Security Concerns and Countermeasures in Network Coding Based Communications Systems: A Survey

Vahid Nazari Talooki[1,*], Riccardo Bassoli[1], Daniel E. Lucani[2], Jonathan Rodriguez[1], Frank H. P. Fitzek[2], Hugo Marques[1], Rahim Tafazolli[3]

[1]Instituto de Telecomunicações, Aveiro, Portugal
{vahid, bassoli, hugo.marques, jonathan}@av.it.pt
[2]Department of Electronic Systems, Aalborg University, Aalborg, Denmark
{del, ff}@es.aau.dk
[3]Centre for Communication Systems Research, Surrey, UK
r.tafazolli@surrey.ac.uk
[*]Corresponding Author, Tel.: +351234377900
Campus Universitário de Santiago, Aveiro, Portugal, P-3810-193

Abstract

This survey paper shows the state of the art in security mechanisms, where a deep review of the current research and the status of this topic is carried out. We start by introducing network coding and its variety of applications in enhancing current traditional networks. In particular, we analyze two key protocol types, namely, state-aware and stateless protocols, specifying the benefits and disadvantages of each one of them. We also present the key security assumptions of network coding (NC) systems as well as a detailed analysis of the security goals and threats, both passive and active. Current proposed security mechanisms and schemes for NC in the literature are classified too.This paper also presents a detailed taxonomy of the different NC security mechanisms and schemes reported in the literature.

Keywords—Network Coding; Security Attacks; Eavesdropping Attacks; Byzantine Attacks; Corruption Attacks;

1 Introduction

In coding theory, there are three main coding families: source coding, channel coding, and network coding (NC). The first aims to compress information at the source, while the second introduces redundant bits at the link layer to guarantee reliable communications. On the other hand, the third consists in a coding process that takes place at intermediate nodes in the network and at different layers of the network stack.

Network coding [1] represents a generalisation of classical store-and-forward routing for information flow. This new code-and-forward paradigm considers that the source messages are algebraic entities upon which operations can be performed at the intermediate nodes. This contrasts with current state-of-the-art routing solutions, where source messages are merely routed from source to destination. The main result of [1] was the enunciation of the max-flow min-cut theorem for information flow. In

particular, the authors demonstrated that the multicast capacity is achieved by applying network coding. This achievement can bring a drastic change in improving the data throughput of networks. Therefore, several studies focused on implementing practical network coding based approaches that lead network coding to its current level.

In 2003, [2] demonstrated that linear operations at the nodes were sufficient to achieve the max-flow min-cut bound: linear network coding (LNC) was defined in directed acyclic networks with single-source multicast. Side by side, [3] provided an algebraic formulation of linear network coding by using abstract algebra and algebraic geometry, and demonstrated equivalent results in the new framework for both acyclic and cyclic networks. This algebraic formulation opened the way to random linear network coding (RLNC) [4], a kind of network codes, in which the coefficients of linear combinations are randomly chosen over a finite field. [5] provided a first description of how to implement RLNC in practice: it analysed both its benefits in terms of throughput and its main issues. In fact, real information is flowing asynchronously so RLNC can suffer delays and losses, and it can eventually experience congestions and link failures. So, to be practical, the RLNC should be designed with a special packet structure by taking into account new key characteristics to overcome these issues.

1.1 *Secure network Coding*

The new way of managing information, that network coding introduces in actual networks, presents new several challenges in terms of security: processing (recoding) the received data packets from neighbors in the intermediate nodes and then forwarding them, opens a myriad of challenging security issues [6] . In fact, network coding can have either positive or negative secure aspects. In the former case, by sending linear combinations of packets and not merely source data, an adversary that is intercepting some transmissions collects information that results to be useless. On the other side, coding operations across packets can make the overall network more vulnerable against several types of attacks. The research on secure network coding has been growing by mainly investigating both byzantine and eavesdropping attacks. In fact, protocols based on network coding present vulnerability against many threats and attacks including but not limited to impersonation, byzantine (fabrication, modification and replay) attacks, blackhole, and eavesdropping.

Some of the first analyses on secure network coding are [7-11]. These works considered an eavesdropper seeing information transmitted on a subset of network channels in a single-source scenario. In order to study secure network coding, different models has been used. [9, 10] for first time proposed a model for a collection of subsets of wiretap channels for an NC system called *wiretap network*. Each wiretrapper in the model has full access to only one of these subsets; however, by applying secure linear network codes, none of the wiretapper is able to extract any information from the transmitted message. At the first analyses, shown in [12, 13], the measure of security has been done in terms of either information quantities or decoding probability. Next, [14] proposed an algebraic secure criterion. Side by side, the issue of designing secure network codes has

been also investigated from another point of view: first in 2003 and then in 2006, [8, 15, 16] described how network coding could be seen as a generalization of classical error correction. In particular, network error correction (NEC) coding has been proved to be optimum in correcting random errors, erasures and errors due to malicious nodes in the network. [17] started researching on secure network coding through NEC coding. The objective of that work was the correction of errors injected by wiretapper and the protection of source messages from wiretapping.

At the best of authors´ knowledge, this is the first survey to include the most relevant literature in the diverse research areas related to security attacks and mitigation techniques in network coding based communication systems. This survey includes more than 200 references and makes a broad description of security threats and attacks in network coding based systems, reviewing the current mechanisms against these security attacks and the latest results for proving a secure network coding approach. We believe both advanced and initial researchers in this area can benefit from this survey.

Other relevant surveys on principal concepts of network coding theorem are [6, 18], and the most recent [19]. Some tutorial on security attacks and threats in network coding based systems and summarizing the current mitigation techniques are [20-25]. Network coding website [26] provides several studies and papers on this filed too. Also, another useful source is [27] that includes "Bibliography on Secure Network Coding" and list of paper works in the scope of secure NC systems from 2006 to 2014.

1.2 *Structure of the Survey*

In what follows, in Section 2 we review the fundamentals of network coding, the security assumptions in NC systems and state-aware and stateless NC protocols. Then in Section 3, the security threats and attacks in the NC based systems are studied. In Section 4 we classify the proposed security mechanisms and schemes for NC in the literature. In Section **Errore. L'origine riferimento non è stata trovata.** we present a timeline of these mechanism and schemes. Finally Section 5 presents the considerations and conclusions.

2 Preliminaries

2.1 *Principles of Random Linear Network Coding*

As an example to show the capability and the benefits of using NC in improving network throughput, Figure 1 shows a possible simple scenario for both traditional store and forward mechanism and network coding elegant *store-process-forward* paradigm. Here the source node S wants to multicast some packets toward two sink nodes D_1 and D_2. Each packet like $p_{Time\ Stamp}^{Packet\ Number}$ has a packet number and time stamp that shows the packet number which was assigned by source node to it and the time that packet was forwarded. For simplicity each packet (or symbol) is considered as one bit.

A traditional store and forward mechanism, would achieve maximum throughput of 1.5 bits/s but NC allows both D_1 and D_2 to achieve a rate of 2 bits/s at the same time which means more than 30% improvement in throughput for this scenario.

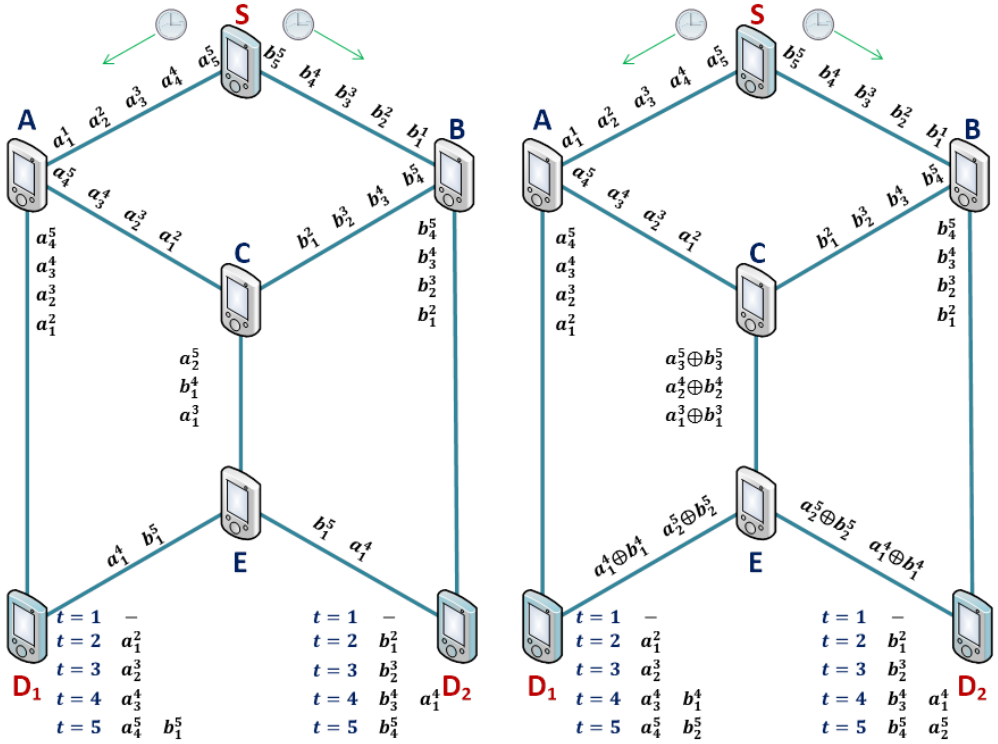

Figure 1: (a) Traditional store and forward mechanism vs. (b) Network coding store-process-forward paradigm: network coding reaches to 2 bits/s multicast throughput for this scenario which is 30% better than traditional scheme.

In general, RLNC can be designed in practice according to two main approaches, called respectively intra-session and inter-session. In the former [28-31], routers combine packets belonging to the same session. It is typically used in multicast application and in case of unpredictable topologies, and it has been demonstrated to improve reliability. The decoding operations are only performed at the destination. In the latter [32-36], packets from distinct flows are mixed when they pass through a common router. This approach is especially suitable for unicast applications and static topologies. Its main benefit is to improve the throughput. Figure 2 depicts an example of random linear network code in a butterfly scenario.

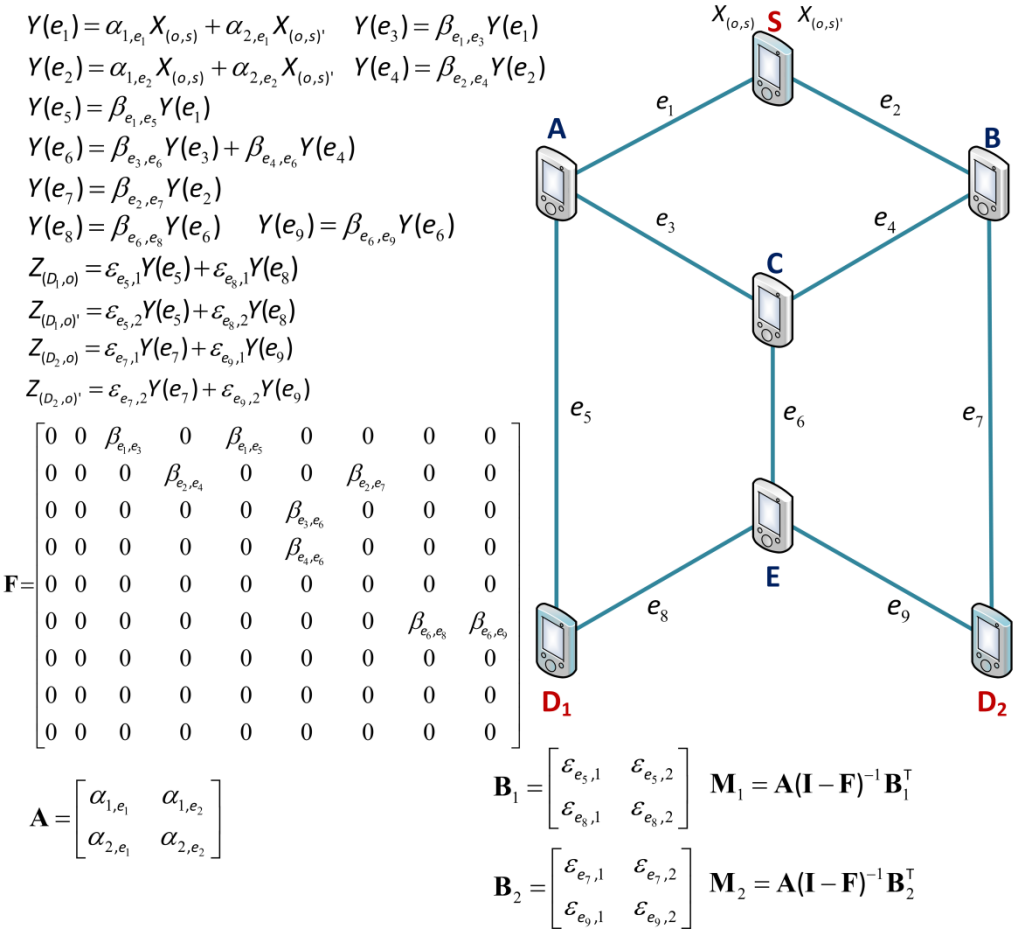

Figure 2: Representation of a random linear network code in the butterfly network with correlated source processes. Either all the coefficients of the linear combinations or a part of them are randomly chosen over a finite field of size q. In particular, $X_{(o,s)}$ and $X_{(o,s)'}$ denote the source processes, Y the processes at the edges, $Z_{(D_1,o)}$ and $Z_{(D_1,o)'}$ the processes collected by sink 1, and $Z_{(D_2,o)}$ and $Z_{(D_2,o)'}$ the processes collected by sink 2. Also, here **I** is an Identity matrix in with ones only on the main diagonal and zeros elsewhere. Matrices **A** and **B** are random matrices that contains the coefficients, matrix **F** is the adjacency matrix of the directed labeled line graph [3]. The transfer matrix of the system is **M**.

The randomized approach of RLNC is distributed, easier to implement than LNC, and especially suitable for changing topologies, large networks or in the presence of dynamically varying connections. The key characteristic of RLNC is that coefficients of linear combinations are chosen randomly over a finite field. That implies the transmission of coding vectors to the receivers by appending them as an overhead to the header of the messages. This overhead is quantified as $h \log q$, where h is the number of information flows at the source and q is the size of the finite field.

However, the capability of designing network codes without knowledge about the network is paid in terms of successful decoding probability: in fact, in randomised scenarios, the capability of a receiver to completely decode the information depends on the number of sinks and the size of the finite field. That also introduces a trade-off between the decoding error probability and the complexity of the coding operations over the finite field.

In a network composed of several source nodes (encoders), intermediate nodes (recoders), and sink nodes (decoders) where each source node generates at least g linear combinations (codewords) of h native data packets and floods them into the network as one generation. Also, q is size of finite filed $\mathbb{F}_q$ whose binary representation length is m (where $q = 2^m$). In this approach, a sink may fail to fully recover the native packets even in a network with ideal channels that leads to increasing in *decoding error probability*.

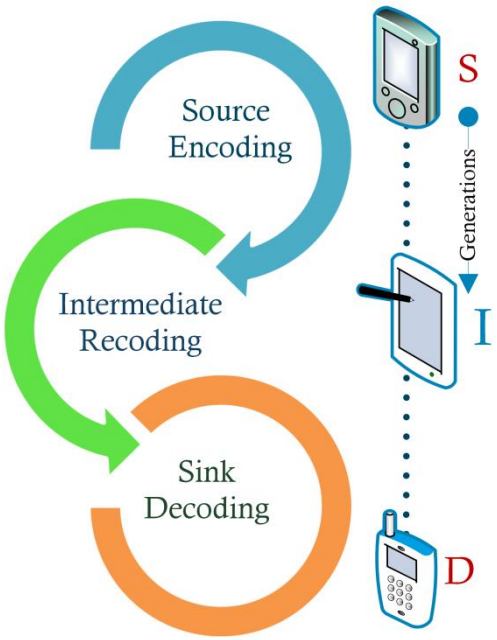

Figure 3: A sample scenario that three mobile nodes communicate with each other in a RLNC based line network

The possible reasons for decoding error at sink node are:

- Insufficient receiving codewords: the sink node may not receive g codewords.
- Insufficient rank: the rank of total received codewords is less than h which means at least some of the receiving codewords are not innovative. The erroneous and erasure channels also can lead to insufficient rank of received codewords at sink node.

- Security attacks: in this case, the number and rank of received codewords is sufficient but these codewords are polluted or fabricated by a malicious intermediate node.

Figure 4 shows the decoding error probability P_e versus m. "SD", "SI1D", and "SI5D" label in Figure 4 shows there is no intermediate node in this line network, there is one intermediate node, and there are five intermediate nodes. The generation size is *32* and each native packet size is *1.5 Kbits*. As illustrated by Figure 4, the lower P_e could be achieved by a larger filed size.

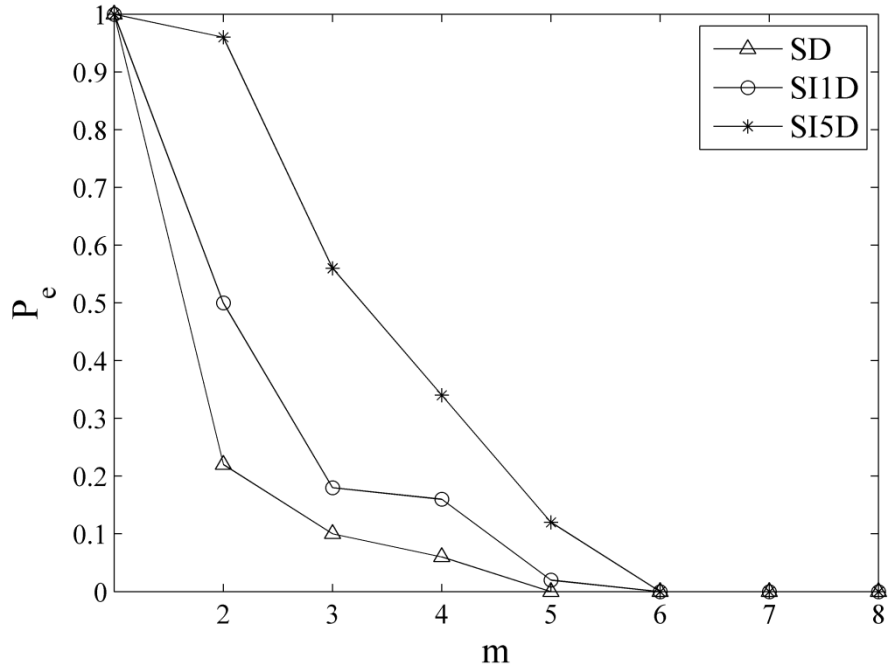

Figure 4: Error decoding probability (P_e) versus m

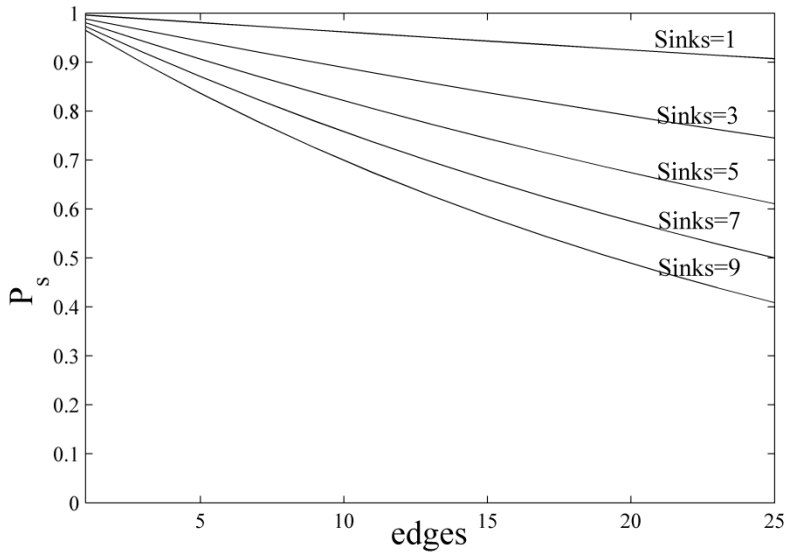

Figure 5: P_s versus number of edges

Since the coefficients of the linear combinations in RLNC are randomly chosen over a finite field of size $q = 2^m$, the decoding process has an intrinsic decoding error probability. In case of an acyclic network, the lower bound on the successful decoding probability is $P_s = (1 - \frac{d}{q})^{|E|}$, where d is the number of destinations, q is the size of the finite field and *E* is the number of edges with associated random coefficients. The value of the probability approaches 1 when $q \to \infty$ [37]. Figure 5 shows P_s versus number of edges for different number of sink nodes in the network where $q = 256$.

2.2 *Security Assumptions for NC Systems*

Beside general and basic features such as capability for sending and receiving packets, and being equipped with a normal processing power and storage device, such as CPU and memory, we expect several rules that should be followed by a well behaved, Benign node [38, 39]:

- *Coding:* Performing valid coding operations such as mixing, encoding different data packets, and contributing actively and correctly in the overall *store-process-forward* mechanism.
- *Recoding:* Recoding the data packets that are merely intended for it, therefore satisfying the basic confidentiality requirements and enabling the sink nodes to correctly decode data the packets and extract the information. Then the recoded packets should be forwarded correctly and validly.
- *State Dissemination:* Participating in the timely dissemination of correct state information (applicable in the state-aware NC protocols) [40].

When a node violates one or more behaving well rules, the NC system will be vulnerable to several attacks. There are several assumptions related to an adversary node in an NC system [41]:

- Adversary nodes may violate one or several rules of well behaving nodes.
- In a source-sink data flow, an adversary that is an intermediate node can observe the transmission and the data packets.
- An adversary node, like other well-behaved nodes, has a full access to the required algorithms and procedures of encoding and decoding.
- An adversary node is not unlimited in terms of resources (such as CPU, memory, and bandwidth) and therefore it is not able to break hard cryptographic primitives.

A malicious node may bogus data packets or corrupts them and injects them into the network to perform a *pollution attack* [42]. There is a wide range of security threats and attacks in NC based systems. We will discuss and list them in more detail in Section 5. Furthermore, even in a network composed of benign nodes, lossy or erroneous channels may lead to receive corrupted codewords at sink nodes. Figure 6 shows the decoding error probability (i.e., p_e as discussed in section 2.1) in the a network, illustrated by Figure 3, when the intermediate malicious node attacks the transitive symbols and *corrupts* some of them. Figure 6 shows that in the recoding phase in a malicious intermediate node, even a very low percentage of corrupted symbols at intermediate node (X-axis in Figure 6) leads to $p_e = 1$.

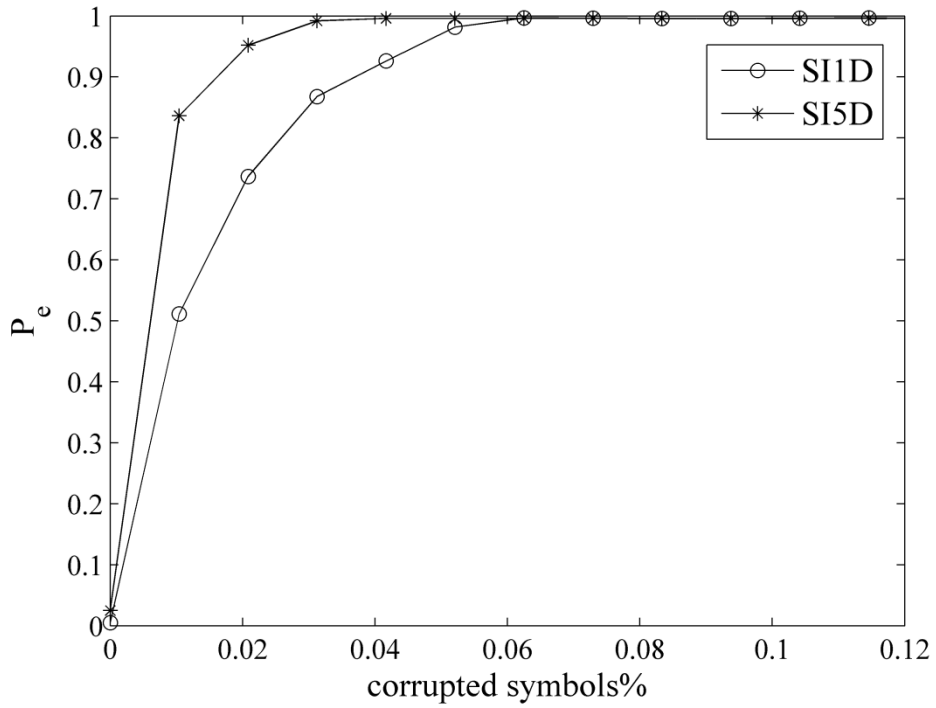

Figure 6: decoding error probability versus percentage of corrupted symbols

The main security goals, that a secure NC system should be *aware* of are as follows:

- Authentication: It involves data integrity, data origin authentication, and nonrepudiation [43, 44].

- Confidentiality: The transmitted information (native data) between any two nodes should be protected and not accessible to an outsider who is not authorized.
- Detection and Isolation: It should be able to identify misbehaviours in the network, to detect adversary nodes, and to isolate them (quarantine) in order to prevent any damage in the network.
- Availability: this feature shows whether a node in the network is able to use the resources and whether the network is accessible for the message to be communicated.

Security goals and requirements for an NC network are not limited to this short list and there are also other requirements, such as ordering, timestamp, location privacy, and self-organization [45-47].

2.3 *NC Protocols Categorization*

Different categorizations of NC aware routing protocols for wireless networks, namely centralized [48], source routing [49-51], hop-by-hop [28, 32, 52-56], and active [57, 58], are presented and reviewed in [59]. In general scene, NC protocols can be in two main groups based on their use of network state information [41]. The security threats and attacks and also the solutions are sometimes different for the stateless and the state-aware NC protocols.

A. Stateless NC Protocols

The stateless NC protocols do not rely on network state information, such as topology, link cost, and node location, to perform coding operations like the mixing of data packets. In stateless NC protocols, nodes do not rely on any assumptions about network topology accordingly coding operations in a communication (coding at source node(s), recoding in intermediate node(s), and decoding in sink node(s)) are independent from dynamically changing topologies. A sample for the networks with a dynamic topology is Mobile Ad hoc NETwork (MANET) [60], where no dedicated infrastructure exists-users collaborate in the routing process by storing and forwarding data packets to their neighbours.

As discussed in Section 2, by properly implementing RLNC, only the sink nodes who have access to sufficient decoding vectors can recover native packets [61, 62]. So in comparison to traditional routing protocols, nodes not only may be naturally able to protect data packets against common security threats and attacks but also gain other benefits [38, 40]:

- The stateless NC protocols do not rely on topology state information; therefore, they are more immune against using wrong e.g., through Byzantine modification attack, fake (e.g., through link spoofing and pollution attacks), or obsolete topology and link state information (e.g., through Byzantine fabrication attack).
- The sink nodes only need to receive sufficient linearly independent packets (no matter from which neighbours), setup a system of linear equations via these coded packets, and solve this system via Gaussian elimination to extract native

data packets. Therefore, decoding operations in sink nodes in stateless protocols is independent from the identity of nodes who sent those coded packets which guaranties immunity against some attacks such as impersonation;

- Coded packets may reach to the sink node(s) in several different paths and intermediate nodes. This native redundancy property of stateless NC mechanisms like RLNC can limit the negative impact of adversarial nodes misbehaviour in traffic relay refusal.

The use of the stateless NC protocols also brings some additional points of concern [63, 64]:

- Due to the independency from network topology, some techniques like optimization based on local topology information are not applicable anymore. Thus these protocols need more sophisticated coding operations. Also, packet level functions (storing, mixing, encrypting, and decrypting) need to be fully distributed in order to provide flexibility to dynamic network topology changes;
- To guarantee that randomly chosen network codes can properly perform encryption and decryption operations in source and sink nodes, they should be selected from a sufficiently large field which increases data overhead. The required decoding vector should be included in the header of each data packets.

B. State-aware NC Protocols

The state-aware NC protocols rely on partial or full network state information to optimize the coding operations carried out by each node. Here, nodes have some information about network topology; hence, they can use local information to achieve the most optimized encryption codes. Also, they can exchange the required coding vectors at the beginning of communication.

The optimization process in the state-aware NC protocols may target the throughput or the delay. The optimization process in a node can be based on exchanging information only with close neighbours (local optimization) or it can address the end-to-end communication across the entire network (global optimization). [65] proposes a polynomial time algorithm for global optimization. The COPE [66] protocol, which runs between the IP and MAC layers, is a state-aware NC protocol that uses local information and network sate information. The other state-aware NC protocols are explained in [34, 67, 68]. By using network state information, these protocols try to achieve improvements in terms of throughput and robustness however other issues raises that should be handled such as:

- In the state-aware NC, nodes need to know about local topology (e.g. one or two levels of neighbors) or global topology of network [59] that prepare a fortune for adversarial nodes for distributing invalid control-traffic and network state information.
- In order to optimize the encodings process via more efficient coding coefficients and less overhead, the state-aware protocols rely on the vulnerable control information disseminated among nodes or network state information that they opportunistically overheard from their neighbors. However, this mechanism

gives a chance to malicious nodes for crafting and injection of false control packets.

On the other hand, nodes in the state-aware protocols have more knowledge about nodes location, topology, and link state which can lead to some benefits:

- By using network state information, exchanging topology information between neighbours, and using fixed predefined and optimized coding vectors at the beginning of communication, the state-aware NC protocols can achieve a higher performance in terms of throughput or delay.
- The required decoding vectors can be calculated and exchanged at the beginning of communication and consequently, unlike the stateless protocols, they do not need to be included in the header of all data packets, which subsequently decreases the overhead.

Table 1 lists the benefits and drawbacks of the state-aware and the stateless NC protocols especially in terms of security issues. Other attacks like Byzantine attacks (described in next section) can have considerable negative impact on all NC protocols regardless of being stateless or state-aware.

Table 1. Benefits and drawbacks of state-aware and stateless NC protocols

Protocol Type	Key Features	Benefits	Drawbacks
Stateless NC Example: RLNC	Do not rely on network state information	Do not use control packets for updating their knowledge about topology state	Need more sophisticated coding operations
	Coding operations can work properly even under a dynamic network topology	More prone against wrong network state information and invalid control traffic packets that leads to be more immunity against some types of fabrication, modification, and impersonation attacks.	To guarantee a successful decoding, coefficients should be selected from a sufficiently large field which increases the overhead.
	Nodes chose encoding coefficients randomly and independently	Extracting information from received packets in receivers is independent from the identity of sender nodes	The required decoding coefficients vector should be included in the header of each data packets.
State-aware NC Example: COPE	Rely on local(one or two levels of neighbors) or global network state information	Instead of including encoding coefficient at the header of all packets, the fixed optimized coefficients codes can be exchanged at the beginning of communication.	Rely on network state information, routing tables, control packets, and so on.
	Nodes can use network state information to achieve the most optimized encoding codes.	Local or global optimization schemes	Rely on vulnerable control packets which lead to more threats and attacks

3 Active versus Passive: The Usual Suspects

Common security threats and attacks in NC systems can be divided in two main categories [63]:

- ***Passive*** threats and attacks do not disturb the normal operation of the network. Malicious nodes that perform a passive attack only snoop (or read) the exchanged information without modifying it. The detection of this type of attacks is difficult because it does not compromise the operation of the network. Passive attacks mainly violate confidentiality or are used to reveal information on the network topology or capturing sensitive information such as passwords.
- ***Active*** attacks, contrary to passive attacks, try to disrupt the normal network operation and may also alter, corrupt, or delete the data packets being exchanged in the network. Active attacks can be caused by an external (outside the network) or an internal (belonging to the network) attacker. Internal attacks are more difficult to be detected and mitigated [47]. As an example of an active attack in NC systems, a malicious node can store and pollute (corrupt) data packets by using malware processing functions and then forward (inject) the polluted packets into the network. This attack can extend (advance) rapidly, leading to a network full of polluted packets.

In this section, we shortly review general security threats that are vital to be managed by a NC system. Not addressing these threats may nullify the performance gain of coded networks or even worse they can completely disrupt the whole network operation. We shall divide our discussion in passive attacks and active attacks, particularly focusing on eavesdropping, Byzantine modification and pollution, which are some of the most important potential security weaknesses of NC systems. Then, in the next section we will focus on eavesdropping resulting from passive attacks and Byzantine modification and pollution resulting from active type, that are the main wide range attacks which can cover most of the security issues in all NC systems.

3.1 *Passive threats and attacks*

Two of the main and most common passive attacks are eavesdropping and traffic analyses.

A. Eavesdropping

An eavesdropper attacker reads data traffic to obtain sensitive information (e.g., native data, secret keys, and location) about the other nodes. In NC stateless protocols, a malicious intermediate node can act as an eavesdropping attack if it has access to a sufficient number of linearly independent combinations of packets. In this case, the malicious node can easily decode the packets and can have access to all transmitted information. In state-aware NC protocols this attack has a higher probability of succeeding because the number of the packets needed for the decoding operation is less

than that of stateless NC protocols. This makes eavesdropping attacks to state-aware NC protocols more dangerous and serious.

Figure 7 shows several possible scenarios for security attacks in a network coding system such as eavesdropping, selectively dropping, Byzantine modifying, and Pollution attacks (they will be discussed later in this section). Unlike figure 1, the time stamp of packets is disregarded in figure 7, for simplicity. Suppose that node E in figure 7.(a) only should only send packet $a \oplus b$, where $\oplus$ represents the bit by bit XOR of the two packets, and node E is not authorized to have access to the native packets a and b. Here if node E, as an adversary internal eavesdropper, success to overhear at least one of two links AD_1 and BD_2, eventually it can decode XOR packets and obtain fully access to the both native packets a and b. Beside possible internal eavesdroppers, the network could be threaten by an external eavesdropper such as node F in figure 7.(a); it tries to overhear AD_1 and ED_1 links to successfully decode the XOR packets and obtain unauthorized native packets a and b.

As one of the first studies, [69, 70] investigated classical security measures in NC systems: how, without any shared keys and secure channels and by network codes, we can have a secure data transmission over a network in which some wiretapped links are controlled by computationally unbounded and hidden adversaries who have full access to the messages for observing and modifying. [69] proposed network codes by the maximum secure throughput $(1-p)|E|$ where an adversary controls a fraction $p < 0.5$ of the $|E|$ edges. [70] proposed a network code for this problem in which the adversary capacity (i.e. the number of coded messages that an eavesdropper has full access to them) is less than the overall network multicast capacity. In this model, only the source and the sink nodes are responsible for the secure coding and error detecting and correcting is not *on-the-fly*. [71] showed that eavesdropping attack in wiretapped NC systems can be considered as a network generalization of the Ozarow-Wyner Wiretap channel of type II and determined a bound on the required secure code alphabet size.

Let us consider a network where an *eavesdropper* is listening to all the messages sent on a subset of edges. A system is called Shannon secure if the eavesdropper has no information about the source. On the other hand, if the attacker is seeing the information the source is transmitting but he is not getting any meaningful information the system is called weakly secure. A random network code is implicitly providing weak security: in fact, an attacker that is intercepting random linear combinations of packets is not guessing any meaningful information unless the number of packets he owns is less than the rank of the transfer matrix of the system. In particular, the probability that an eavesdropper is obtaining meaningful information about the source is less than $|\mathcal{A}|uk/q^{h-k}$, where $\mathcal{A}$ is a collection of sets of edges, u is the multicast rate of the code and k is the number of independent messages [72]. A random code is not Shannon secure with a probability $|\mathcal{A}|k/q$: this shows that the security of a random network code increases with the size of the finite field. So, there is a tradeoff between the complexity of the code and its security.

There are several solutions for handling an Eavesdropping attack, as a main and the most important passive attack in the NC systems. In the next section, a history of works on eavesdropping attacks in NC systems, attack modelling, and proposed mechanisms for handling it, is presented.

B. Traffic Analysis and Monitoring

An attacker may monitor and analyse packet transmission in order to extract information about the source and the destination as well as the network topology.
Generally, traffic analysis and monitoring threat is due to violating the privacy of nodes by an adversary node. Handling these threats could be more challenging because of intermediate nodes authorization for processing the packets in the NC systems However, in the other hand, due to the nature of coded networks in using coded packets in intermediate nodes, NC has a potential to thwart these threats if a proper coding mechanism is applied. Both the state-aware and stateless NC protocols can be jeopardized by this threat. There are several works in the literature that have focused on traffic analysis threats and attacks [73-75].

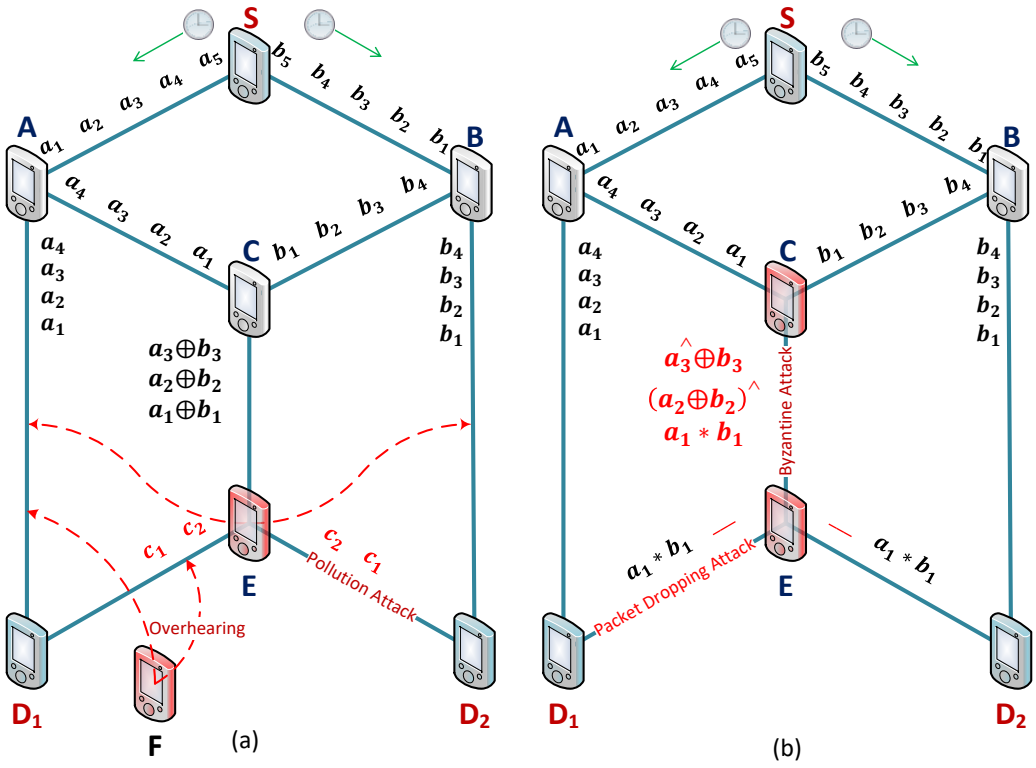

Figure 7: (a) node E bogus fake packets to carry out a pollution attack. Here also node E and F are internal and external eavesdropper respectively that try to overhear some unauthorized packets to obtain required information for decoding XOR messages. (b) Shows the Byzantine modifier node C that modifies received packets and performs invalid operation on them (e.g., the original packet a_3 that is polluted by malicious node C is shown by packet $a_3^{\wedge}$). Also the attacker node E selectively forwards some packets and drops the others.

3.2 *Active threats and attacks*

Contrary to passive attacks, active attacks try to disrupt the normal network operation and may also alter, corrupt, or delete the data packets being exchanged in the network. Active attacks can be caused by an external (outside the network) or internal (belonging to the network) attacker. Internal attacks are more difficult to be detected and mitigated [45, 47]. In an NC system, one malicious node can store and pollute (corrupt) data packets by using malware processing functions and then forward (inject) the polluted packets into the network - this procedure can extend rapidly, leading to a network full of polluted packets. This type of attacks is called pollution attack. Below a list of some attacks which can threaten NC systems are presented.

A. Denial of Service (DoS)

A benign node may get flooded by other nodes of network, which in extreme cases the victim may no longer be able to operate properly or it may even fails. In a NC system, when a victim node receives lots of requests (such as packets processing and forwarding) from either benign or adversary nodes, that may lead to lack of sufficient resources, e.g., bandwidth, CPU power, memory, and battery level in the victim. Also, if attackers have enough resources like computing power and bandwidth they can perform a more severe distributed denial of service (DDoS) attack. DoS can have a variety of origins and it affects different layers of network [76].

DoS may have an unintentional origin in the NC aware multi-hop routing protocols. For their location in the topology of network, some particular nodes may become unfairly burdened to support many packet mixing, coding, and forwarding functions and this leads to more loads on these hot spots in terms of radio jamming and battery power exhaustion [77].

In NC aware schemes, a malicious node can easily perform a DoS attack by flooding its neighbors via injecting lots of junk or corrupted block of coded packets or even clean but old and reparative packets into downstream victim nodes who should perform frequently the decryption, encryption, and forwarding functions that leads to high traffic and rapid battery exhaustion in victims.

DoS attacks can also happen in different forms in several layers of network stack like jamming and tampering at physical layer, collision, exhaustion and unfairness at link layer, sleep deprivation, black holes, routing table overflow at network layer, malicious flooding and de-synchronization at transport layer, and finally failure in the remote services, e-banking, and web servers at application layer [45]. Due to distributed mechanism of NC schemes and intermediate nodes role in packet mixing, DoS and DDoS are very challenging to handle and need efficient techniques.

Lima et. al [78] investigate the impact of DDoS as a type of Byzantine attack on peer-to-peer topologies for distributed NC aware storage that shows severe degrade of network performance even for a small number of Byzantine nodes.

Statistical analysis of network traffic for detecting the suspicious nodes that send a large amount of packets, control messages, and requests for their neighbours, or applying an authentication and verification of traffic flows. Also, packet leashes such as adding a field to the header of packet in order to limiting the maximum allowed hops for the packet could be another possible mitigation technique.

B. Jamming

A malicious node can perform a particular Denial of Service (DoS) if it prevents the reception (jam) of valid packets, stops forwarding packets towards a destination, or injects lots of fake packets into the network. In comparison of traditional networks, NC aware schemes are more vulnerable to the jamming attacks, especially to the cooperative jamming attacks [79]. Also, jamming attack has more negative impact on the state-aware NC protocols in comparison with the stateless protocols.

Authors in [80] proposed a polynomial-time, rate-optimal distributed NC scheme for a secure communication in a network by capacity C and in the presence of malicious nodes by jamming rate Z_O and eavesdropping rate Z_I . It can reach to the optimal secure source rate of $C - Z_O - Z_I$. Several countermeasures have been proposed in the literature [80-84] for handling malicious jammer nodes. However, these solutions, such as using computational expensive hash functions (look at next section), heavy monitoring algorithms, and limiting the malicious jammers by null space properties of NC [83], may severely decrease the performance of NC mechanism.

C. SYN Flooding

In 2006, S. Katti et. al [32] proposed a wireless mesh protocol (COPE) that can handle both TCP and UDP connections. However, they referred to several problems like a high collision rate due to hidden terminals and lack of coding opportunities in the case of TCP communications, which leads to poor gain of utilizing NC for TCP communications. Y. Huang et. al [85] later showed that these problems are related to the COPE design, which does not consider opposite DATA and ACK flow directions in the TCP. Thus, they proposed an NC aware approach that improves TCP communication throughput by opportunistically XORing DATA and ACK packets within a TCP session without modifying TCP or the underlying MAC protocol [86].

In 2007, [87] proposed a MAC layer NC scheme to improve TCP performance over wireless networks. Some works modified the traditional MAC layer for further improvements in coding gain [88]. [89] is one of the earliest work proposing a new mechanism called TCP/NC that incorporates NC into TCP to improve TCP mechanism in lossy networks. Later, many other studies worked on improving TCP performance; however, most of them require TCP and network stack layer modifications [89-96].

Although the previously mentioned works [85-96] showed NC capabilities in TCP communications and improved the throughput of network, they can lead to a myriad of insecurity issues that should be handled. One of the most important security attacks in transport network layer, which is due to TCP connections, is Synchronization (SYN) flooding attack. Here, the attacker generates lots of half-opened TCP connections with

some victim nodes but never fulfill the handshaking procedure to fully open the connections. Applying NC mechanism on TCP connections is quite new; therefore, to the best of our knowledge, there is still no existing efficient mechanism for handling TCP related attacks like SYN flooding in NC systems.

D. Resource Exhaustion

Coding and mixing different packets into one packet at intermediate node, is an essential task of NC systems. However, the authorization of intermediate nodes for recoding or processing the packets as a part of *store-process-forward* paradigm of NC increases the system throughput but in other hand it may lead to several security issues that can potentially nullify the performance gain of coded networks.

An adversary intermediate node can execute a wide range of Byzantine attacks e.g., Byzantine modification, on the transit packets or forges bogus packets and inject them into the network as a part of a pollution attack. These improperly coded or polluted packets will be forwarded to the next nodes for further recoding processes and forwarding functions that lead to even more prevalence of pollution in the network. Worse yet, when finally these polluted packets reach to the sink nodes they are not decodable and should be dropped. This disruptive and epidemic effect of packet corruption leads to network resource exhaustion and it must be resolved. A possible solution to mitigate the resource exhaustion attack is applying homomorphic hash functions [97] in which all the intermediate nodes can verify the validity of the encoded packets *on-the-fly* prior to recoding them without knowing the content of native data packets. Therefore, the intermediate node can detect the polluted packets and drop them.

Packet pollution may originate unintentionally like jamming or interference. Resource exhaustion attack can also be customized and modified to the new form of attacks like DoS, jamming, and SYN flooding attacks that were described before.

E. Blackhole, selective forwarding/dropping, and wormhole

A blackhole attacker may exploit routing protocols to advertise itself as a valid -and usually the shortest- path to a destination. This leads to position itself in the path of data packets toward that destination [46]. Then, the attacker can intercept/eavesdrop data traffic, or as a blackhole attack simply deny the routing operations like packet forwarding, that leads to decreasing the network performance.

Both state-aware and stateless protocols suffer from performance degradation caused by this attack; however, state-aware NC protocols can slightly heal blackhole backwashes via using local optimization techniques. In this regard, [98, 99] proposed an *Algebraic watchdog* (explained in Section 4.2) for NC aware wireless networks in which nodes can identify malicious behaviors probabilistically and monitor their downstream neighbors locally using overheard messages.

Finding an efficient, scalable, and extremely lightweight solution for blackhole attack in the NC aware protocols is an ongoing research. Also, blackhole attack can easily lead to the first step of further attacks by a malicious node like man-in-the-middle, route

poisoning, pollution, and DoS attacks. It also can appear in more subtle forms like selective forwarding and dropping and also wormhole attacks.

Selective forwarding or dropping attacker selectively forwards some packets and drops the others. Refusing to forward packets and dropping them in intermediate nodes results in lack of sufficient coefficients for decoding in sink nodes and render the native packets non-decodable. Figure 7.(b) shows the adversary nodes E that selectively forwards some packets and drops the others. NC aware systems may use an algebraic watchdog mechanism [98, 99], compel source node to generate more coded packets to increase the chance of delivery [100], and find several paths for each source-destination pairs to alleviate disruptive consequences of selective forwarding and dropping attacks. However, in other hand, all these mechanisms may lead to more overhead and thus nullify the throughput gain of NC.

Wormhole attack, is a severe active attack that needs two colluding adversaries: the first malicious node can receive packets from well behaved network nodes and tunnels to another malicious node [63]. One solution that is proposed by [101], estimates the distance between the sender and the receiver for detecting the fake neighbors and links. Also, [102] proposes a distributed detection defense technique by exploring the change of the data flow directions of the innovative packets originated by wormholes and show that the robustness of proposed technique relies on the node density in the network. To the best of our knowledge, there is no ultimate and efficient solution for the Wormhole attack in current NC aware link state routing protocols.

F. Byzantine Fabrication

A Byzantine fabrication attacker generates messages containing false information. It can disrupt the routing operation of network in different ways such as forwarding data packets through non-optimal or even invalid routes and generating routing loops. Also, this attack may appear in forms of modifying and/or altering packet headers, routing table overflow, route poisoning, and ACK pollution.

In state-aware NC protocols, packet headers normally contain topology states and routing information. Also, in stateless NC protocols, headers normally contain required decoding vectors. Therefore, the Byzantine fabrication attacks are disruptive for both stateless and state-aware NC protocols. Some of the most important Byzantine fabrication attacks include:

- **Routing table overflow**

Similar to the traditional MANET protocols, NC aware routings can apply proactive (table driven) and reactive (on-demand) routing approaches. In proactive protocols, in which routing techniques are either link-state or distance vector, nodes try to find all possible routes for each source-sink pairs regardless of the use or need of such routes. But in reactive protocols, in which routing techniques are either source routing or hop-by-hop, routes are created only when a source node requests and requires them [103]. A

malicious node, especially in a proactive approach, can advertise lots of new routes to the nonexistent nodes in the networks to overflow other node's routing tables.

- **Route poisoning**

Malicious nodes can continuously flood fake or invalid control packets (such as routing requests, replies, errors, and hello packets) into the network to perform a routing table poisoning attack. Also, leading network layer misbehaviors like dropping routing control packets, creating routing loops, extending or shortening service routes, mis-reporting in packet reception, increasing end-to-end delay, and link quality falsification or modification, the attackers can decrease quality of services or worse yet they can lead to network partitioning and DoS attack through mixing all these behaviors [45].

- **ACK pollution**

In an NC aware system, for each flow of data packets between a source-sink pair of nodes, the source node continuously flood coded packets toward sink node for the current generation until an acknowledgment (ACK) is received from the sink [24]. This ACK mechanism prepares an ideal situation for an attacker to perform a verity of misbehaviors.

The attacker can deceive the source node by creating and injecting fake ACKs or modify them leading the source node move onto the next generation. Therefore, the sink may not receive all required batches for decoding the whole generation of coded packets. Cryptography mechanisms, like digital signatures, can deal with this attack. Similar to blackhole and wormhole attackers, malicious forwarder nodes can drop ACK packets; this leads the source node keeps on sending coded packets from the current generation forever. Solutions for alleviating the negative effect of blackhole and wormhole attacks were mentioned in their related sections. Malicious node can also make a more subtle attack by ACK latency leading to degrade of network throughput. Source and sink nodes can ward off this attack by setting a timeout for each ACK [24].

ACK pollution attacks can be generally handled by packet authentication and cryptography mechanisms; however, these mechanisms are not suitable for those ACK pollution attacks that target the network quality of services (QoS). [104] propose a multipath acknowledgment mechanism for handling these types of attacks against ACK packets.

State-aware NC protocols are more susceptible to ACK pollution attacks because nodes store network topology and link state information in their routing cache or table. [24, 104] are two infrequent papers referring to attacks against the ACK packets in an NC aware system.

G. Byzantine Modification and Pollution Attack

An adversary node may perform invalid coding operations on the transit packets and modify them incorrectly to perform a Byzantine modification attack. Many of the previously discussed attacks like wormhole, blackhole, selective forwarding and dropping attack, man-in-the-middle, link spoofing, routing attacks, and repudiation can

be considered as a special type of Byzantine attacks. Figure 7.b shows a possible scenario for a Byzantine attacker in which the adversary node C, is supposed to perform valid XOR operation on the received packets, create correct packets like $a \oplus b$, and eventually forward them toward downstream nodes. But it wrongly modifies the native packets, modifies the XOR packets, and instead of valid XOR operation, performs other invalid process on them.

Figure 7.(a) shows a possible scenario for packet corruption attack in which the adversary node E, receives some XOR packets like $a \oplus b$ from upstream node C that should be forwarded toward sink nodes D_1 and D_2 but it bogus fake and corrupted packets like c and forwards them toward sink nodes.

MinJi et. al [105] studied three different schemes in NC aware systems for detecting Byzantine attackers: end-to-end error correction (like [106-108]), packet-based Byzantine detection scheme (like [97, 109-111]) , and generation-based Byzantine detection scheme (like [112]). They compared the transmission overhead of these schemes at a node.

Byzantine modification and pollution attacks are the most important active attacks and probably, beside eavesdropping attacks, are three most studied security attacks in NC systems. We study them in the coming sections by modeling these attacks and reviewing current mechanisms and schemes for handling them.

H. Impersonation

An impersonate attacker by a bogus authenticity can send messages pretending to be another legitimate node. By this bogus authenticity it may trigger a set of misbehaviors from simple eavesdropping of transitive data to severe pollution attacks such as route conflicts and loops, link spoofing, and network partitioning. Authentication in a network may involve three properties [43]:

- *Data integrity*: This property refers to the data that have not been changed, destroyed, or lost in a malicious or accidental manner.
- *Data origin authentication*: It verifies and validates the identity of the origin of the data.
- *Nonrepudiation*: It defines a security service that prevents an entity from denying previous obligations or actions, like guaranteeing that neither the origin of the data can later deny having originated and sent it nor the receiver can deny the reception.

Cryptographic mechanisms can be a possible solution for authentication issues. These mechanisms can be [44]:

- *Unconditional secure* which means they are robust even against a powerful attacker that has unlimited computational resources, and data packets in these techniques can be verified only by intended receivers.

- *Computationally secure* which means they are vulnerable against a powerful attacker that has unlimited computational resources, and data packets in these techniques can be verified by anyone who has a public verification algorithm.

To achieve three mentioned authentication properties, the data messages can be appended by an augmented packet such as:

- A digital signature: provides all three authentication properties and can be computationally or unconditionally secure [97, 109-111, 113-118].
- A Message Authentication Code (MAC): does not cover non-repudiation property and it is computationally secure [119-125].
- An authentication code (also called tag): does not cover non-repudiation property and it is unconditionally secure [44, 126].

Authenticity is important for all source-sink communications in traditional networks. However, due to the role of intermediate nodes even in a peer-to-peer source-sink communications in coded networks, authenticity turns to a critical and required key feature for NC aware systems [127-129].

Unlike stateless NC protocols that do not rely on the identity of the nodes for coding operations, in state-aware protocols, nodes are dependent to their neighbours for obtaining the state and network topology information which can lead to the severe attacks of impersonate adversary nodes [41]. Lack of authentication mechanism in network may lead to several forms of attacks such as Sybil or Man-in-the-middle. In Sybil attack, a malicious node behaves as if it was a larger number of nodes, for example by impersonating other nodes or simply by claiming false identities. In the worst case, an attacker may generate an arbitrary number of additional node identities, using only one physical device [76]. The Sybil attack can defeat the redundancy mechanisms of distributed storage systems, pose a threat to routing mechanisms in cooperative networks, trick reputation and voting systems, and manipulate resource allocation [21].

In Man-in-the-middle attack, a malicious node can lie on a data flow between the sender and receiver and then, by using link spoofing techniques (such as advertising fake links and sending routing control packets, including wrong information), impersonates other nodes and relays received messages. Therefore, being unaware of the attacker, the victims believe that they are communicating directly with the correct node. Man-in-the-middle threats can also be the first step the attacker takes to commit further misbehaviours like blackhole and routing attacks.

Applying strong cryptographic authentication mechanisms [109-111, 113-124, 130] can mitigate impersonation attacks; however, they may have some flaws like disability in covering all authentication properties (data integrity, data origin authentication, and non-repudiation), requiring a secure channel and a trusted third authority party for generating and distributing the keys among the nodes before establishing a trusted communication, or leading to the resource exhaustion, for example in a coded mesh

network with battery and processing power limited nodes. Accordingly, designing an efficient lightweight authentication scheme is extremely required. Algebraic watchdog or [131] which are based on physical layer NC mechanism can be useful for handling this type of impersonate attacks.

I. Entropy

While eavesdropping or pollution attacks have received a great deal of consideration, there is another attack which has not been studied extensively enough: entropy attack. It happens when an adversary intermediate node generates *valid but non-innovative* packets that are trivial linear combinations of the stale packets, stored or overheard at an earlier time by the attacker [132].

The simulation results, presented in [133], show the impact of entropy attacks and differentiate between local entropy (sending non-innovative packets to the local neighbouring nodes) and the more subtle one called global entropy (sending seemingly innovative packets to the local neighbouring that are non-innovative for one or more distant downstream node). These valid but non-innovative packets decrease the decoding opportunities at sinks in both stateless and state-aware NC systems, waste network resources, and eventually degrade the overall throughput rate [132, 133].

[133] proposed a mechanism for mitigating the effect of local entropy. Two other infrequent studies [109, 132] consider neither the impact of entropy attack on routing nor the possibility of a global entropy. However, [134] proposes an efficient edge-based authentication scheme by changing random linear coding to deterministic message combining rule. Also, [135] propose a new protocol that enables any node in the network to run a “verification test” as a technique against byzantine attacks but it can be used as a mitigation technique against entropy attacks in NC based communication systems too.

Table 2 lists a summary of different threats and attacks for NC enabled systems.

Table 2. A summary of different threats and attacks for NC enabled systems

Attack Name	**Attack Taxonomy**	**Description**	**Countermeasures**
Eavesdropping	-Passive -Multiple Layers -State-aware & less	The attacker eavesdrops transmitted packets and tries to obtain sensitive information	Several works studied Eavesdropping attack [9, 10, 12, 80, 82, 136-143] especially on a wiretap channel [9, 10, 137, 140-143]. Some solutions are: ***homomorphic hash functions*** [97, 109, 144-146], ***homomorphic digital signatures*** [110, 111, 113, 114, 147], ***symmetric keys*** [113, 119], and ***network codes*** [10, 70, 148-150].
Traffic Analysis and Monitoring	-Passive -Multiple Layers -State-aware & less	The attacker may monitor and analyse data traffic to obtain network topology related information and data packets.	It happens due to violating the privacy of nodes [73-75, 151]. ***Packet encoding*** and ***digital signatures*** can mitigate it.
Denial of Service (DoS)	-Active -Multiple Layers -State-aware & less	The attacker is able to deny services in the network. Denied services include, but are not limited to, routing, switching, name resolution, session establishment, processing and memory capabilities.	Investigating the handling mechanisms for traditional ad hoc networks [152-155]. [78] shows severe impact of DDoS on NC. ***Statistical analysis***, ***authentication and verification*** of traffic flows, and ***packet leashes*** are some possible mitigation techniques.
Jamming	-Active -Physical & MAC -State-aware & less	The attacker prevents the reception (jam) of valid packets, stops forwarding packets towards a destination or injects lots of fake packets into the network.	A polynomial-time rate-optimal distributed ***network codes*** [80], ***Null Keys*** [83], Jamming Evasive NC Aware Algorithm(***JENNA***) [81], and more [82, 84].
SYN Flooding	-Active -Transport - State-aware & less	The attacker generates lots of half-opened TCP connection with some victim nodes but never fulfills the handshaking procedure to fully open the connections.	[32] showed NC mechanism is possible in TCP communications and then several works developed after that [85-96]. However SYN attack is an infrequent study.
Resource Exhaustion	-Active -Network - State-aware & less	Attacker overwhelms victims by sending a huge amount of junk data packets for recoding consuming network resources and eventually rendering the network inoperable.	Can be customized and modified into several forms of attacks such as DoS, jamming, and SYN flooding. ***Node and data-origin authentication***, ***data integrity and confidentiality***, and

Attack Name	Attack Taxonomy	Description	Countermeasures
			homomorphic hash functions are possible mitigation techniques.
Blackhole, selective forwarding/dropping, and wormhole	-Active -Network - State-aware & less	Nodes are convinced to relay data packets through the attacker, who then refuses to forward them.	Some works proposed ***algebraic watchdog*** for mitigating this attack [98, 99, 156]. also, **estimating the distance** between the sender and the receiver is proposed for wormwhole attack in NC systems [101].
Byzantine Fabrication	-Active -Multiple Layers - State-aware & less	The attacker may create and send invalid messages containing false routing information, leading to routing loops, routing table overflow, route poisoning, and ACK pollution.	Byzantine fabrication attacks are investigated under Byzantine modification [21, 24, 44, 78, 97-99, 105-113, 117, 124, 136, 146, 156-170] in the literature, however some of them are not frequently studied like routing table overflow, route poisoning, and ACK pollution. ***Node and data-origin authentication*** and monitoring neighbours via ***intrusion detection and prevention*** mechanisms are some mitigations techniques for handling these attacks.
Byzantine Modification and Pollution	Active -Multiple Layers - State-aware & less	The Byzantine attacker invalidly or incorrectly modifies and alters the message in transit or it bogus fake packets and injects these polluted packets into network.	Byzantine modification [78, 82, 105, 112, 117, 136, 146, 156, 160, 162, 163, 165, 169, 171-177] and pollution attacks [21-23, 44, 113, 116, 119, 124, 145, 166-168, 178] are the most important and studied active attacks. ***Network codes***, ***error detecting and error correcting*** mechanisms, and ***intrusion detection/prevention*** technique can mitigate this attack.
Impersonation	-Active -MAC & Network -State-aware	The attacker sends messages pretending to be another legitimate node.	Using strong ***authentication mechanisms*** (such as digital signature [97, 109-111, 113-118], MAC [119-124, 130], and tag [44, 126, 179]) and other ***cryptographic mechanisms*** can alleviate this attack.
Entropy	-Active -Multiple layers	The attacker resends valid but non-innovative packets	The simulation results presented in [133] showed

Attack Name	Attack Taxonomy	Description	Countermeasures
	- State-aware & less	(like the old packets which were previously accepted in the network) aiming to exhaust network resources.	disruptive impact of entropy attacks, however it is not a frequently studied attack [109, 132, 133]. ***edge-based authentication scheme*** [134] and ***verification test*** [135] are two possible mitigation techniques.

4 Security Mechanisms Taxonomy

This section presents some mechanisms that can be used by NC to handle some security attacks that have been reviewed in the previous sections.

4.1 *Security via Network Codes*

Although NC mechanism for using intermediate nodes in packet coding may lead to several security issues; in the other hand, NC scheme itself has a security aware nature that can be useful for handling these attacks. For example, in RLNC, by applying a proper coding algorithm on the data packets, we can limit the capability of attackers to perform eavesdropping, since only the destination nodes who have access to sufficient decoding vectors can recover native packets [61, 62].

As some of the key research works, [106-108] proposed three distributed network codes, which are rate optimal and they run in polynomial time. The algorithms are secure against byzantine attacks of different strengths. The first algorithm considers an attacker in a secret shared scenario. On the other hand, the second assumes an omniscient attacker while the third is analysing the behaviours of an eavesdropper with limited power. In particular, both the first and the third schemes, by a lower complexity than the second one, achieve optimal rate $C - z$ where the network capacity is C and adversary can perform eavesdropping, jamming, or byzantine attacks on maximum z links; but the second one will be limited to the optimal rate $C - 2z$.

Another inherent property of NC that can be achieved by applying a proper coding is subspaces properties [162, 169, 180, 181]. In randomized NC, due to the intermediate nodes behaviour in randomly choosing and mixing incoming packets and sending the recoded packet toward their neighbours, always a random subspace of the space spanned by the source packets will be collected by intermediate nodes. These random subspaces potentially and implicitly carry topological information about the network which leads to a good opportunity for several applications such as detecting topology inference, bottleneck discovery in peer-to-peer systems, and locating Byzantine attackers [169]. [165] via subspace properties, proposed a new homomorphic MAC

scheme (called SpaceMac) for expanding subspaces which can detect the precise location of all Byzantine attackers in intra-session NC systems.

In a system that errors occur frequently or there are several malicious nodes that inject a large number of polluted packets into the network, the capability of error correction in NC system can be degraded and overwhelmed [24].

4.1.1 Handling Eavesdropping Attack

An adversary intermediate node can perform an eavesdropping attack on the transit packets for extracting some unauthorized information. The problem of eavesdropping attack can be modeled as follows [20, 61]:

1. Suppose Alice is the source node and she wants to send the original coded packet *x* composed of *N* symbols to the sink node.
2. Bob is the sink node and he receives a coded data packet $\boldsymbol{y}$ composed of *N* symbols.
3. Calvin, who is the malicious node, eavesdrops the coded packet $\boldsymbol{z}$ composed of *R* symbols.

The coded packets $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ can be represented as:

$$\boldsymbol{x} = (x_1\ x_2\ \dots\ x_N)$$

$$\boldsymbol{y} = (y_1\ y_2\ \dots\ y_N)$$

$$\boldsymbol{z} = (z_1\ z_2\ \dots\ z_R)$$

Suppose that the source node in the eavesdropping attack model, sends *N* packets to the sink node and the malicious node can eavesdrop *k* out of *N* packets. Without NC and by applying a mechanism like using a $(N, N-k)$ linear *Maximum Distance Separable* (MDS) code [180, 182], the source node still has a chance to securely send $(N, N-k)$ packets to the sink. If NC is used, due to vulnerabilities associated to the linear operations on the packets, we will miss this chance even if the secure wiretap channel code is applied [61].

To achieve a proper linear network code that guarantees the secure delivery of at least $N-k$ $(k < N)$ packets in a wiretap channel, [10] proposed a communication model based on linear network codes who can work in a wiretap channel called *k-secure* NC – this was later improved in [141]. A stronger linear network code scheme was proposed by [148] and [149]; it illustrated if input packets are encoded properly, then the *k-secure* NC for a wiretap channel can always be achievable.

The information is considered *theoretically secure* if the applied NC mechanism guarantees that an eavesdropper cannot extract any information about $\boldsymbol{x}$ from $\boldsymbol{z}$ in a sense that [20]:

$$\mathcal{P}(\boldsymbol{x}|\boldsymbol{y}) = 0 \text{ AND } \mathrm{I}(\boldsymbol{x}, \boldsymbol{z}) = 0 \qquad (1)$$

Where $\mathcal{P}(x|y)$ is the conditional entropy of *x* given *y* and $\mathrm{I}(x,z)$ is the mutual information between *x* and *z*. A *weakly secure* NC can be achieved by modifying the strong assumption of (1) in the following way:

$$\mathcal{P}(\mathbf{x}|\mathbf{y}) = 0 \text{ AND } I(\boldsymbol{x}_i, \mathbf{z}) = 0, \text{ where: } (i=1, 2, \ldots, N) \quad (2)$$

In weakly secure NC, the eavesdropper cannot extract any meaningful information about native data packets from coded packets, due to insufficient encryption coefficients. [70, 150] showed that the source node can deliver *N* packets to the sink node in a weakly secure NC scheme.

Several solutions [9, 10, 12, 80, 82, 136-143] have been proposed to handle eavesdropping attacks in NC systems especially in a wiretap channel [9, 10, 137, 140-143]. NC, itself, due to using packet coding, is one of these solutions [10, 70, 148-150].

4.1.2 Correcting Corrupted Packets (Error Correction)

By using appropriate network codes, it is possible not only to detect corrupted packets but also to correct them, mitigating a significant part of most well-known active attacks. However, error detection and error correction (erasure) mechanisms for NC systems, may lead to some undesired problems. An error detection scheme creates monitoring overhead and error correction is possible only after occurring pollution attacks, which may bring about epidemic disruptive problems for an NC system.

An adversary intermediate node can show a wide range of Byzantine misbehaviors like Byzantine modification on the transit packets or it injects bogus packets into the network as a part of a pollution attack. Pollution attacks can be modeled based on three elements: source node, sink node, and malicious node.

1. Alice is the source node and she wants to send *N* packets to the sink node. Each packet is composed of *L* symbols as the plain text and *N* symbols as the code.
2. Bob is the sink node and he wants to receive *N* packets (like before, composed of *N+L* symbols) from the source node.
3. Calvin is malicious node and he injects *R* polluted packets (composed of *N+L* symbols) into the Alice-Bob communication.

Therefore, Alice sends matrix $\boldsymbol{A}$ to Bob, Calvin injects polluted packets in the form of matrix $\boldsymbol{C}$, and finally Bob will receive matrix $\boldsymbol{B}$ which is an unknown combination of two matrices $\boldsymbol{A}$ and $\boldsymbol{C}$. Also $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are *N×(N+L)*, *N×(N+L)*, and *R×(N+L)* matrices, respectively. The coded packets $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ can be represented as:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ . \\ . \\ . \\ \boldsymbol{a}_N \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ . \\ . \\ . \\ \boldsymbol{b}_N \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ . \\ . \\ . \\ \boldsymbol{c}_R \end{bmatrix}$$

The coded packet ($\boldsymbol{B}$), received at sink node, is:

$$\boldsymbol{B} = \boldsymbol{W}\boldsymbol{A} + \boldsymbol{W}'\boldsymbol{C} \quad (3)$$

Where $\boldsymbol{W}$ is a *N×N* coding matrix related to $\boldsymbol{A}$ which is calculated based on the paths from Alice to Bob. It should be stated that these paths exclude the paths that contain Calvin, because he will not forward correct packets. Also $\boldsymbol{W}'$ is a *N×R* coding matrix

related to $\boldsymbol{C}$ which is calculated based on the paths from the malicious node (Calvin) to the sink node.

When the sink node in the pollution attack model receives a corrupted packet, it still has a chance to remove it. Suppose that each packet $\boldsymbol{d}_i$ is composed of L symbols in a sufficiently large finite field GF(q):

$$\boldsymbol{d}_i = [\boldsymbol{d}_i^1 \quad . \quad . \quad . \quad \boldsymbol{d}_i^L]$$

Let $\boldsymbol{D}$ be a $N \times L$ matrix composed of N packets like $\boldsymbol{d}_i$ ($i=1,2,\ldots,N$):

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{d}_1 \\ . \\ . \\ . \\ \boldsymbol{d}_N \end{pmatrix} = \begin{bmatrix} d_1^1 & . & . & . & d_1^L \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ d_N^1 & . & . & . & d_N^L \end{bmatrix}$$

Where $\boldsymbol{D}$ (a $N \times L$ matrix) represents real data or plaintext composed of native packets. Also suppose that $\boldsymbol{J}$ is an $N \times N$ unit matrix, consisting of all 1s. By appending matrix $\boldsymbol{D}$ and J, we obtain a $N \times (N + L)$ matrix, like $\boldsymbol{A}$:

$$\boldsymbol{A} = (\boldsymbol{D} \quad \boldsymbol{J}) = \begin{pmatrix} \boldsymbol{a}_1 \\ . \\ . \\ . \\ \boldsymbol{a}_N \end{pmatrix} = \begin{bmatrix} d_1^1 & . & . & . & d_1^L & 1 & . & . & . & 1 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ d_N^1 & . & . & . & a_N^L & 1 & . & . & . & 1 \end{bmatrix}$$

We can randomly choose N symbols like s_k ($k=1,2,\ldots,N$) from GF(q) and create a parity check matrix like $\boldsymbol{H}$ using these N symbols. $\boldsymbol{H}$ is a $L \times \delta$ matrix where δ is a design parameter. So the (i,j)th element of matrix $\boldsymbol{H}$ is s_j^i where ($i=1,2,\ldots,\delta$) and ($j=1,2,\ldots,L$).,

$$\boldsymbol{H} = \begin{bmatrix} s_1^1 & . & . & . & s_1^\delta \\ . & . & . & . & . \\ . & . & s_j^i & . & . \\ . & . & . & . & . \\ s_L^1 & . & . & . & s_L^\delta \end{bmatrix}$$

By using native data packets matrix D and parity check matrix $\boldsymbol{H}$, source node is able to obtain a hashed information in the form of an $N \times \delta$ matrix like $\boldsymbol{P}$:

$$\boldsymbol{P} = \boldsymbol{D}\boldsymbol{H} \tag{4}$$

Now that all required concepts were presented, the attack anatomy can be described like this:

1. Alice generates native data packets matrix $\boldsymbol{D}$ and creates unit matrix $\boldsymbol{J}$ to calculate augmented data packet matrix $\boldsymbol{A}$. Alice also creates parity check matrix $\boldsymbol{H}$ and N symbols (s_k);
2. Using a hash function, Alice produces matrix $\boldsymbol{P}$;
3. Alice sends matrix $\boldsymbol{A}$ to the sink node (Bob);

4. Using a secure channel, Alice sends matrix $\boldsymbol{P}$ and s_k ($i=1,2,\ldots,N$) symbols to Bob.
5. Calvin (the malicious node) starts a pollution attack by injecting polluted packets in the form of matrix $\boldsymbol{C}$ into the Alice-Bob communication.

Bob will receive matrix $\boldsymbol{B}$ and $\boldsymbol{P}$ and also the s_k symbols.

As a result, matrix $\boldsymbol{P}$ will be the hashed information that is sent from Alice to Bob (e.g., via a secret channel) and it contains the information about the native data packets ($\boldsymbol{D}$). Matrix $\boldsymbol{C}$, as polluted packets, was created by Calvin and was injected into the Alice-Bob communication. Finally, Bob will receive matrix $\boldsymbol{B}$ from Calvin and $\boldsymbol{P}$ from Alice.

Now Bob takes decoding steps to decode matrix $\boldsymbol{B}$ in order to extract the native data packets. Matrix $\boldsymbol{B}$ and $\boldsymbol{C}$ in equation (3) are composed of a hash data and coded header so that:

$$\boldsymbol{B} = (\boldsymbol{B}_1 \quad \boldsymbol{B}_2) \tag{5}$$

$$\boldsymbol{C} = (\boldsymbol{C}_1 \quad \boldsymbol{C}_2) \tag{6}$$

Therefore (3), (5), and (6) yield:

$$\boldsymbol{B}_1 = \boldsymbol{W}\boldsymbol{D} + \boldsymbol{W}'\boldsymbol{C}_1 \tag{7}$$

$$\boldsymbol{B}_2 = \boldsymbol{W}\boldsymbol{J} + \boldsymbol{W}'\boldsymbol{C}_2 = \boldsymbol{W} + \boldsymbol{W}'\boldsymbol{C}_2 \tag{8}$$

Because:

$$\boldsymbol{W} = \boldsymbol{B}_2 - \boldsymbol{W}'\boldsymbol{C}_2 \tag{9}$$

$\boldsymbol{B}_1$ *can* be rewritten like:

$$\boldsymbol{B_1} = \boldsymbol{B_2}\boldsymbol{D} + \boldsymbol{U}_1 \tag{10}$$

Where $\boldsymbol{U_1}$ is an unknown $N\times L$ matrix like:

$$\boldsymbol{U_1} = \boldsymbol{W}'(\boldsymbol{C}_1 - \boldsymbol{C}_2\mathbf{D}) \tag{11}$$

This unknown matrix can be obtained by post-multiplying both sides of (11) by $\boldsymbol{H}$ and using (4):

$$\boldsymbol{U}_1\boldsymbol{H} = \boldsymbol{B_1}\boldsymbol{H} - \boldsymbol{B}_2\boldsymbol{P} \tag{12}$$

And by 12 the sink node can obtain $\boldsymbol{U_1}$ and from (10) it can finally obtain matrix $\boldsymbol{D}$ which is real data and includes native data packets [174, 183].

[136]showed that considering equation 12, the probability of finding $\boldsymbol{U_2}$ in a sense that:

$$\boldsymbol{U}_2 \neq \boldsymbol{U}_1 \quad \& \quad \boldsymbol{U}_2\boldsymbol{H} = \boldsymbol{U}_1\boldsymbol{H} \tag{13}$$

Will be like the following equation ([136], Claim 5):

$$p \leq \left(\frac{N}{q}\right)^{\delta} \tag{14}$$

So if q, which is the size of filed **GF**(q), is large enough, then p will be very small and so the equation 14 has a unique solution [20].

4.2 *Security via Cooperative Mechanisms*

In an NC system, an intermediate node in a data flow, may have a simple *store-forward* role in which the *non-recoding* intermediate node simply forwards packets. In the other hand, the intermediate node, depending on its position in the network topology or NC mechanism, can be also a *recoding* node that performs *store-process-forward* paradigm: it receives several packets from upstream nodes and, via different paths, mixes and recodes them into one packet, and forward it to downstream nodes.

In information theoretic approaches [112, 171], intermediate nodes may simply insert some redundant decoding information into packets, recode packets, and forward them toward sink nodes. Then, by means of this redundant information, the sink node, the only one being responsible to verify the received packets, can detect corrupted packets, recover or correct them if possible, and drop the unrecoverable or uncorrectable packets [24]. Therefore, one corrupted packet epidemically may lead to several corrupted packets and can severely degrade the throughput of the network.

A customized version of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) for store and forward networks like MANETs [184-186] could be helpful for NC systems too due to store and forward property of NC. The measures for detecting misbehavior of adversary nodes can be any of the security attacks like fabrication, jamming, man-in-the-middle, impersonation, route poisoning, wormhole, and blackhole. Several dynamic, autonomous, distributed, and self-healing mechanisms are proposed to detect bogus injected or corrupted packets [109, 167, 168, 187-191].

One of those mentioned mechanisms is algebraic watchdog. Like traditional watchdog [164] for MANETs [60], an upstream node in an NC aware protocol can run watchdog mechanism and overhear its *non-recoding* downstream neighbours and detect malicious behaviors in them, like packet dropping, false routing information, and packet modification. After monitoring and detection phase, nodes will inform each other about these malicious nodes and finally they run isolating phase. However, traditional watchdog fails in detecting misbehaviours of *recoding* malicious nodes in NC based systems. When benign upstream nodes forward packets toward a downstream attacker for more recoding operations, the attacker can perform invalid recoding operations. In this scenario, the benign upstream nodes may have no chance to detect these misbehaviors via overhearing, due to the lack of sufficient information for decoding the downstream packets flooded by downstream attackers [24]. As a result, it paves the way for the malicious node for creating a variety of misbehaviours.

In 2009, [98] customized traditional watchdog for MANETs and proposed the first version of a*lgebraic watchdog* for NC networks. In algebraic watchdog, nodes can verify their neighbours probabilistically and, by means of overheard messages, can police them locally. As the first step, algebraic watchdog introduces a graphical model to simulate the inference process by which the nodes monitor their downstream neighbors. Then, the graphical model will be mapped into the Viterbi algorithm, and the probabilities of misdetection and false detection will be calculated that, eventually,

leads into detecting the presence of a malicious downstream neighbor [98, 99, 156, 192].

4.3 *Cryptographic Schemes*

Cryptographic based defenses against pollution attacks in NC based communication systems include a wide range of solutions.

4.3.1 Keys, signatures, null space, and authentication verification

The inherent security, provided by some NC protocols, like RLNC, can be easily combined (strengthen) with the application of other security techniques, such as the use of different types of digital signatures (e.g., lattice signature [193]), and symmetric or public key encryption. A key management mechanism can be used to exchange shared keys with the sink nodes, which are used for the encryption of the coding coefficients [64, 70, 111, 194].

Also, using null space property of NC [83, 178] is another proposed technique where all network nodes can verify the integrity of a received batch of data by checking if it belongs to the subspace spanned by the source batch. In this scheme, every node in the network has a vector, called null key, orthogonal to all combinations of the source blocks. These null keys belong to the null space of the source batch and random combinations of them are distributed by the source before the start of communication.

Another type of the mentioned defense techniques against pollution attacks is DART that was proposed in [23, 195] and it provides a time-based authentication in combination with random linear transformations. Therefore, this work achieves a new approach by providing a computationally more lightweight scheme at the cost of making an additional assumption; the security of DART relies on *time asymmetry* that requires time synchronization among the nodes in the network. Also, it has an enhanced version, called EDART, based on the optimistic forwarding scheme that enables quick attacker isolation and, therefore, achieves higher performance.

4.3.2 Homomorphic Hash Functions (HHFs)

As mentioned in Section 3.2, under the resource exhaustion attack, NC systems are very vulnerable due to the coding role of intermediate nodes. Prolonging the verifying process of the encoded packets to the extent that they reach to the sink nodes can be potentially the main reason for intensifying the disruptive epidemic damage of pollution attacks in the NC systems.

This problem can be mitigated if intermediate nodes get able to verify polluted coded packets without knowing the native packets. Homomorphic hash functions [97, 109, 144-146] or homomorphic digital signatures [110, 111, 113, 114, 147] for NC, which, for the first time, were proposed in [97], has this helpful property for NC systems, however it is computationally expensive. Bellow, we discuss how homomorphic hash functions are used in detecting corrupted packets (error detecting).

Any hash function like $h(.)$ can map a normally large message like m into a typically small size output $h(m)$ and satisfies two properties: i) it is computationally very hard to reach m by having $h(m)$ and ii) finding m' in a scene that $h(m) = h(m')$ is very difficult. Beside these strong properties, Homomorphic hash functions have an additional property called *homomorphism*, in which hash of some native messages (like hash value of a linear combination of some messages in a RLNC system) is equal to combinations of the hashes of those messages [109].

More specifically, if m' is the linear combination of n messages like:

$$m' = \sum_{i=1}^{n} c_i m_i$$

Then the homomorphic hash value of this combination, i.e $h(m')$, is:

$$h(m') = \prod_{i=1}^{n} h^{c_i}\, {m_i}^{c_i}$$

Before computing hashes, the source should share some hashing parameters between the sink and the intermediate nodes that are shown in [97, 109].

Therefore, because of the *homomorphism* property of these hash functions, all intermediate nodes in the pollution attack model, presented in the previous section, are able to verify the validity of encoded packets *on-the-fly* prior to mixing them algebraically[39]. Now that the intermediate nodes can collaborate in verifying the transit packets, polluted packets will be dropped very soon and malicious nodes can be detected and isolated. Also, because of *homomorphism* property, the intermediate nodes can combine and encode the incoming hash packets and forward them without knowing the content of native packets or private key of the source node that prevents them from performing an eavesdropping attack.

In a very earliest attempts, [159] proposed a security hashing scheme based on homomorphic functions to validate blocks of rateless codes only at source and receiver nodes. So, Lemma 4.1 in [157] described how to compute the hash values to make it possible to check the correctness of both the encoded blocks and their coefficient vectors. On the other hand, [196] developed a different homomorphic signature scheme, which was not assuming the existence of a separate secure channel to transmit hash values to all nodes. In particular, the scheme is able to sign linear combination of packets without solving the linear system. [97] work was improved by [147] and, instead of pre-distributing a large number of hash values and evaluating every single packet, they proposed a new batch delivery and verification scheme in which the authentication information of a message is embedded with the message.

Homomorphic hash functions can have other applications in the scope of NC based communication systems too. [97] proposed homomorphic hash function for general content distribution networks that enables a downloader to perform *on-the-fly* [157, 197] verification of erasure-encoded blocks. [187] focused on finding a lightweight mechanism for identifying the malicious nodes in P2P streaming via hash functions too.

Homomorphic hash functions may have several drawbacks. It is computationally expensive and several works reported the poor performance of the scheme even by applying powerful CPUs [109, 198]. Also, the hash function parameters should be distributed among the network nodes before the communication and, in some implementation, a secure channel is needed for this purpose [199]. Using symmetric keys [113, 119] and batch verification can alleviate computational complexity of homomorphic hash functions.

It is noteworthy to mention that all these cryptographic mentioned techniques *may* lead to traffic overhead or they need frequently pre-distributing verification codes between nodes in a communication or require a large bandwidth [187]. Developing an extremely efficient lightweight security mechanism for NC still is required and it is an ongoing research. A classification of these researches which represents the security taxonomy in network coding systems is shown in figure 8.

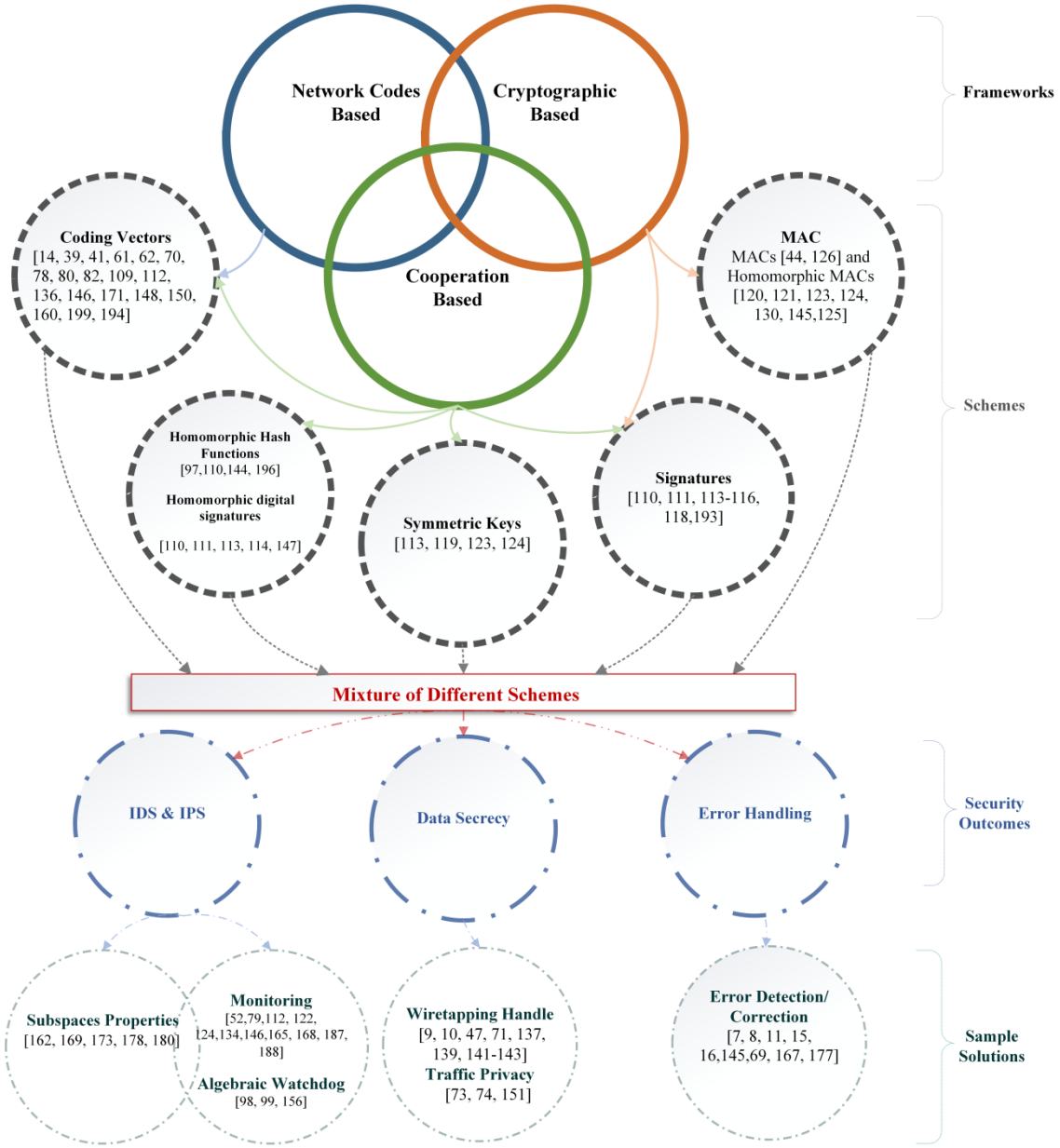

Figure 8: Security Taxonomy in Network Coding Systems

5 Conclusions and future directions

At the best of authors' knowledge, this is the first survey to include the most relevant literature in the diverse research areas related to security attacks and mitigation techniques in network coding based approaches. The Paper set out with the main fundamental concepts in network coding theorem and security challenges that any network coding based system should consider them.

In the first section, we started by introducing the secure network coding concept and current survey structure. Section 2 presented the principles of RLNC and three rules that a well behaved node should follow - when a node violates one or more of these rules, the security is compromised. This section also reviewed the features of stateless and state-aware NC protocols. Section 3 showed that most of the security threats and attacks commonly found in other types of networks also threat NC enabled systems. On top of these, there are other attacks, exclusive to NC, such as the pollution attacks - the data mixing operations performed by the intermediate nodes clearly lead to vulnerabilities. There are a few exclusive attacks threatening NC systems; *pollution* attacks are the most studied. The pollution of packets is considered highly problematic in NC systems because of the spread velocity, since it only takes one incoming corrupted packet to cause a well behaved node to produce a corrupted output. To deal with this threat and others, Section 4 provides three mitigation strategies, the anatomy (how it works), and mitigation techniques of the three attacks: *eavesdropping*, *Byzantine modification*, and *pollution* attacks. Section 5, presents the discussions and future directions in NC enabled systems and also presents a research and papers timeline.

Designing an efficient network code in the presence of all kind of adversaries and erroneous cannels in the network still needs furthered studies. There are several mechanisms for the on-the-fly verification of packets in intermediate nodes, prior to mixing algebraically and forwarding them. On-the-fly verification is possible via Homomorphic Hash Function (HHF) based approaches such as homomorphic digital signatures, Message Authentication Code (MAC), and authentication codes (tag). However more researches are required to find more efficient and lightweight homomorphic hash functions that are cheaper in terms of complexity.

Also, via other techniques such as using subspace properties of network coding or null keys, the intermediate node can detect and drop the corrupted packets. These techniques generally need a key distribution algorithm beside a secret channel for key exchange or appending the keys to the header of packets and protecting them via other techniques such as HHF.

Mixing RLNC and other error correcting codes and empowering them via *pre-coding* methods, such as Reed–Solomon (RS) code or Luby Transform (LT) [200], that is rateless erasure code which generates a variable quantity of encoding symbols according to the needs, is another green research field in the scope of network coding.

6 Acknowledgment

The research leading to these results has received funding from Fundação para a Ciência e Tecnologia (PTDC/EEA-TEL/119228/2010-SMARTVISION) and EC [FP7/2007-2013-285969 - CODELANCE] and partially financed by the Green Mobile Cloud project granted by the Danish Council for Independent Research (Grant No. DFF - 0602-01372B).

7 References:

[1] R. Ahlswede, C. Ning, S. Y. R. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on,* vol. 46, pp. 1204-1216, 2000.

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

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *Networking, IEEE/ACM Transactions on,* vol. 11, pp. 782-795, 2003.

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

[5] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," presented at the 51st Allerton Conf. Communication, Control and Computing,, Oct. 2003.

[6] C. Fragouli and J. W. Jean-Yves Le Boudec, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication,* vol. 36, pp. 63 - 68, 2006.

[7] N. Cai and R. W. Yeung, "Network coding and error correction," in *Information Theory Workshop, 2002. Proceedings of the 2002 IEEE*, 2002, pp. 119-122.

[8] N. Cai and R. W. Yeung, "Network error correction," in *Information Theory, 2003. Proceedings. IEEE International Symposium on*, 2003, p. 101.

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

[10] N. Cai and R. W. Yeung, "Secure Network Coding on a Wiretap," *IEEE TRANSACTIONS ON INFORMATION THEORY 1,* 2002.

[11] S. Lihua, R. W. Yeung, and C. Ning, "Zero-error network coding for acyclic networks," *Information Theory, IEEE Transactions on,* vol. 49, pp. 3129-3139, 2003.

[12] K. Jain, "Security based on network topology against the wiretapping attack," *Wireless Communications, IEEE,* vol. 11, pp. 68-71, 2004.

[13] J. Tan and M. Medard, "Secure Network Coding with a Cost Criterion," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, 2006, pp. 1-6.

[14] L. Lima, M. Medard, and J. Barros, "Random Linear Network Coding: A free cipher?," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, 2007, pp. 546-550.

[15] R. W. Yeung and N. Cai, "Network Error Correction, I: Basic Concepts and Upper Bounds," *Commun. Inf. Syst.,* vol. 6, pp. 19-36, 2006.

[16] N. Cai and R. W. Yeung, "Network Error Correction, II: Lower Bounds," *Commun. Inf. Syst.,* vol. 6, pp. 37-54, 2006.

[17] C.-K. Ngai and R. W. Yeung, "Secure error-correcting (SEC) network codes," in *Network Coding, Theory, and Applications, 2009. NetCod '09. Workshop on*, 2009, pp. 98-103.

[18] M. Di Renzo, M. Iezzi, and F. Graziosi, "Beyond routing via Network Coding: An overview of fundamental information-theoretic results," in *Personal Indoor and Mobile*

Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on, 2010, pp. 2745-2750.

[19] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network Coding Theory: A Survey," *Communications Surveys & Tutorials, IEEE,* vol. 15, pp. 1950-1978, 2013.

[20] T. MATSUDA, T. NOGUCHI, and T. TAKINE, "Survey of Network Coding and Its Applications," *IEICE TRANS. COMMUN,* vol. E94–B, 2011.

[21] D. Jing, R. Curtmola, C. Nita-Rotaru, and D. K. Y. Yau, "Pollution Attacks and Defenses in Wireless Interflow Network Coding Systems," *Dependable and Secure Computing, IEEE Transactions on,* vol. 9, pp. 741-755, 2012.

[22] J. Dong, R. Curtmola, C. Nita-Rotaru, and D. Yau, "Pollution Attacks and Defenses in Wireless Inter-Flow Network Coding Systems," in *Wireless Network Coding Conference (WiNC), 2010 IEEE*, 2010, pp. 1-6.

[23] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," presented at the Proceedings of the second ACM conference on Wireless network security, Zurich, Switzerland, 2009.

[24] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure network coding for wireless mesh networks: Threats, challenges, and directions," *Comput. Commun.,* vol. 32, pp. 1790-1801, 2009.

[25] D. Jing, R. Curtmola, R. Sethi, and C. Nita-Rotaru, "Toward secure network coding in wireless networks: Threats and challenges," in *Secure Network Protocols, 2008. NPSec 2008. 4th Workshop on*, 2008, pp. 33-38.

[26] N. c. website. Available: http://www.networkcoding.info/bibliography.php

[27] (2015). *Bibliography on Secure Network Coding (Maintained by Rongxing LU and Chen LI and Xiaoli XU and Xing ZHANG ed.)*. Available: http://www.ntu.edu.sg/home/rxlu/securenetworkcodingbib.htm

[28] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *SIGCOMM Comput. Commun. Rev.,* vol. 37, pp. 169-180, 2007.

[29] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," presented at the Proceedings of the ACM SIGCOMM 2008 conference on Data communication, Seattle, WA, USA, 2008.

[30] H. Tracey and H. Viswanathan, "Dynamic Algorithms for Multicast With Intra-Session Network Coding," *IEEE transactions on INFORMATION FORENSICS AND SECURITY,* vol. 55, pp. 797-815, 2009.

[31] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez, "An Optimization Framework for Opportunistic Multipath Routing in Wireless Mesh Networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, pp. 2252-2260.

[32] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *SIGCOMM Comput. Commun. Rev.,* vol. 36, pp. 243-254, 2006.

[33] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-optimal co-ordinated coding in wireless multihop networks," presented at the Proceedings of the 2007 ACM CoNEXT conference, New York, New York, 2007.

[34] S. SENGUPTA, S. RAYANCHU, and BANERJEE, "An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing," in *In Proceedings of the 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007)*, 2007.

[35] E. Rozner, A. P. Iyer, Y. Mehta, L. Qiu, and M. Jafry, "ER: efficient retransmission scheme for wireless LANs," presented at the Proceedings of the 2007 ACM CoNEXT conference, New York, New York, 2007.

[36] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," presented at the Proceedings of the 13th annual ACM international conference on Mobile computing and networking, Montréal, Québec, Canada, 2007.
[37] H. Tracey, M. Medard, R. Koetter, D. R. Karger, M. Effros, S. Jun, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *Information Theory, IEEE Transactions on,* vol. 52, pp. 4413-4430, 2006.
[38] L. Lima, J. P. Vilela, P. F. Oliveira, and J. Barros, "Network Coding Security: Attacks and Countermeasures," *IEEE,* 2008.
[39] L. Lima, "Network Coding Security: Algebraic Properties and Lightweight Solutions," PhD, Faculty of Science, Universidade of Porto, 2010.
[40] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*: Cambridge University Press, 2011.
[41] L. Lima, J. Vilela, P. Oliveira, and J. Barros, "Network Coding Security: Attacks and Countermeasures," *CoRR,* vol. abs/0809.1366, 2008.
[42] J. Krithiga and R. C. Porselvi, "Efficient CodeGuard mechanism against pollution attacks in interflow Network coding," in *International Conference on Communications and Signal Processing (ICCSP)*, 2014, pp. 1384-1388.
[43] F. Rodriguez-Henriquez, Saqib, N.A., Díaz Pérez, A., Koc, C.K., *Cryptographic Algorithms on Reconfigurable Hardware*: Springer, 2007.
[44] F. Oggier and H. Fathi, "An Authentication Code Against Pollution Attacks in Network Coding," *Networking, IEEE/ACM Transactions on,* vol. 19, pp. 1587-1596, 2011.
[45] G. Padmavathi and S. D., "A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks," *(IJCSIS) International Journal of Computer Science and Information Security,* vol. 4, 2009.
[46] M. e. Illyas, *The handbook of ad hoc wireless networks, chapter 30, security in wireless ad hoc networks*: CRC press Boca Raton, 2003.
[47] N. Cai and T. Chan, "Theory of secure network coding," *Proceedings of the IEEE,* vol. 99, pp. 421-437, 2011.
[48] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh 2006)*, 2006, pp. 157-159.
[49] W. Xin, Z. Li, X. Ji, and W. Qingyun, "Network Coding Aware Routing Protocol for Lossy Wireless Networks," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, 2009, pp. 1-4.
[50] Y. Yan, Z. Zhuang, Z. Baoxian, H. T. Mouftah, and M. Jian, "Rate-Adaptive Coding-Aware Multiple Path Routing for Wireless Mesh Networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008, pp. 1-5.
[51] J.-z. Sun, Y.-a. Liu, H.-f. Hu, and D.-m. Yuan, "On-demand coding-aware routing in wireless Mesh networks," *The Journal of China Universities of Posts and Telecommunications,* vol. 17, pp. 80-92, 2010.
[52] W. Yunnan, S. M. Das, and R. Chandra, "Routing with a Markovian Metric to Promote Local Mixing," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 2381-2385.
[53] Y. Yan, Z. Baoxian, H. T. Mouftah, and M. Jian, "Practical Coding-Aware Mechanism for Opportunistic Routing in Wireless Mesh Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, 2008, pp. 2871-2876.
[54] L. Jilin, J. C. S. Lui, and C. Dah-Ming, "DCAR: Distributed Coding-Aware Routing in Wireless Networks," *Mobile Computing, IEEE Transactions on,* vol. 9, pp. 596-608, 2010.

[55] Y. Lu, C. Shen, Q. Xia, and J. Tao, "ICM: A Novel Coding-Aware Metric for Multi-Hop Wireless Routing," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, 2009, pp. 1-4.

[56] G. Hui, Q. Yi, L. Kejie, and N. Moayeri, "Backbone Routing over Multihop Wireless Networks: Increased Network Coding Opportunity," in *IEEE International Conference on Communications (ICC), 2010*, 2010, pp. 1-5.

[57] J. Xianlong, W. Xiaodong, and Z. Xingming, "Active Network Coding Based High-Throughput Optimizing Routing for Wireless Ad Hoc Networks," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, 2008, pp. 1-5.

[58] H. Song, Z. Zifei, L. Hongxing, C. Guihai, E. Chan, and A. K. Mok, "Coding-Aware Multi-path Routing in Multi-Hop Wireless Networks," in *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, 2008, pp. 93-100.

[59] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of network coding-aware routing protocols in wireless networks," *Journal of Network and Computer Applications,* vol. 34, pp. 1956-1970, 2011.

[60] MANET, "Working group in IETF, http://datatracker.ietf.org/wg/manet," ed, 2013.

[61] C. Fragouli and E. Soljanin, "Network coding applications," *found and trends netw,* vol. 2, pp. 135-269, 2007.

[62] R. Koetter and F. R. Kschischang, "Coding for Errors and Erasures in Random Network Coding," *IEEE Transactions on Information Theory,* vol. 54, pp. 3579-3591, 2008.

[63] J. Barros, "Network Coding Security, Seminar on Security in the Information Society," University of Porto, Department of Computer Science, Porto2008.

[64] P. F. Oliveira and J. Barros, "A Network Coding Approach to Secret Key Distribution," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,* vol. 3, 2008.

[65] S. Jaggi, P. Sanders, P. A. Chou, Michelle Effros, S. Egner, K. Jain, and L. M. G. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 51, 2005.

[66] S. Katti, D. Katabi, W. Hu, H. S. Rahul, and M. M´edard, "The importance of being opportunistic: Practical network coding for wireless environments," presented at the In Proc. 43rd Annual Allerton Conf. Commun. Control Comput., 2006.

[67] L. TOLEDO and WANG, "Efficient Multipath in Sensor Networks using Diffusion and Network Coding," in *In Proceedings of the 40th Annual Conference on Information Sciences and Systems*, 2006.

[68] Hundeb, x00F, M. ll, 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 *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012, pp. 1-5.

[69] S. Jaggi, M. Langberg, T. Ho, and M. Effros, "Correction of adversarial errors in networks," in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, 2005, pp. 1455-1459.

[70] K. Bhattad and K. R. Narayanan, "Weakly Secure Network Coding," 2005.

[71] S. Y. El Rouayheb and E. Soljanin, "On Wiretap Networks II," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, 2007, pp. 551-555.

[72] K. Bhattad and K. R. Narayanan, "Weakly secure netwaork coding," in *in Proc. of the First Workshop on Network Coding, Theory, and Applications(NetCod)*, Riva del Garda, Italy, 2005.

[73] F. Yanfei, J. Yixin, Z. Haojin, C. Jiming, and X. S. Shen, "Network Coding Based Privacy Preservation against Traffic Analysis in Multi-Hop Wireless Networks," *Wireless Communications, IEEE Transactions on,* vol. 10, pp. 834-843, 2011.

[74] F. Yanfei, J. Yixin, Z. Haojin, and S. Xuemin, "An Efficient Privacy-Preserving Scheme against Traffic Analysis Attacks in Network Coding," in *INFOCOM 2009, IEEE*, 2009, pp. 2213-2221.

[75] H. Sousa-Pinto, D. E. Lucani, and J. Barros, "Hide and code: Session anonymity in wireless line networks with coded packets," in *Information Theory and Applications Workshop (ITA), 2012*, 2012, pp. 262-268.

[76] P. Jawandhiya, M. Ghonge, M. S. Ali, and J. S. Deshpande, "A survey of Mobile ad hoc network attacks," *Internationnal Journal of Engineering science and Technology,* vol. 2, pp. 4036-4071, 2010.

[77] V. Talooki, H. Marques, J. Rodriguez, x, H. gua, N. Blanco, and L. Campos, "An Energy Efficient Flat Routing Protocol for Wireless Ad Hoc Networks," in *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, 2010, pp. 1-6.

[78] L. Lima, J. Barros, and R. Koetter, "Byzantine attacks against network coding in peer to peer distributed storage," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, 2009, pp. 1164-1168.

[79] T. Truc Thanh and K. Hyung Yun, "An application of network-coding technique into cooperative jamming," in *27th Biennial Symposium on Communications (QBSC)*, 2014, pp. 218-222.

[80] Y. Hongyi, D. Silva, S. Jaggi, and M. Langberg, "Network Codes Resilient to Jamming and Eavesdropping," in *Network Coding (NetCod), 2010 IEEE International Symposium on*, 2010, pp. 1-6.

[81] A. Asterjadhi and M. Zorzi, "JENNA: a jamming evasive network-coding neighbor-discovery algorithm for cognitive radio networks [Dynamic Spectrum Management]," *Wireless Communications, IEEE,* vol. 17, pp. 24-32, 2010.

[82] S. Jaggi and M. Langberg, "Resilient network codes in the presence of eavesdropping Byzantine adversaries," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, 2007, pp. 541-545.

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

[84] W. Shanshan, Y. E. Sagduyu, Z. Junshan, and J. H. Li, "Traffic shaping impact of network coding on spectrum predictability and jamming attacks," in *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, 2011, pp. 293-298.

[85] H. Yong, M. Ghaderi, D. Towsley, and W. Gong, "TCP Performance in Coded Wireless Mesh Networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, 2008, pp. 179-187.

[86] C. Chien-Chia, C. Chen, P. Joon-Sang, S. Y. Oh, M. Gerla, and M. Y. Sanadidi, "Multiple network coded TCP sessions in disruptive wireless scenarios," in *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, 2011, pp. 754-759.

[87] L. Scalia, F. Soldo, and M. Gerla, "PiggyCode: A MAC Layer Network Coding Scheme to Improve TCP Performance Over Wireless Networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, 2007, pp. 3672-3677.

[88] P. Samuel David and A. Kumar, "Network coding for TCP throughput enhancement over a multi-hop wireless network," in *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, 2008, pp. 224-233.

[89] J. K. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network Coding Meets TCP: Theory and Implementation," *Proceedings of the IEEE,* vol. 99, pp. 490 – 512, 2011.

[90] S. Gheorghiu, A. Lopez, and T. P. Rodriguez, "Multipath TCP with Network Coding for Wireless Mesh Networks," in *IEEE ICC 2010 proceedings*, 2010.

[91] L. Hongquan, C. Jiong, and G. Yuantao, "A New Mechanism to Incorporate Network Coding Into TCP in Multi-radio Multi-channel Wireless Mesh Networks," in *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*, 2011, pp. 256-260.

[92] M. Kim, M. Medard, and J. Barros, "Modeling Network Coded TCP Throughput: A Simple Model and its Validation," in *IEEE INFOCOM*, 2011.

[93] H. Liu and Y. Gu, "TCP with hop-oriented network coding in multi-radio multi-channel wireless mesh networks," *Networks, IET,* vol. 1, pp. 171-180, 2012.

[94] T. Nage, F. R. Yu, and M. St-Hilaire, "TCP-Aware Network Coding with Opportunistic Scheduling in Wireless Mobile Ad Hoc Networks," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, 2010, pp. 1-5.

[95] H. M. Ruiz, M. Kieffer, and B. Pesquet-Popescu, "Redundancy adaptation scheme for network coding with TCP," in *Network Coding (NetCod), 2012 International Symposium on*, 2012, pp. 49-54.

[96] X. Zhuo-qun, C. Zhi-gang, M. Zhao, and L. Jia-Qi, "A Multipath TCP Based on Network Coding in Wireless Mesh Networks," in *Information Science and Engineering (ICISE), 2009 1st International Conference on*, 2009, pp. 3946-3950.

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

[98] M. Kim, M. Medard, J. Barros, and R. Kotter, "An algebraic watchdog for wireless network coding," presented at the Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 2, Coex, Seoul, Korea, 2009.

[99] M. Kim, M. Medard, and J. Barros, "Algebraic Watchdog: Mitigating Misbehavior in Wireless Network Coding," *Jou. of Sel. Areas in Comm.,* vol. 29, pp. 1916-1925, 2011.

[100] M. Chuah and P. Yang, "Impact of Selective Dropping Attacks on Network Coding Performance in DTNs and a Potential Mitigation Scheme," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, 2009, pp. 1-6.

[101] Z. Li, D. Pu, W. Wang, and A. Wyglinski, "Forced collision: Detecting wormhole attacks with physical layer network coding," *Tsinghua Science and Technology,* vol. 16, pp. 505-519, 2011.

[102] S. Ji, T. Chen, S. Zhong, and S. Kak, "DAWN: Defending against wormhole attacks in wireless network coding systems," presented at the INFOCOM, 2014 Proceedings IEEE, Toronto, ON, April 27 2014-May 2 2014.

[103] V. N. Talooki and J. Rodriguez, "Jitter based comparisons for routing protocols in mobile ad hoc networks," in *Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on*, 2009, pp. 1-6.

[104] Y. Zhang, W. Znaidi, C. Lauradoux, and M. Minier, "Flooding attacks against network coding and countermeasures," in *Network and System Security (NSS), 2011 5th International Conference on*, 2011, pp. 305-309.

[105] M. Kim, M. Medard, and J. Barros, "Counteracting Byzantine adversaries with network coding: An overhead analysis," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, 2008, pp. 1-7.

[106] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient Network Coding in the Presence of Byzantine Adversaries," in *INFOCOM 2007*, 2007.

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

[108] S. Jaggi and M. Langberg, "Resilient network codes in the presence of eavesdropping Byzantine adversaries," in *ISIT '07*, 2007.

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

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

[111] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," in *In Proc. of International Symposium on Information Theory (ISIT)*, 2007.

[112] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, 2004, p. 144.

[113] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Signature-based Scheme for Securing Network Coding against Pollution Attacks," in *IEEE INFOCOM*, 2008.

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

[115] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a Linear Subspace: Signature Schemes for Network Coding," in *12th International Conference on Practice and Theory in Public Key Cryptography, PKC '09*, 2009, pp. 68-87.

[116] S. Vyetrenko, A. Khosla, and T. Ho, "On combining information-theoretic and cryptographic approaches to network coding security against the pollution attack," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, 2009, pp. 788-792.

[117] M. Kim, L. Lima, Z. Fang, J. Barros, M. Medard, R. Koetter, T. Kalker, and K. J. Han, "On counteracting Byzantine attacks in network coded peer-to-peer networks," *Selected Areas in Communications, IEEE Journal on,* vol. 28, pp. 692-702, 2010.

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

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

[120] C. Cheng and T. Jiang, "An Efficient Homomorphic MAC with Small Key Size for Authentication in Network Coding," *Computers, IEEE Transactions on,* vol. PP, pp. 1-1, 2012.

[121] C. Chi and J. Tao, "A Novel Homomorphic MAC Scheme for Authentication in Network Coding," *Communications Letters, IEEE,* vol. 15, pp. 1228-1230, 2011.

[122] J. C. Corena and T. Ohtsuki, "A Multiple-MAC-Based Protocol to Identify Misbehaving Nodes in Network Coding," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012, pp. 1-5.

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

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

[125] L. Chen, L. Rongxing, L. Hui, C. Le, and L. Xiaoqing, "A Novel Homomorphic MAC Scheme for Authentication in Network Coding," *Communications Letters, IEEE,* vol. 18, pp. 2129-2132, 2014.

[126] F. Oggier and H. Fathi, "Multi-receiver authentication code for network coding," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, 2008, pp. 1225-1231.

[127] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE authentication for network coding," presented at the Proceedings of the 29th conference on Information communications, San Diego, California, USA, 2010.

[128] A. Fathy, T. ElBatt, and M. Youssef, "SANC: Source authentication using network coding," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, 2011, pp. 1012-1017.

[129] K. Jaffres-Runser and C. Lauradoux, "Authentication Planning for XOR Network Coding," in *Network Coding (NetCod), 2011 International Symposium on*, 2011, pp. 1-6.

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

[131] W. Weichao, P. Di, and A. Wyglinski, "Detecting Sybil nodes in wireless networks with physical layer network coding," in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, 2010, pp. 21-30.

[132] 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, pp. 3089-3101, 2009.

[133] A. J. Newell, R. Curtmola, and C. Nita-Rotaru, "Entropy attacks and countermeasures in wireless network coding," presented at the Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, Tucson, Arizona, USA, 2012.

[134] R. Iguchi and Y. Manabe, "An Efficient Edge-Based Authentication for Network Coding against Entropy Attacks," in *Distributed Computing Systems Workshops (ICDCSW), 2014 IEEE 34th International Conference on*, 2014, pp. 133-139.

[135] R. A. Popa, A. Chiesa, T. Badirkhanli, and M. Médard, "Going Beyond Pollution Attacks: Forcing Byzantine Clients to Code Correctly," *CoRR,* vol. abs/1108.2080, 2011.

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

[137] A. R. Hammons, Z. Qinqing, and B. Haberman, "On the Eavesdrop Vulnerability of Random Network Coding over Wireless Networks," in *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on*, 2009, pp. 201-207.

[138] A. R. Hammons Jr, Q. Zhang, and B. Haberman, "An Eavesdrop Vulnerability Analysis of Random Network Coding over Wireless Ad-Hoc Networks," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, 2010, pp. 1-6.

[139] G. Zhenzhen, Y. Yu-Han, and K. J. R. Liu, "Anti-Eavesdropping Space-Time Network Coding for Cooperative Communications," *Wireless Communications, IEEE Transactions on,* vol. 10, pp. 3898-3908, 2011.

[140] C. Ning and R. W. Yeung, "Secure Network Coding on a Wiretap Network," *Information Theory, IEEE Transactions on,* vol. 57, pp. 424-435, 2011.

[141] N. Cai and R. W. Yeung, "Secure Network Coding on a Wiretap Network," *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 57, 2011.

[142] S. El Rouayheb, E. Soljanin, and A. Sprintson, "Secure Network Coding for Wiretap Networks of Type II," *Information Theory, IEEE Transactions on,* vol. 58, pp. 1361-1371, 2012.

[143] C. Xiangmao, W. Jin, W. Jianping, V. Lee, L. Kejie, and Y. Yixian, "On Achieving Maximum Secure Throughput Using Network Coding against Wiretap Attack," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, 2010, pp. 526-535.

[144] M. Adeli and L. Huaping, "Secure network coding with minimum overhead based on hash functions," *Communications Letters, IEEE,* vol. 13, pp. 956-958, 2009.

[145] L. Anh and A. Markopoulou, "On detecting pollution attacks in inter-session network coding," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 343-351.
[146] T. Ho, L. Ben, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine Modification Detection in Multicast Networks With Random Network Coding," *Information Theory, IEEE Transactions on,* vol. 54, pp. 2798-2803, 2008.
[147] Q. Li, D.-m. Chiu, and J. C. S. Lui, "On the Practical and Security Issues of Batch Content Distribution Via Network Coding," presented at the Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols, 2006.
[148] k. Harada and H. Yamamoto, "Strongly secure linear network coding," *IEICE Trans. Fundamentals,* vol. E91-A, pp. 2720–2728, 2008.
[149] J. Feldman, T. M. Servedio, R. A., and C. Stein, "On the Capacity of Secure Network Coding," in *42nd Allerton Conf. Commun.,Control, and comput.*, Monticello, 2004.
[150] D. Silva and F. R. Kschischang, "Universal Weakly Secure Network Coding," in *ITW 2009, Volos, Greece*, 2009.
[151] W. Zhiguo, X. Kai, and L. Yunhao, "Priv-Code: Preserving privacy against traffic analysis through network coding for multihop wireless networks," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 73-81.
[152] G. Carl, G. Kesidis, R. R. Brooks, and R. Suresh, "Denial-of-service attack-detection techniques," *Internet Computing, IEEE,* vol. 10, pp. 82-89, 2006.
[153] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "DoS Attacks in Mobile Ad Hoc Networks: A Survey," in *Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, 2012, pp. 535-541.
[154] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," *Communications Surveys & Tutorials, IEEE,* vol. 13, pp. 245-257, 2011.
[155] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc networks," presented at the Proceedings of the 10th annual international conference on Mobile computing and networking, Philadelphia, PA, USA, 2004.
[156] M. Kim, Me, x, M. dard, and J. Barros, "A multi-hop multi-source Algebraic Watchdog," in *Information Theory Workshop (ITW), 2010 IEEE*, 2010, pp. 1-5.
[157] C. Gkantsidis and P. Rodriguez Rodriguez, "Cooperative Security for Network Coding File Distribution," Microsoft Research, Cambridge2006.
[158] C. Gkantsidis and P. Rodriguez Rodriguez, "Cooperative Security for Network Coding File Distribution," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006, pp. 1-13.
[159] M. Krohn, M. FreedMan, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *IEEE Symposium on Security and Privacy*, Berkeley, CA, 2004.
[160] L. Nutman and M. Langberg, "Adversarial models and resilient schemes for network coding," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, 2008, pp. 171-175.
[161] R. Villalpando, C. Vargas, and D. Munoz, "Network coding for detection and defense of sink holes in wireless reconfigurable networks," in *3rd International Conference on Systems and Networks Communications*, 2008.
[162] M. J. Siavoshani, C. Fragouli, and S. Diggavi, "On Locating Byzantine Attackers," in *Network Coding, Theory and Applications, 2008. NetCod 2008. Fourth Workshop on*, 2008, pp. 1-6.
[163] O. Kosut, T. Lang, and D. Tse, "Nonlinear network coding is necessary to combat general Byzantine attacks," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, 2009, pp. 593-599.
[164] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," presented at the Proceedings of the 6th annual international

conference on Mobile computing and networking, Boston, Massachusetts, United States, 2000.

[165] L. Anh and A. Markopoulou, "Locating Byzantine Attackers in Intra-Session Network Coding Using SpaceMac," in *Network Coding (NetCod), 2010 IEEE International Symposium on*, 2010, pp. 1-6.

[166] L. Buttyan, L. Czap, and I. Vajda, "Detection and Recovery from Pollution Attacks in Coding-Based Distributed Storage Schemes," *Dependable and Secure Computing, IEEE Transactions on,* vol. 8, pp. 824-838, 2011.

[167] Q. Wenbo, L. Jian, and R. Jian, "An Efficient Error-Detection and Error-Correction (EDEC) Scheme for Network Coding," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, 2011, pp. 1-5.

[168] L. Yongkun and J. C. S. Lui, "Identifying Pollution Attackers in Network-Coding Enabled Wireless Mesh Networks," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, 2011, pp. 1-6.

[169] M. J. Siavoshani, C. Fragouli, and S. N. Diggavi, "Subspace Properties of Network Coding and Their Applications," *Information Theory, IEEE Transactions on,* vol. 58, pp. 2599-2619, 2012.

[170] J. C. Corena, A. Basu, S. Kiyomoto, Y. Miyake, and T. Ohtsuki, "XOR network coding pollution prevention without homomorphic functions," in *IEEE 11th Consumer Communications and Networking Conference (CCNC)*, 2014, pp. 293-300.

[171] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Médard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 54, 2008.

[172] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," *ACM Trans. Inf. Syst. Secur.,* vol. 10, pp. 1-35, 2008.

[173] G. Sharma, S. Jaggi, and B. K. Dey, "Network tomography via network coding," in *Information Theory and Applications Workshop, 2008*, 2008, pp. 151-157.

[174] T. Ho, L. Ben, K. Ralf, M. Médard, Michelle, Effros., and K. David, "Byzantine Modification Detection in Multicast Networks With Random Network Coding," *IEEE Transactions on Information Theory,* vol. 54, pp. 2798-2803, 2008.

[175] T. Ho, L. Ben, K. Ralf, Médard, and Karger, "Byzantine Modification Detection in Multicast Networks With Random Network Coding," *IEEE Transactions on Information Theory,* vol. 54, pp. 2798-2803, 2008.

[176] R. Curtmola and C. Nita-Rotaru, "BSMR: Byzantine-Resilient Secure Multicast Routing in Multihop Wireless Networks," *IEEE Transactions on Mobile Computing,* vol. 8, pp. 445-459, 2009.

[177] K. Sukwon, T. Ho, M. Effros, and A. S. Avestimehr, "Network Error Correction With Unequal Link Capacities," *Information Theory, IEEE Transactions on,* vol. 57, pp. 1144-1164, 2011.

[178] A. Newell and C. Nita-Rotaru, "Split Null Keys: A null space based defense for pollution attacks in wireless network coding," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, 2012, pp. 479-487.

[179] X. Wu, Y. Xu, C. Yuen, and L. Xiang, "A Tag Encoding Scheme against Pollution Attack to Linear Network Coding," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, pp. 33 - 42, 2014.

[180] F. R. Kschischang, "Subspace codes and network coding," in *Information Theory (CWIT) 12th Canadian Workshop, Univ. of Toronto*, Toronto, ON, Canada, 2011.

[181] M. J. Siavoshani, S. Mohajer, C. Fragouli, and S. N. Diggavi, "On the Capacity of Noncoherent Network Coding," *Information Theory, IEEE Transactions on,* vol. 57, pp. 1046-1066, 2011.

[182] G. Solomon, "Generation of Maximum Distance Separable Codes," TDA Progress Report, communication research section1990.
[183] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Transactions on Networking,* vol. 16, pp. 497-510, 2008.
[184] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," presented at the Proceedings of the 6th annual international conference on Mobile computing and networking, Boston, Massachusetts, USA, 2000.
[185] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer, and R. A. Kemmerer, "An intrusion detection tool for AODV-based ad hoc wireless networks," in *Computer Security Applications Conference, 2004. 20th Annual*, 2004, pp. 16-27.
[186] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *Wireless Communications, IEEE,* vol. 11, pp. 48-60, 2004.
[187] W. Qiyan, V. Long, K. Nahrstedt, and H. Khurana, "MIS: Malicious Nodes Identification Scheme in Network-Coding-Based Peer-to-Peer Streaming," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1-5.
[188] L. Yongkun and J. C. S. Lui, "On detecting malicious behaviors in interactive networks: Algorithms and analysis," in *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, 2012, pp. 1-10.
[189] S. Acedanski, S. Deb, M. M´edard, and R. Koette, "How good is random linear coding based distributed network storage," in *1st Workshop on Network Coding, Theory, and Applications,Riva del Garda, Italy*, 2005.
[190] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *IEEE INFOCOM'05*, Miami, FL, 2005.
[191] A. G. Dimakis, P. B. Godfrey, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *IEEE INFOCOM'07*, Anchorage, Alaska, 2007.
[192] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the Usefulness of Watchdogs for Intrusion Detection in VANETs," in *Communications Workshops (ICC), 2010 IEEE International Conference on*, 2010, pp. 1-5.
[193] T. Shang, H. Pei, and J. Liu, "Secure network coding based on lattice signature," *Communications, China,* vol. 11, pp. 138-151, 2014.
[194] J. P. Vilela, L. Lima, and J. Barros, "Lightweight Security for Network Coding," in *IEEE International Conference on Communications (ICC 2008)*, 2008.
[195] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in wireless network coding," *ACM Trans. Inf. Syst. Secur.,* vol. 14, pp. 1-31, 2011.
[196] D. Charles, K. Jain, and K. Lauter, "Signatures for Network Coding," in *Conference on information Sciences and Systems (CISS '06)*, Princeton, NJ, 2006.
[197] C. Gkantsidis, J. Miller, and P. Rodriguez, "Anatomy of a P2P Content Distribution System with Network Coding," presented at the Proc. Int'l Workshop Peer-to-Peer Systems, Feb. 2006.
[198] Z. Kaiyong, C. Xiaowen, W. Mea, and J. Yixin, "Speeding Up Homomorpic Hashing Using GPUs," in *Communications, 2009. ICC '09. IEEE International Conference on*, 2009, pp. 1-5.
[199] K. Han, T. Ho, R. Koetter, M. Medard, and F. Zhao, "On network coding for security," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, 2007, pp. 1-6.
[200] M. Luby, "LT codes," in *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, 2002, pp. 271-280.