well. Further, in 2016, [80] studies the KEPTE protocol and improvements in the key distribution and management scheme are proposed. KEPTE discuss the computational complexity for the initialization or the key distribution separately and then discuss the computations required for signing and verifying the tags. For the initializing process the overhead is in the order $(N^3 + N^2(m+n))$, where N is the number of tags. For signing the packet, source node has to perform computations in the order $(N(m+n))$ where for verification process at the receiving nodes has a complexity $O(m+n+N)$. Each packet has N tags attached to it which gives an overhead ratio of $N/m+n$. From the storage point of view, the source has to store N keys each of size equal to a data packet and the recipient nodes need to store two secret vectors each with $(N+m+n)$ $|log_2 P|$ bits.

A null space-based homomorphic MAC scheme [81] capable of detecting pollution attacks was introduced in 2016. This work focuses on using the null space properties to prevent the tags from getting corrupted. The integrity of the packet is ensured by using cryptographic tags as proposed in the previous schemes and then these tags are swapped with few of the symbols of the coded packet so that the adversary cannot distinguish between the coded packet and the tags. This makes it difficult to corrupt the tags and prevents tag pollution attacks with an attack probability which will depend on the probability of successfully guessing the swap. But the swapping of tags with the packet symbols is decided on a pseudorandom function derived swapping vector (swapping integer) which is shared between only the source and destination nodes. It also makes it difficult for the adversary to even verify if it succeeded in properly guessing the tag position or not. This scheme doesn't have any significant extra overhead than creating and communicating the tags compared to its peer schemes other than the requirement of sharing the secret swapping vector. Otherwise, the overhead due to creation of tags is same as that of KEPTE and the verification overhead is only $(m+n+N)$ because in this scheme only one tag is verified per receiving node.

Esfahani et. al. also proposed dual MAC-based security schemes to prevent pollution attacks in network coded environment in a series of works [82], [83]. Both data pollution attack and tag pollution attacks are addressed in the works. Initially, they proposed a dual HMAC scheme [82] which utilizes a set of MACs to ensure the integrity of the packet and another set of authentication codes called D-MACs calculated over the MACs to protect them from tag modification attacks. A c-cover free based key distribution system is in place to ensure pollution detection even if multiple nodes are compromised. This also ensures that in the worst case, a pollution attack will be detected at $c-1$ hopes later. However, the dual HMAC scheme can still be vulnerable to a dual-tag pollution attack with considerable probability. To address this issue, they also incorporated an idea from MacSig to their work and signs the tags attached to each packet. Thus an efficient HMAC scheme is proposed in [83] where MACs, D-MACs and a signature over them are combined to provide a secure network coded environment. The cover free based key distribution scheme helps to protect the system against a coalition of adversaries. It also reduces the computations required to verify the tags at intermediate and destination nodes. In the improved HMAC scheme, the verification process requires $l'+l$ exponentiations to verify the signature and $l \times (m+n+1)$ multiplications to verify MACs and $l'(l+1)$ multiplications to verify the D-MACs. The number of tags depends directly on the collision resistance value c and other security parameters. For the simulations they have considered three different values 27,42, and 54 where each tag is of size $|log_2\ p|$ bits. However, chances of malicious packets travelling some hopes still exist with this approach. Further, since not all tags are verified at each node, the communication overhead incurred by the protocol is not being utilized to its maximum in terms of security.

Table 1 gives a summary of the subsection.

B. INFORMATION THEORETIC APPROACHES

In a series of works [84], [85], S. Jaggi et.al proposes rate optimal, information theoretic based network codes. Authors try to address the issue of Byzantine adversaries trying to inject malicious packets to a multicast network coded system. They propose polynomial time algorithms to prevent a malicious node from injecting corrupted packets. In [85], different types of network and adversary models are studied and tested whether the proposed scheme is capable of achieving the optimal rate. The optimal rates for the networks are determined under different assumptions and adversary capabilities. Against the strongest, omniscient adversary, the proposed scheme achieves a rate of $C - 2z_0$ with encoding/decoding complexity in the order of nC^3 where C is the network capacity, n is the length of each packet and z_0 is the number of packets the adversary can inject. The error-correcting codes proposed by Jaggi. et.al were few of the initial works in the direction of information theoretic approaches against pollution attacks.

Secure Practical netwOrk Coding (SPOC) [86] was a lightweight security scheme based on the idea of locked coefficients attached to the packets. Vilela et.al. explain about this scheme as an extension of the shared secret model explained in [85]. However, they discard the use of a separate secure channel for sharing the secret by attaching a few locked coefficients to the native packets. Few of the coefficients generated at the source node will be encrypted with a secret key shared only between the source nodes and destination nodes (this needs to be done only once and expected to happen offline or before the beginning of communication process) and these are called locked coefficients and other coefficients which are not encrypted are called unlocked coefficients. Intermediates nodes operate on the received packets without any distinction between the coefficients. Thus when a packet reaches the sink node, then the unlocked coefficients will be used to decode the locked coefficients and then decrypt it with the pre-shared key. Then the decoding matrix is computed and the original packets can be decoded. The scheme ensures that decoding of original packets are not possible

TABLE 1. Summary of cryptographic approaches against pollution attacks.

Scheme	Type	Main characteristics	Security Assumption/ Hardness	Computational Complexity	Communication overhead
[34]	Homomorphic Signature	Elliptic curve based pairing	Discrete logarithm problem (DLP) and Diffie-Helman (D-H) problem	$(m+n)\times$ pairing and $(m+n+1)\times$ multiplications	Fixed bandwidth overhead which depends on the chosen prime numbers
[59], [60]	Homomorphic Hashes	A collaborative mechanism to reduce complexity	Cooperative security using hash functions	$(m+n)$ exponentiations	Throughput slightly higher than 10Mbps
[61], [62]	Homomorphic Signature	Enables batch verification	Integer factorisation (RSA based)	$(2+m+n)$ exponentiations	Fixed bandwidth overhead
[63], [64]	Homomorphic signature	NCS1: random oracle model and NCS2 : weaker DLP over linear subspace	NCS1: computational D-H in bilinear groups and NCS2: weaker DLP	$(m+n)$ exponentiations	lower bound on the signature size as $mlog_2q$
[65]	Homomorphic MAC	Initial work on polynomial time homomorphic Message Authentication Codes	Probabilistic Polynomial Time (PPT) algorithm (with each tag provides security level of $1/q$)	$m+n$ multiplications	Number of tags depend on the collusion resistance coefficient, c
[68], [72]	Homomorphic Signature	Sign an orthogonal vector to detect multiple contaminations in a single node	Discrete Logarithm Problem	$(m+n)$ exponentiations	$6(m+n)/mn$ approximately 6%
[69]	Identity based Signature	Multi-level BAT to mitigate corrupted packets	computational Diffie Helman problem on elliptic curves	2 pairing, $(m+n)$ multiplications and 1 exponentiation	66 bytes
[71]	Homomorphic MACs	Integration of cryptographic and information theoretic approaches	PPT algorithm with time specified transmission protocol	$N\times(n+m+(N-1)/2)$ to verify N tags	increases as it travel number of hopes
[73]	Hybrid	Integration of homomorphic signatures with MACs to prevent tag pollution attacks	MAC using Orthogonality principle (Security level of $1/q$ per tag), DLP for signature	$(m+N+1)$ exponentiations and $(m+n+1)N$ multiplications	$(N+1)/(m+n)+32N/\|p\|(m+n)$
[74]	Homomorphic signature	Two schemes without using random oracle model	q-strong DH and RSA	$(m+n)$ exponentiations	$f(\lambda)$, where λ is the security parameter
[77]	Homomorphic MACs	Loose time synchronization and delayed key sharing as proposed in TESLA	PPT algorithm (with each tag provides security level of $1/q$)	$(n+2m+l)(\|P_N\|$+1) multiplication over F_q	$\|P_N\|$ ·(3l+1)log$_2q$ bits
[79]	Homomorphic MACs	Specific key distribution algorithm to ensure security	PPT algorithm (with each tag provides security level of $1/q$)	Multiplications in $O(m+n+N)$ for verification	$N\times log_2q$ bits
[81]	Homomorphic MACs	A null space-based security scheme in which tags are swapped with symbols from the packet	PPT algorithm (with each tag provides security level of $1/q$) and additional security of $1/m$ by swapping	$(m+n+N)$ multiplications for verification	$N\times log_2q$ bits
[82], [83]	Hybrid	Both homomorphic signatures and MACs along with specific key distribution scheme	MAC using Orthogonality principle (Security level of $1/q$ per tag), discrete logarithm problem for signature	N'+N exponentiations for the signature, $N\times(m+n+1)$ multiplications for MACs and N'(N+1) multiplications for D-MACs	number of tags N and N'depends directly on the collision resistance value c

without decrypted values of the locked coefficients and thus ensures the integrity of packets received at the sink node. They eliminated the separate secret sharing channel required to carry the hash for each generation at the expense of a one-time pre-shared key which can be performed even offline with an extra computational overhead of encrypting some of the coefficients which in the third order of generation size. However, like most of the other non-cryptographic approach, SPOC also detects the pollution attacks only at the sink node.

DART and EDART [55] schemes are time asymmetric based integrity schemes against pollution attack. They use time asymmetric based checksums to ensure the received packets are genuine. It also enables detection of corrupted packets at intermediate nodes which is very rare in non-cryptographic protocols against pollution attacks. In DART protocol, checksums are created at the source on the generation of packets using efficiently computed random linear transformations and attached with time stamps. Every forward node in the system will verify the received packets only if they also receive a checksum confirming the authenticity of already received packets. Only verified packets are used for encoding at a forwarding node so that polluted packets won't be propagated further. However, this accounts for a delay at each node for verifying a sufficient number of packets before forwarding. EDART is proposed to reduce this delay with an optimistic forwarding. In EDART scheme, the nodes farther from the attacker will forward the packets without waiting for verification where nodes near to attacker will verify the packets with checksum before forwarding. Every node will initially start with forwarding mode and any node detecting a mismatch in checksum will shift to verify mode. Nodes following a verify node will decide the delay and forward timings depending on the security parameters and pollution frequency detected. This scheme was computationally much simpler compared to any cryptographic scheme but not suitable for the delay-intolerant systems. The slight computational overhead of 5 signatures per second for the source node is due to creation of checksums and these checksums also account for a 18 kbps bandwidth overhead per forwarder node.

SpaceMac [87] by Anh Le and Athina Markopoulou consider an expanding subspace and extend homomac [65] to prevent both data pollution and tag pollution attack. Further, they enforce cooperative security along with a controller to exactly locating the malicious node making it a hybrid scheme capable of not only detecting pollution attacks, but also locating the adversary nodes. Spacemac considers the network with the parent-child cooperation to enable security. Any node N will be enforced to create its sending packets only from the packets it received from its parent nodes. This is enforced by the cooperation of parents and children of N. Since the parent nodes are able to sign over the subspace from which a node can create the packets, Spacemac can be applied to extending subspaces. Since every node will check the tags attached to the received packets, tag pollution attack won't have adversary effect on Spacemac because packets with corrupted tags won't travel any further in the network. Further, with a central controller and cooperation of all the nodes, Spacemac identifies the exact location of the attacker also. However, the central controller needs to know the complete network topology which may not be the case with a wireless network. Also, the active collusion of adjacent adversary nodes can win over the spacemac security scheme and it provides vulnerability if a particular area is compromised by a strong adversary. The combined detection and locating scheme of SpaceMac has a bandwidth overhead of $(3 + \lambda)|log_2\ P|$ bits as tags attached to the packets and computational cost of $(3+d+\lambda)(n+2m)+w$ per packet per receiving node, where λ is the total number of tags attached to the packet, d is the number of tags verified by that node and w is the average number of packets.

VI. COMPARISON OF DIFFERENT STATE OF THE ART APPROACHES IN INTEGRITY SCHEMES

Pollution attacks in network coding enabled environment need to be addressed very carefully. It is necessary to detect the attack in the earliest possible point. Both information theoretic and cryptographic methods are proposed to solve this problem and many of them are analyzed above. This section provides an analysis of these schemes in general and discusses the pros and cons of different approaches. It also discusses the differences between homomorphic signature-based and homomorphic MAC based schemes separately. This analysis and comparison are made with the concern of using network coding for 5G deployments in a mobile small cell environment [9]. Thus the main points of analysis include the latency, energy efficiency, and bandwidth optimization.

Information theoretic approaches can be considered as the least complex solutions against pollution attacks. These approaches do not require complex computations but the stress on specific characteristics of the system to ensure security. Most of these schemes allow detection of polluted packets only in the sink node. It also requires a secret shared between only the source and destination most of the time. Another approach of information theoretic solutions depends on time asymmetry. In such protocols, detection of polluted packets depends on the time latency and extra symbols attached with the message. This approach is incurring latency in communication. Further, the synchronization of the whole system becomes an inherent criterion for these schemes. Another stream in which non-cryptographic approaches against pollution attacks is based on the cooperation of neighbouring nodes. However, these approaches are never been completely non-cryptographic. It is mostly like a hybrid approach, where some characteristics and cooperation of the network is used to enforce the cryptographic techniques in an efficient way. Approaches like RIPPLE [71] and Spacemac [87] are examples for this. In essence, we can say that even though non-cryptographic approaches to prevent pollution attacks are computationally efficient, they require specific conditions to be satisfied and mostly inefficient in timely detection of polluted packets.

When it comes to cryptographic approaches, they can be divided broadly into homomorphic signature-based approach and homomorphic message authentication code based approach. The first type depends on asymmetric keys while the latter one depends on symmetric keys. With the basic principle of network coding, the packets may not be the same as sent by the source but a linear combination of the original packets, any computations over the packet to ensure integrity will also require verification of the linear combination of original data. Thus, homomorphic schemes are essential for cryptographic integrity schemes in network coding. Further, this forces to have more computations and more extra bits added to the packets for security. However, depending on the key distribution schemes, pollution attacks can be more efficiently stopped by these schemes at the nearest benign node. Detecting polluted packet at the nearest benign node is of utmost importance. Otherwise, it will degrade the performance of the system by affecting more packets and flows down its way. In terms of latency, cryptographic techniques don't impose any latency as part of its security mechanism; but, the computations will take some time. However, this computational delay depends on the computational capabilities of devices and small compared to the inherent delay for security purpose in the information theoretic based approaches. Still, the key distribution can cause a longer initial delay in setting up the system. Considering the bandwidth requirements, cryptography-based integrity schemes always depend on some cryptographic functions performed over the message and require the proper communication of these computed values as well. This overhead is unavoidable, but need to be kept to a minimum.

Let's analyze the difference between homomorphic signature and homomorphic MAC based schemes now. As already mentioned the public key based homomorphic signature schemes and secret key based homomorphic MAC schemes are two different directions of cryptographic integrity schemes in network coding. Both approaches have their own pros and cons. In most cases, the signature based schemes require more costly computation for the verification compared to MAC-based schemes. On the other hand, having a shared secret key between the source and receiving nodes requires more effort and efficient key distribution protocol compared to the public key management. Further, the MAC based approaches will have a larger bandwidth overhead due to the larger number of bits required to ensure security in the system. MAC-based approaches are also susceptible to tag pollution attacks in network coding. One solution to this problem is multiple levels of tags. However, it may not be suitable because it is again susceptible and increase the computational and bandwidth overhead considerably. Another direction of research to protect network coded networks from tag pollution attack leads to the hybrid cryptographic schemes. In such cases, eg: MacSig [73], a combination of homomorphic MACs and homomorphic signatures are both used to ensure security. This approach finds a better trade-off between the computational complexity and communication complexity without providing any security flaw. Thus finding efficient cryptographic integrity schemes could lead to secure network coding enabled mobile small cells.

VII. SECURE NETWORK CODING ENABLED MOBILE SMALL CELLS

Proceeding further from the state of the art integrity schemes, we are looking at the integrity schemes for network coding enabled mobile small cells, shown in fig. 7. Since the whole idea of a future network involves low latency high resilient mobile network, the cryptographic approaches look like a better match to the system. Nevertheless, it should also consider the energy efficiency of the schemes. Thus we are trying to extend and modify the existing approaches to be more efficient and secure using the characteristics of the proposed architecture. However, the existing schemes cannot be directly adapted to the mobile small cell environment. Most of them are studied in less dynamic network topology and the complexity of these schemes increases exponentially with scaling. Further, most of the schemes require a pre-installation phase of key sharing which is hard to achieve in the highly dense and dynamic network conditions. In the existing homomorphic MAC based approaches, the security against a coalition of malicious users or a set of compromised nodes in an area is always depending on this key generation schemes. On the other hand, the future networks provide some support to the integrity schemes and help to reduce computational complexity and bandwidth overhead of the existing schemes. BBU pools and SDN based small cell management [88], [89] are such domains which can be used to develop more secure and suitable integrity schemes for 5G mobile small cells.

We propose two initial integrity schemes which suit with the proposed system architecture of 5G small cells, shown in fig 7. The first scheme is utilizing a central unit to ensure secure sharing of tags while the second proposal considers a more distributed network environment. Both the schemes are following the homomorphic MACs for ensuring the integrity of the packets in transition and utilize the network architecture to ensure the MACs are securely shared with all the nodes. Further, these schemes ensure inter small cell communications inside a macro cell happens smoothly by making the information available over the macro cell. These two schemes are explained and the security schemes are analyzed in the following subsections.

A. CENTRALIZED TAG SHARING APPROACH FOR SECURE NC AWARE SMALL CELLS

The first proposal [90] discusses a secure integrity scheme for the small cells where the centralized SDN controller is used for tag sharing. This scheme uses homomorphic MACs to ensure the integrity of the packets and use a central unit as a secure way of sharing these tags with all the participating nodes. Even though the proposed scheme follows Homomac [65] for tag creations, it also tries to reduce the key size by creating the tags on native packets instead of augmented

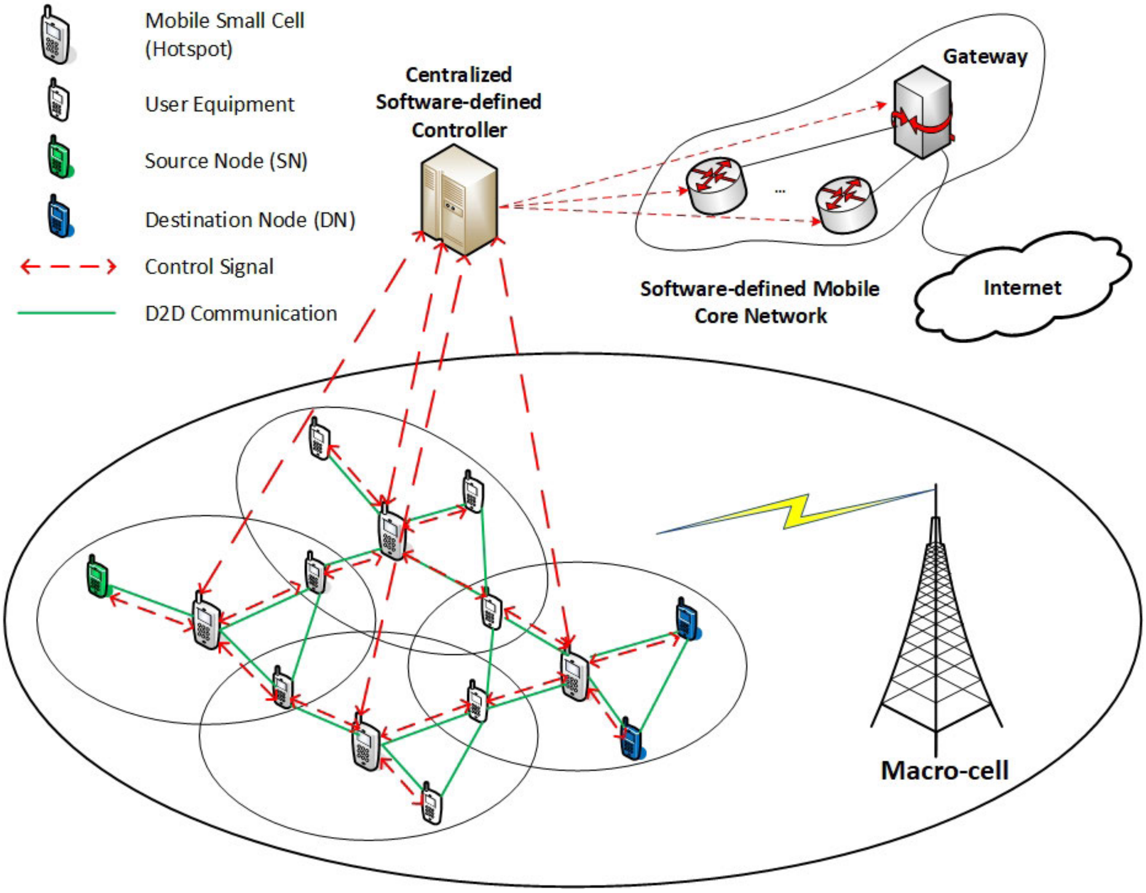

FIGURE 7. Mobile small cells scenario.

packets. Further, this scheme also tries to ensure the network is secure against pollution attacks and colluding malicious users. The proposed scheme is explained below in four steps.

1) Initialization: The MACs or tags are created using symmetric key-based cryptography. The source node and the receiver nodes share the same set of secret keys to ensure tag creation and verification respectively. These secret keys need to be securely shared between all the nodes. A Key Distribution Center (KDC) is usually present in the system as we discussed earlier. However, this process can be done prior to the starting of actual communication. A set of keys, K_s, where each key K_i consists of $n + 1$ symbols will be pre-distributed. This key size is depending solely on the symbol size of the native packets, not the augmented packets, which makes it smaller compared to other existing schemes like HMAC and MacSig. Further, we reduce the requirement of any particular key sharing scheme and allow all the participating nodes to have the same set of symmetric keys to create and verify the tags. Additionally, each node will have its own public-private key pair to create digital signatures whenever necessary. The public keys will be distributed all over the network with the help of KDC.

2) Tag generation: Tag generation in this scheme is based on the homomorphic tag generation scheme explained in Homomac [65]. However, the tags are generated directly on the native packets by the source node, even before the augmentation process starts. This reduces the key size and also the number of computations required. A tag is created as per the equation

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

Since the packets are usually transmitted as generations, lets say of size m, each generation will have an overhead of $L \times m$ number of tags. These tags will be also sent to the central unit via a secure channel by the source node along with its signature and the generation number. The central unit stores these tags and passes it to the other nodes whenever requested. This helps for the additional verification that ensures the tags are not modified during the transition.

3) Verification: The participating nodes verifies the integrity of the packets using the verification algorithm 1 whenever it receives a packet through the communication channel. This verification process happens in two

steps. As soon as a generation of packets is received, the verifier node will check for the corresponding entry of tags in the central unit using the source ID and generation number. The authenticity of these entries can be verified using the signature associated with it. Then these tags will be compared with the tags received along with the packets. However, the tags in the central unit are without any encoding but the tags received would have undergone the encoding process. To perform this comparison, the tags retrieved from the central unit will be multiplied by the coefficients matrix associated with the received packets. This check will ensure that the tags are not modified during the transition (ie, no tag pollution attack) other than the network coding operations. Afterwards, the node can verify the integrity of packets using the keys available with it, similar to the tag verification process in any of the homomorphic MAC based scheme. To perform this, the verifier node will try to recreate the tags over the received packet using the secret keys it holds. This step ensures that there is no data pollution attack. Once the generation passes the verification process successfully, at the intermediate node, it will be re-encoded and transmitted further or decoded at the destination node. Otherwise, a pollution attack is detected and the packets will be dropped.

4) Re-encoding: The re-encoding process in this scheme is very much similar to the general RLNC re-encoding. The intermediate nodes do not recreate the tags, but simply consider them as part of the packet and re encode as any other packet symbol. Since the tags are generated and attached to the native packets at the source node, the intermediate nodes do not differentiate tags from other symbols of the normal packet. This also reduces the computational complexity at the intermediate nodes. Thus the re-encoding of packets in this scheme is simply the multiplication of the verified packets with the locally generated coefficients.

Algorithm 1 Verification Algorithm

```
Data: Received packet $\mathbf{C}_i$, $\mathbf{L}$ tags corresponding to $\mathbf{C}_i$
      retrieved from the central unit, Key set $\mathbf{K}_s$
Result: 1 if verification is successful and 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 central unit with the
corresponding coefficients
Step 3:
Compare the tags with those appear in the received
codeword.
if they don't 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 central unit then
    1 ⟵ Return
else
    0 ⟵ Return
```

1) SECURITY ANALYSIS

This section analyses how the proposed scheme ensures protection against pollution attacks using the central controller. The security scheme is analyzed over a butterfly network in a small cell environment, supported by a central controller, like an SDN system, as shown in fig. 8. Before proceeding to the security analysis, it is necessary to define the capabilities of the adversary node. In the scenario described in this paper, only the intermediate nodes are considered susceptible to attacks. The source nodes are considered as trusted and secure. Also, the key distribution scheme is considered as secure, especially the asymmetric keys used for signing the entries to the central unit is kept secure and not shared by the source nodes. However, when an attacker compromises an intermediate node, it can have full control over the resources available to the compromised node. Thus if the attacker compromises an intermediate node, it can access the whole key set available to the node as well as decode and analyze the original packets and tags attached to them given that enough number of packets are received at the node. Thus the adversary has strong knowledge over the messaging scheme. Further, we consider a situation in which the attacker could compromise more than one node in a neighbourhood and perform a coordinated attack. However, the direct connection from the central unit to each node in the network nullify any additional advantages achieved by such mass compromising of nodes since the security systems at the immediate benign node will be able to detect and discard polluted packets.

1) Data Pollution Attacks: The adversary tries to modify the content of the packet and forward the message to the neighbouring nodes. This pollutes the corrupted packet instantly and with further alterations pollutes more and more genuine packets. This points to the necessity of finding out the corrupted packet at the earliest possible instant and prevent it from mixing with other benign packets. In our scheme, a two-level verification of tags ensures that the data pollution attacks are detected efficiently at the immediate genuine node receiving a polluted packet. Since the receiving node already has the key set used to create the tags, it can check whether the tags are genuine to the corresponding message part in the received codeword. However, since the adversary

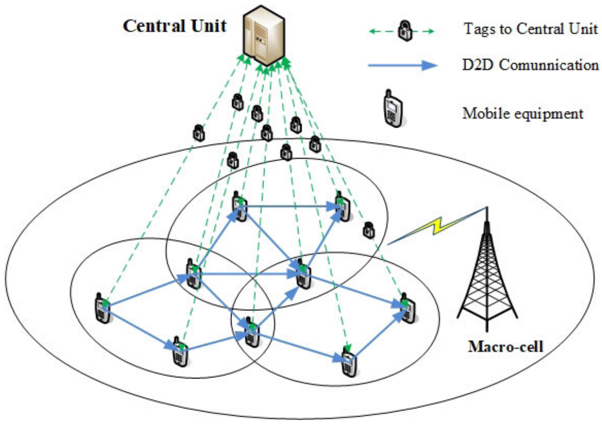

FIGURE 8. Simplified architecture for security scheme.

also has the keyset available from the compromised node, it may have forged the tag for the corrupted message and attach it to the packet. Thus a strong adversary can pass the first verification. However, the second level of verification is matching the tags received in the code word with the corresponding tags retrieved from the central controller. If the adversary has to pass this verification, it needs to forge a corrupted packet which will give exactly the same tag as the original packet. That is the same as finding another symbol in the symbol space of the original packet such that the MAC generation will result in the same tag for both the corrupted and original packets. This can be considered as a probability of $1/q$, where q is the field size. Further, if there are L tags that will be checked, then it needs to satisfy all these tags and then the probability of creating a corrupted packet that will pass the verification test is $1/q^L$. In practical cases, $q = 2^8$ and $L = 8$ give a very satisfactory level of protection against the data pollution attacks.

2) Tag Pollution Attacks: Tag Pollution attacks are a serious problem faced by the homomorphic hash/MAC based security schemes in network coded environment. In such cases, the tags created and attached to the original benign packets are altered by the adversary intermediate nodes. Then these packets will travel till it will detect an altered tag and discarded. Such attacks create two serious issues; network resource underutilization and dropping of genuine packets. Thus it is necessary to detect the tag pollution attack at the immediate neighbouring benign node and further process the transaction of genuine packets. This is ensured in our scheme using the secure communication of tags to the central unit and its a comparison. The authenticity of the entries in the central unit is verified using the signature attached to it. Comparing the corresponding tags in the received packet with those retrieved from the central unit results in the detection of tag pollution attack. In case of the detection of a tag pollution attack, that node can check whether the content is still the same by creating tags for the packet and comparing with the tags retrieved from the central unit and proceed with the communication after marking a warning against the malicious node. By this way, the network resource wastage and unnecessary dropping of genuine packets can be tackled. It is ensured that this detection of pollution attacks happen at the immediate benign node in the system after adversary. However, the reporting process is efficient against only the last adversary node, even if there is more than one compromised node.

However, this approach requires a central controller to facilitate the scheme. Additionally, secure communication between this central controller and the nodes so that the tags are not changed during this transition. Even though the network of small cells may be supported by an SDN or central unit in the cloud, it may be a better idea to look for a more distributed approach for ensuring the secure communication of tags. From this perspective, a distributed blockchain like architecture for tag sharing is proposed. This is an extension of the central unit based approach which also gives an easily scalable system of secure network coding enabled mobile small cells.

B. DISTRIBUTED BLOCKCHAIN BASED APPROACH FOR SECURE NC AWARE SMALL CELLS

A distributed ledger, like a blockchain, can be used to store and share the tags with the participating nodes to avoid the issues like central point of vulnerability and requirement of a secure communication channel for all nodes with the central unit. In [91], a blockchain based integrity scheme for 5G deployments is proposed. This scheme is more elaborated and extended in [92] to adapt to the small cell environment and handle the overhead to maintain the blockchain in an efficient way. Source node will create a candidate block that consist of the tags it generated along with the source ID and generation number. These candidate block will be stored in the blockchain once it is verified and available to other interested participants to access afterwards. This scheme proposes bighchainDB [93], a blockchain like distributed database to be used as a means for sharing the tags in the network. BigchainDB provides a blockchain like secure database, distributed over different nodes. It also provides a very fast and energy efficient block verification scheme which makes it suitable for the use in 5G related applications. Further, the data stored in bigchainDB can be accessed using the metadata, in our case source ID and generation number, by query service. They use a byzantine fault tolerance algorithm based on proof of stake concept to verify the blocks which makes it energy efficient and fast. The overhead of this verification process is distributed over the small cell heads [92] and thus avoid the requirement of signatures by the source node, reducing the computational complexity at the source node. This proposed architecture is shown in fig 9.

Adversary Model: The vulnerable points in the network are the intermediate nodes. An adversary can have control over

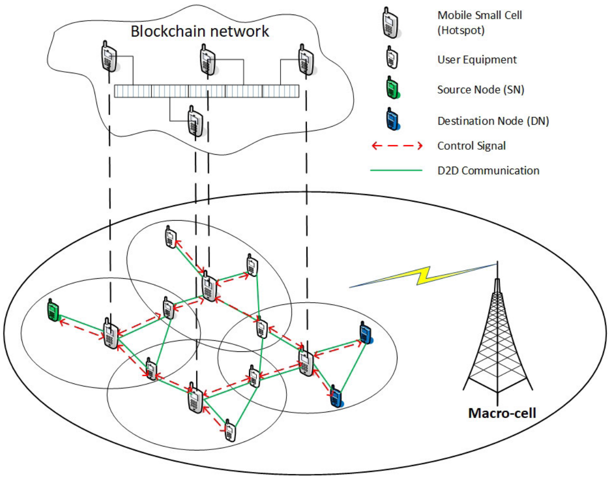

FIGURE 9. Simplified structure of a butterfly network employing blockchain based security scheme.

a compromised intermediate node. We expect the adversary can completely utilize the abilities of the 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 it can also view the original tags of the native message. As we consider it receives enough number of codewords from a generation before transmitting, it can also decode and see the native message. Thus the adversary has a strong knowledge.

1) SECURITY ANALYSIS

Here we discuss how our proposed scheme is secure against pollution attacks. Our security scheme is experimented against both data pollution attacks and tag pollution attacks.

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 blockchain and it should be matched with the tags created from the corresponding packet part in the message. Since adversary has 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 the 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 packet's content which will again satisfy the MAC algorithm with the modified packet, i.e, $MAC(P_i, K_s) = MAC(P_i', K_s)$. 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 equals $q = 2^8$. 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 secure against the data pollution attack with a probability of $1/q^L$ chance of vulnerable to data pollution attack. It shows we are achieving much better security compared to MacSig [73] and Dual HMAC [83] at the expense of a lesser number of keys. 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.
2) Tag pollution attacks: The security of our scheme against tag pollution attack is dependent on the concept of blockchain to securely compare with the received tags with providing the ability to create and testing the tags for received message to the node 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 if there is tag pollution or not. If the tags in the received packet do not match with the tags received, but matches with the original tags received from the blockchain, a tag pollution attack is detected. However, in such cases, a benign node can create a warning in the system against the malicious node and continue with the recoding since the message matched with the original tag.

This integrity scheme uses homomorphic MACs to protect the network from pollution attacks. It also ensures that any attempt to pollute the data in transition or to reduce the efficiency of the network by tag pollution attack is detected at the earliest genuine node. The performance analysis of this scheme is performed for computational complexity, communicational overhead and latency induced by the integrity scheme. The computational complexity of the proposed scheme is has mainly two parts. The first one is due to the creation and verification of the homomorphic MACs by the source and other nodes while the other part of complexity arises from the block verification process of the blockchain. However, as shown in the architecture, the blocks are verified by the small cell heads and distributed among them [92], this block verification overhead is not being considered in the analysis of performance analysis of participating nodes. However, the authors acknowledge that this verification process is based on PoS based byzantine fault tolerance algorithm which does not require a hard and exhaustive computation of hashes and this overhead is not being considered because it may not be incurred by any participating node directly, but only by the small cell head which is expected to be a computationally stronger participant in that small cell. An extensive study of the blockchain incurred overheads is expected to be a part of the future works

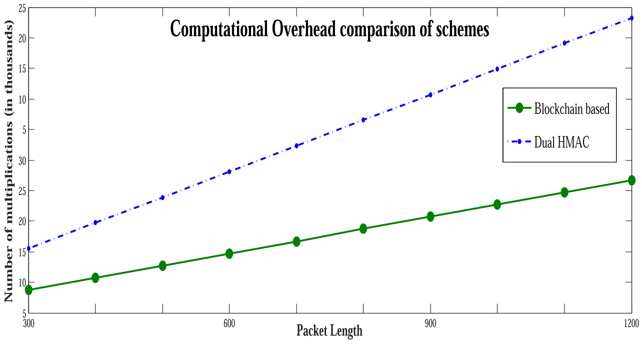

FIGURE 10. Computational overhead for different schemes.

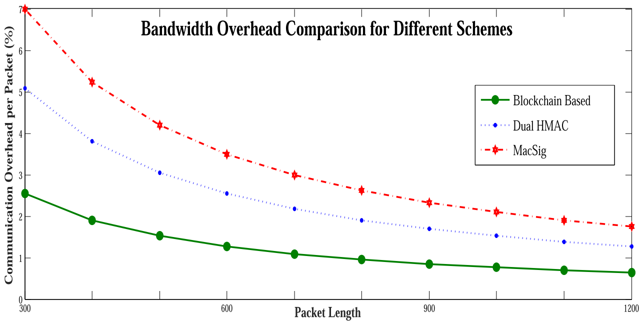

FIGURE 11. Communication overhead for different schemes.

proposed in this paper. On the other hand, the computational overhead due to tag creation and verification in this proposed scheme is less compared to the some of the recent and well known integrity schemes discussed in previous section [73], [83]. The computational overhead is measured in terms of the number of required finite field multiplications. The tags are being created over the native packets of length n, and $L(n+1)$ multiplications are required to create or verify L tags. Thus the overhead due to the tag creation and verification over each participating node is $L(n+1)$, a function of only the number of tags used and the original packet size. All participating nodes can verify all the tags attached to the packets and ensure that the probability of an adversary breaking the integrity scheme is limited to a maximum of $1/q^L$ where q is the field size.

The bandwidth overhead of the integrity scheme is based solely on the number of tags. Each tag adds $|log_2\ q|$ bits. Further, these tags are added to the blockchain and retrieved by every node for verification. This makes the bandwidth overhead over the communication channel as $L \times |log_2\ q|$ bits and the overhead due to communications with the blockchain is approximately $L \times |log_2\ q|$ bits per node. Further, the key storage space required by the proposed scheme is comparatively smaller since we do not require a large key sets at any node. Also the size of a single key is reduced to n + 1 where n is the size of the original packet. The proposed schemes doesn't depend on the neighbourhood of any node in deciding it's security and similarly avoid the requirement for large key sets to be stored in the nodes. A detailed performance evaluation and comparative study of these overheads are mentioned in [91]. These improvements in the integrity schemes make it more suitable for the 5G small cell environment. Also it helps in smooth scaling of the system to a dense small cell network. This work marks the initial step to build an RLNC based secure 5G network.

The blockchain verification incurs some latency to the system. Current simulations using bigchainDB requires a minimum time gap of one second between validation of blocks. However, multiple candidate blocks that are produced during this time period, called collection period, will be validated together. Once the blocks are verified then it will be available for all other nodes to access. Thus the latency will be only applicable to the first hope and it has the upperbound of one second. However, different blockchain based architectures [94], [95] can be studied and analysed to achieve better results. It is also planned to extend the work to address more security challenges in the network coded systems. Further improvements in the underlying blockchain concept to reduce the complexity and overhead due to block validation is also planned as an extension to the work.

VIII. CONCLUSION

The network coding enabled mobile small cells can potentially be an answer to the ultra-reliable, high throughput requirements that the 5G world requires. However, to utilize the potential of this system completely, it needs to be a secure system against internal and external malicious users. Even though network coding provides inherent weak security against some of the common threats, it also comes with specific and new challenges like pollution attacks. Due to the characteristics of network coding, mixing of packets on the fly, it also requires different integrity schemes than the ones being used widely now. Basic encryption and decryption schemes will not work with the network coding environment since the packets undergo alterations on the go. From a cryptographic point of view, homomorphic schemes could address this scenario through homomorphic message authentication codes or homomorphic signature schemes. However, it also needs to be supported by proper key distribution schemes. This paper discusses different existing integrity schemes for network coding secure against pollution attacks. It studies the major classification of the schemes and compares the advantages and disadvantages of them considering the environment of a wireless mobile network. We understand that the scalability of the previous integrity schemes are limited and may not be suitable for the dense networks that will form the future wireless environment. Thus specific studies are done on the two major proposals for a secure network coding enabled mobile small cell environment. These schemes based on a central controller or a distributed blockchain like ledger could provide better performance in the cooperated small cell system. Further, the computational and communication overheads are reduced considerably on the expense of using the available resources as the cloud-based central controller in [90] and at the expense of a lightweight ethereum based blockchain ledger in [91]. However, these studies are still going through the validation phase. It needs to be verified and validated against different 5G use cases to ensure the theoretical advantage also accounts for similar results in a practical scenario. As soon as the security challenges are addressed, the network coded cooperation of small cells could be playing an integral role in meeting the stringent throughput requirements that the 5G paradigm demands, securely and efficiently.

REFERENCES

[1] H. Jung, "Cisco visual networking index: Global mobile data traffic forecast update 2010–2015," Cisco, San Jose, CA, USA, Tech. Rep. 520862, 2011.

[2] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[3] G. Intelligence, "Understanding 5G: Perspectives on future technological advancements in mobile," GSMA Intell., London, U.K., White Paper 201401, 2014, pp. 1–26.

[4] M. Series, *IMT Vision-Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, document Rec. ITU 0-2083, 2015.

[5] C. Tselios and G. Tsolis, "A survey on software tools and architectures for deploying multimedia-aware cloud applications," in *Algorithmic Aspects Cloud Computing* (Lecture Notes in Computer Science), vol. 9511. Springer, 2016, pp. 168–180.

[6] G. Bianchi, E. Biton, N. Blefari-Melazzi, I. Borges, L. Chiaraviglio, P. de la Cruz Ramos, P. Eardley, F. Fontes, M. J. McGrath, L. Natarianni, D. Niculescu, C. Parada, M. Popovici, V. Riccobene, S. Salsano, B. Sayadi, J. Thomson, C. Tselios, and G. Tsolis, "Superfluidity: A flexible functional architecture for 5G networks," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 9, pp. 1178–1186, Jul. 2016.

[7] E. Hossain and M. Hasan, "5G cellular: Key enabling technologies and research challenges," 2015, *arXiv:1503.00674*. [Online]. Available: http://arxiv.org/abs/1503.00674

[8] 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 *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6.

[9] 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 *Proc. Internet Technol. Appl. (ITA)*, Sep. 2017, pp. 329–333.

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

[11] M. Celebiler and G. Stette, "On increasing the down-link capacity of a regenerative satellite repeater in point-to-point communications," *Proc. IEEE*, vol. 66, no. 1, pp. 98–100, Jan. 1978.

[12] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[13] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, Jan. 2006.

[14] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," *NetCod*, vol. 7, Apr. 2005, p. 104.

[15] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, 2003, vol. 41. no. 1, pp. 11–20.

[16] P. Georgakopoulos, T. Akhtar, I. Politis, C. Tselios, E. Markakis, and S. Kotsopoulos, "Coordination multipoint enabled small cells for Coalition-Game-Based radio resource management," *IEEE Netw.*, vol. 33, no. 4, pp. 63–69, Jul. 2019.

[17] T. Akhtar, I. Politis, P. Georgakopoulos, and S. Kotsopoulos, "Efficient radio resource management scheme in cooperative network using coalition game," in *Proc. IEEE 24th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw. (CAMAD)*, Sep. 2019, pp. 1–6.

[18] P. Georgakopoulos, T. Akhtar, and S. Kotsopoulos, "On game theory-based coordination schemes for mobile small cells," in *Proc. IEEE 24th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw. (CAMAD)*, Sep. 2019, pp. 1–5.

[19] J. Dong, R. Curtmola, R. Sethi, and C. Nita-Rotaru, "Toward secure network coding in wireless networks: Threats and challenges," in *Proc. 4th Workshop Secure Netw. Protocols*, Oct. 2008, pp. 33–38.

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

[21] R. Parsamehr, G. Mantas, A. Radwan, J. Rodriguez, and J.-F. Martínez, "Security threats in network coding-enabled mobile small cells," in *Proc. Int. Conf. Broadband Commun., Netw. Syst.* Springer, 2018, pp. 337–346.

[22] L. Lima, J. P. Vilela, P. F. Oliveira, and J. Barros, "Network coding security: Attacks and countermeasures," 2008, *arXiv:0809.1366*. [Online]. Available: http://arxiv.org/abs/0809.1366

[23] S. K. D. K. W. Hu and H. R. M. Médard, "The importance of being opportunistic: Practical network coding for wireless environments," *Newslett. ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, 2006.

[24] M. Hundebøll, J. Ledet-Pedersen, J. Heide, M. V. Pedersen, S. A. Rein, and F. H. P. Fitzek, "CATWOMAN: Implementation and performance evaluation of IEEE 802.11 based multi-hop networks using network coding," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2012, pp. 1–5.

[25] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2007, pp. 1028–1036.

[26] A. Toledo and X. Wang, "Efficient multipath in sensor networks using diffusion and network coding," in *Proc. 40th Annu. Conf. Inf. Sci. Syst.*, Mar. 2006, pp. 87–92.

[27] 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 Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[28] X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 622–634, Jun. 2009.

[29] S. Chachulski, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM SIGCOMM*, Aug. 2007, pp. 169–180.

[30] X. Zhang and B. Li, "Dice: A game theoretic framework for wireless multipath network coding," in *Proc. 9th ACM Int. Symp. Mobile ad hoc Netw. Comput.*, 2008, pp. 293–302.

[31] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2006, vol. 36, no. 4, pp. 243–254.

[32] J. Le, J. C. Lui, and D.-M. Chiu, "DCAR: Distributed coding-aware routing in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 4, pp. 596–608, Sep. 2009.

[33] S. Das, Y. Wu, R. Chandra, and Y. C. Hu, "Context-based routing: Techniques, applications and experience," in *Proc. 5th USENIX Symp. Networked Syst. Des. Implement.*, 2008, pp. 379–392.

[34] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proc. 40th Annu. Conf. Inf. Sci. Syst.*, 2006, pp. 857–863.

[35] V. Adat and B. B. Gupta, "Security in Internet of Things: Issues, challenges, taxonomy, and architecture," *Telecommun. Syst.*, vol. 67, no. 3, pp. 423–441, Jun. 2017.

[36] M. De Ree, G. Mantas, A. Radwan, S. Mumtaz, J. Rodriguez, and I. E. Otung, "Key management for beyond 5G mobile small cells: A survey," *IEEE Access*, vol. 7, pp. 59200–59236, 2019.

[37] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun./Jul. 2002, p. 323.

[38] S. Jaggi, M. Langberg, T. Ho, and M. Effros, "Correction of adversarial errors in networks," in *Proc. Int. Symp. Inf. Theory (ISIT)*, 2005, pp. 1455–1459.

[39] N. Cai and R. W. Yeung, "Secure network coding on a wiretap network," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 424–435, Jan. 2011.

[40] S. El Rouayheb, E. Soljanin, and A. Sprintson, "Secure network coding for wiretap networks of type II," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1361–1371, Mar. 2012.

[41] K. Harada and H. Yamamoto, "Strongly secure linear network coding," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 91, no. 10, pp. 2720–2728, 2008.

[42] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *Proc. 28th Conf. Comput. Commun. (INFOCOM)*, Apr. 2009, pp. 2213–2221.

[43] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. S. Shen, "Network coding based privacy preservation against traffic analysis in multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 3, pp. 834–843, Mar. 2011.

[44] H. Sousa-Pinto, D. E. Lucani, and J. Barros, "Hide and code: Session anonymity in wireless line networks with coded packets," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2012, pp. 262–268.

[45] L. Lima, J. Barros, and R. Koetter, "Byzantine attacks against network coding in peer to peer distributed storage," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2009, pp. 1164–1168.

[46] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 245–257, 2nd Quart., 2011.

[47] R. H. Jhaveri, "MR-AODV: A solution to mitigate blackhole and grayhole attacks in AODV based MANETs," in *Proc. 3rd Int. Conf. Adv. Comput. Commun. technol. (ACCT)*, Apr. 2013, pp. 254–260.

[48] B. B. Gupta, R. C. Joshi, and M. Misra, "Defending against distributed denial of service attacks: Issues and challenges," *Inf. Secur. J. A Global Perspective*, vol. 18, no. 5, pp. 224–247, Nov. 2009.

[49] H. Yao, D. Silva, S. Jaggi, and M. Langberg, "Network codes resilient to jamming and eavesdropping," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1978–1987, Dec. 2014.

[50] E. Kehdi and B. Li, "Null keys: Limiting malicious attacks via null space properties of network coding," in *Proc. IEEE 28th Conf. Comput. Commun. (INFOCOM)*, Apr. 2009, pp. 1224–1232.

[51] A. Asterjadhi and M. Zorzi, "JENNA: A jamming evasive network-coding neighbor-discovery algorithm for cognitive radio networks [Dynamic spectrum Management]," *IEEE Wireless Commun.*, vol. 17, no. 4, pp. 24–32, Aug. 2010.

[52] A. J. Newell, R. Curtmola, and C. Nita-Rotaru, "Entropy attacks and countermeasures in wireless network coding," in *Proc. 5th ACM Conf. Secur. Privacy Wireless Mobile Netw. (WISEC)*, 2012, pp. 185–196.

[53] Y. Jiang, Y. Fan, X. Shen, and C. Lin, "A self-adaptive probabilistic packet filtering scheme against entropy attacks in network coding," *Comput. Netw.*, vol. 53, no. 18, pp. 3089–3101, Dec. 2009.

[54] R. Iguchi and Y. Manabe, "An efficient edge-based authentication for network coding against entropy attacks," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2014, pp. 133–139.

[55] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure network coding for wireless mesh networks: Threats, challenges, and directions," *Comput. Commun.*, vol. 32, no. 17, pp. 1790–1801, Nov. 2009.

[56] V. N. Talooki and J. Rodriguez, "Jitter based comparisons for routing protocols in mobile ad hoc networks," in *Proc. Int. Conf. Ultra Modern Telecommun. Workshops*, Oct. 2009, pp. 1–6.

[57] D. G. Padmavathi and M. D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," 2009, *arXiv:0909.0576*. [Online]. Available: http://arxiv.org/abs/0909.0576

[58] Y. Zhang, W. Znaidi, C. Lauradoux, and M. Minier, "Flooding attacks against network coding and countermeasures," in *Proc. 5th Int. Conf. Netw. Syst. Secur.*, Sep. 2011, pp. 305–309.

[59] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 4, Mar. 2005, pp. 2235–2245.

[60] C. Gkantsidis and P. Rodriguez, "Cooperative security for network coding file distribution," in *Proc. IEEE 25TH IEEE Int. Conf. Comput. Commun. (INFOCOM)*, vol. 3, 2006, p. 5.

[61] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Apr. 2008, pp. 1409–1417.

[62] A. Yun, J. H. Cheon, and Y. Kim, "On homomorphic signatures for network coding," *IEEE Trans. Comput.*, vol. 59, no. 9, pp. 1295–1296, Sep. 2010.

[63] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2009, pp. 68–87.

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

[65] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding," in *Proc. Int. Conf. Appl. Cryptography Netw. Secur.* Berlin, Germany: Springer, 2009, pp. 292–305.

[66] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *J. Comput. Syst. Sci.*, vol. 18, no. 2, pp. 143–154, Apr. 1979.

[67] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," in *Proc. IEEE Conf. Comput. Commun., 18th Annu. Joint Conf. IEEE Comput. Commun. Societies, Future is Now (INFOCOM)*, vol. 2, 1999, pp. 708–716.

[68] M. Kim, L. Lima, F. Zhao, J. Barros, M. Medard, R. Koetter, T. Kalker, and K. J. Han, "On counteracting byzantine attacks in network coded peer-to-peer networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 5, pp. 692–702, Jun. 2010.

[69] Y. Jiang, H. Zhu, M. Shi, X. S. Shen, and C. Lin, "An efficient dynamic-identity based signature scheme for secure network coding," *Comput. Netw.*, vol. 54, no. 1, pp. 28–40, Jan. 2010.

[70] Intelligent Transportation Systems Committee, *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages*, Standard 1609, IEEE Vehicular Technology Society Standard, 2006.

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

[72] N. Attrapadung and B. Libert, "Homomorphic network coding signatures in the standard model," in *Proc. Int. Workshop Public Key Cryptography*. Berlin, Germany: Springer, 2011, pp. 17–34.

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

[74] D. Catalano, D. Fiore, and B. Warinschi, "Efficient network coding signatures in the standard model," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2012, pp. 680–696.

[75] D. Boneh and X. Boyen, "Short signatures without random oracles and the SDH assumption in bilinear groups," *J. Cryptol.*, vol. 21, no. 2, pp. 149–177, Sep. 2007.

[76] D. Catalano, D. Fiore, and B. Warinschi, "Adaptive pseudo-free groups and applications," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2011, pp. 207–223.

[77] C. Cheng, T. Jiang, and Q. Zhang, "TESLA-based homomorphic MAC for authentication in P2P system for live streaming with network coding," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 291–298, Sep. 2013.

[78] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, vol. 1, 2001, pp. 35–46.

[79] X. Wu, Y. Xu, C. Yuen, and L. Xiang, "A tag encoding scheme against pollution attack to linear network coding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 33–42, Jan. 2014.

[80] C. Cheng, J. Lee, T. Jiang, and T. Takagi, "Security analysis and improvements on two homomorphic authentication schemes for network coding," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 993–1002, May 2016.

[81] A. Esfahani, G. Mantas, and J. Rodriguez, "An efficient null space-based homomorphic MAC scheme against tag pollution attacks in RLNC," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 918–921, May 2016.

[82] A. Esfahani, D. Yang, G. Mantas, A. Nascimento, and J. Rodriguez, "Dual-homomorphic message authentication code scheme for network coding-enabled wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 7, Jan. 2015, Art. no. 510251.

[83] 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," *Int. J. Inf. Secur.*, vol. 16, no. 6, pp. 627–639, Sep. 2016.

[84] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, Jun. 2005.

[85] 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 Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.

[86] J. P. Vilela, L. Lima, and J. Barros, "Lightweight security for network coding," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 1750–1754.

[87] A. Le and A. Markopoulou, "Cooperative defense against pollution attacks in network coding using SpaceMac," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 442–449, Feb. 2012.

[88] H. Ali-Ahmad, C. Cicconetti, A. de la Oliva, V. Mancuso, M. R. Sama, P. Seite, and S. Shanmugalingam, "An SDN-based network architecture for extremely dense wireless networks," in *Proc. IEEE SDN Future Netw. Services (SDN4FNS)*, Nov. 2013, pp. 1–7.

[89] T. Mahmoodi and S. Seetharaman, "On using a SDN-based control plane in 5G mobile networks," in *Proc. 32nd Wireless World Research Forum, Meeting*, 2014, pp. 1–6.

[90] V. Adat, I. Politis, C. Tselios, and S. Kotsopoulos, "Secure network coding for SDN-based mobile small cells," in *Proc. Int. Conf. Broadband Commun., Netw. Syst.* Springer, 2018, pp. 347–356.

[91] V. Adat, I. Politis, C. Tselios, P. Galiotos, and S. Kotsopoulos, "On blockchain enhanced secure network coding for 5G deployments," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[92] V. Adat, I. Politis, C. Tselios, and S. Kotsopoulos, "Blockchain enhanced SECRET small cells for the 5G environment," in *Proc. IEEE 24th Int. Workshop Comput. Aided Modeling Des. Commun. Links Netw. (CAMAD)*, Sep. 2019, pp. 1–6.

[93] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: A scalable blockchain database," BigchainDB 2.0, The Blockchain Database, BigchainDB GmbH, Berlin, Germany, White Paper, May 2018, version 1.

[94] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[95] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, and Y. Manevich, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, p. 30.

VIPINDEV ADAT VASUDEVAN (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from Mahatma Gandhi University, Kerala, India, in 2014, and the M.Tech. degree in computer engineering (cyber security) from the Department of Computer Engineering, National Institute of Technology, Kurukshetra, India, in 2017. He is currently working as a Researcher and a MSCA Fellow with the Wireless Telecommunication Laboratory, Department of Electrical and Computer Engineering, University of Patras, Greece. His research interests include but are not limited to network security, network coding, 5G small cell networks, and the Internet of Things.

CHRISTOS TSELIOS received the Diploma degree from the Electrical and Electronic Engineering Department, University of Patras, in 2009. He has been a Research Fellow with Ericsson EUROLABS, Aachen, Germany, and the Cassiopeia, Department of Computer Science, Aalborg University, Denmark, since 2011. He is the author of several research articles in international journals, conferences, and edited books. He has also participated in several European (both FP7 and H2020) and national (GSRT) projects related to the ICT domain, including PEACE, ROMEO, DIOGENES, SUPERFLUIDITY, GamECAR, and SMESEC. His research interests include but are not limited to 5G networks (mobile edge computing and fog computing architectures), network security, the Internet-of-Things protocols, and machine-to-machine communication.

ILIAS POLITIS (Member, IEEE) received the B.Sc. degree in electronic engineering from the Queen Marry College London, U.K., in 2000, the M.Sc. degree in mobile and personal communications from the King's College London, U.K., in 2001, and the Ph.D. degree in multimedia communications from the University of Patras, Greece, in 2009. He is currently a Senior Researcher with the Wireless Telecommunications Laboratory, Electrical and Computer Engineering, University of Patras, and a Senior Researcher with Hellenic Open University, Greece. He has been actively involved in all phases of H2020-MSCA-SECRET, H2020-MSCA-SONNET, H2020-RIA-EMYNOS, FP7-ICT-ROMEO, FP7-SEC-SALUS, and FP7-ICT-FUTON projects, as well as, several national funded research projects, while research interests include future generation networks, next-generation multimedia networking, and emergency communications.