Towards Network Coding for Cyber-Physical Systems: Security Challenges and Applications

Pouya Ostovari and Jie Wu
Department of Computer & Information Sciences, Temple University
Philadelphia, PA 19122
Email: {ostovari, jiewu}@temple.edu

***Abstract*—This survey summarizes the research on the applications and security challenges of network coding in wireless networks and Cyber-Physical Systems. Network coding technique generalizes the store-and-forward routing by mixing the received packets at the intermediate nodes to a single packet before forwarding them. Network coding has received a lot of attention from the community, and researchers has widely studied its application in increasing the throughput and transmissions reliability of networks and cyber-physical systems. Applying network coding can be a challenge to the security of the networks. However, network coding can also provide a natural way to conceal the transmitted data from eavesdroppers. As a result of mixing the received packets at the intermediate nodes, the transmission protocols in network coding are more vulnerable against some attacks, such as pollution and Byzantine attacks. A single polluted packet can easily pollute many packets. Also, because of encoding at the intermediate nodes, the security mechanisms that are proposed for the traditional transmission protocols might not be appropriate for the protocols with network coding. On the other hand, since the transmitted packets are encoded, an eavesdropper cannot get meaningful data by overhearing a single encoded packet, which makes the transmissions more robust against eavesdropping attack. In this survey, in addition to discussing the security challenges in network coding protocols, we explain the applications of network coding in providing security. Our main focus in this paper is on the applications of network coding in providing security.**

***Index Terms*—Network coding, security, cyber-physical systems, wired networks, wireless networks, pollution attack, eavesdropper, Byzantine attack.**

I. Introduction

The traditional data forwarding methods in wired and wireless networks use store-and-forward routing, in which the intermediate nodes store the received packets from their downstream nodes and forward them to their upstream nodes. However, the traditional store-and-forward routing cannot use the full capacity of the networks. For this reason, network coding was proposed for the first time in [1], which is a generalization of the classic store-and-forward routing to code-and-forward paradigm. The authors in [1] proposed the concept of network coding to solve the bottleneck problem in wired networks, and achieve the capacity of the multicast problem. They proposed the max-flow min-cut theorem and showed that the multicast capacity can be archived using network coding.

Consider the example in Figure 1. There is a source node s, which wants to transmit 2 packets p_1 and p_2 to destination nodes d_1 and d_2. Let us assume that the capacity of each link is equal to one packet. A possible transmission scheme is shown in Figure 1(a). It can be seen in the figure that there is a bottleneck between nodes r_3 and r_4. If node r_3 transmits packet p_1, node d_1 cannot receive packet p_2. On the other hand, if node r_3 transmits packet p_2, node d_2 only receives packet p_2. In Figure 1(b), node r_3 applies network coding, and transmits a combination of 2 packets received. In this case, nodes d_1 can subtract p_1 from the coded packet p_3 to retrieve p_2. Moreover, node d_2 can retrieve p_1 by subtracting packet p_2 from p_3. As a result, network coding helps us to deliver both of the packets to the destination nodes, and increase the multicast capacity of the network from 1 to 2.

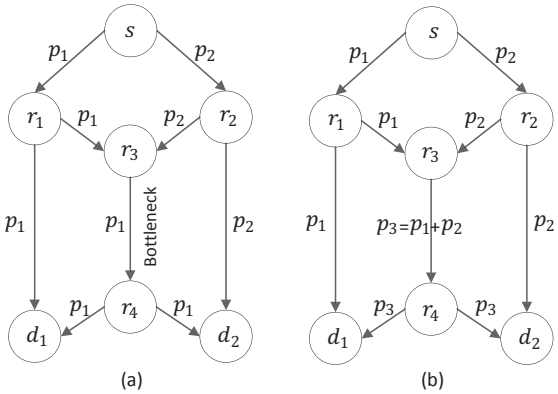

Fig. 1. Network coding in wired networks. Butterfly network.

Later, the authors in [2] proposed linear network coding, in which the coded packets are linear combinations of the original packets. They proved that linear network coding suffices to achieve the optimum multicast solution in wired networks. The idea of random linear network coding was proposed in [3]. The authors show that selecting the coefficients of the coded packets randomly over a large finite field, will likely result in the coded packets being linearly indecent. As a result, random linear network coding is likely sufficient to achieve the capacity of the multicast problem.

Network coding was proposed for the wired networks. However, the researchers found that the wireless nature of the medium in wireless networks makes network coding more attractive in wireless networks. The reliability of the wireless

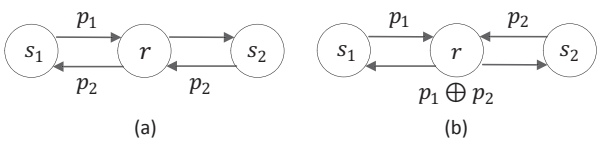

Fig. 2. Network coding in wireless networks.

links are less than the wired links. As a result, providing reliable transmissions in wireless networks is more important than the wired networks. On the other hand, because of the wireless nature of the medium in wireless networks, multiple nodes can overhear each data transmission. This overhearing provides opportunities for network coding to increase the throughput of the wireless networks.

Consider the example in Figure 2(a). There are two source nodes s_1 and s_2, which want to transmit a packet to each other. Since they are not in the range of each other, they first need to forward their packets to the relay node r. Then, the relay node forwards the received packets to nodes s_1 and s_2. As a result, 4 transmissions are required to exchange packets p_1 and p_2. Now consider Figure 2(b), in which network coding is applied at the intermediate node. Instead of 2 transmissions for packets p_1 and p_2, the relay node r transmits a single packet $p_1 \oplus p_2$, where $\oplus$ is the XOR operation. In this case, node s_1 has packet p_1 in its buffer. Node s_1 can perform $p_1 \oplus (p_1 \oplus p_2) = p_2$ to retrieve packet p_2 from the received coded packets. Also, Node s_2 can perform $p_2 \oplus (p_1 \oplus p_2) = p_1$ to calculate packet p_2. In this example, network coding helps to reduce the number of transmissions from 4 to 3, which means 25% improvement in the throughput.

Network coding has been widely studied by the community, and it is used for a variety of applications. The applications of network coding are include, but not limited to, throughput and capacity enhancement in wired and wireless networks, providing reliable transmissions, robustness enhancement against failures in data storages, network tomography inference, and secure transmissions.

From the perspective of network security, network coding can be both a challenge and an opportunity in networks and Cyber-Physical Systems. Network coding makes the network more vulnerable against some types of attacks, such as pollution and Byzantine attacks. This is due to the mixture of packets performed at the intermediate nodes. As a result of the packet mixtures, a polluted packet can easily pollute many coded packets. Moreover, when network coding is applied in a network, the proposed anonymous communication methods, such as onion routing [4], cannot be used. As a result, new methods are needed to protect the network communication protocols against the security threats.

On the other hand, network coding can be used as a tool to defend against eavesdropping attack and to provide secure transmissions. When random linear coding is applied at the source nodes, the transmitted packets are encoded and mixed together. As a result, a single coded packet does not provide any information to the eavesdropper. Any node, including an eavesdropper, needs a sufficient number of coded packets to be able to decode the coded packets and retrieve the original packets. Therefore, network coding is a natural way to conceal data and to provide security. In addition, network coding can be used in a secret key sharing method, as we will discus later in this paper.

In this paper, we will discuss the security threats in the network coding protocols, in addition to the applications of network coding in securing the networks. Our main focus in this work is the application of network coding, specifically in providing security. The remainder of the paper is organized as follows. In Section II, we provide a background on network coding and the applications of network coding. We discuss the security challenges in network coding protocols and classify them in Section III. The proposed defense schemes for making network coding secure against attacks are discussed in Section IV. We discuss the methods that use network coding to provide security in Section V. Section VI concludes the paper.

II. Background on Network Coding and its Applications

In the following subsections, we first provide a background on network coding. We then discuss different applications of network coding in wired networks, wireless networks, data storages, and peer-to-peer systems. Finally, we classify network coding from different aspects.

A. Background and Preliminaries

Network coding generalizes traditional store-and-forward routing. The idea of network coding [5]–[8] is proposed in [1] for the first time. The authors show that the multicast capacity of wired networks can be achieved using network coding. The main contribution of the work in [1] is the min-cut max-flow theorem. The theorem says that the maximum multicast capacity (max-flow) of a network is equal to the min-cut from the source node to the set of destination nodes in the multicast session. The authors in [2] prove that linear network coding suffices to achieve the capacity of the multicast problem, which is equal to the max-flow from the source node to each of the receiving nodes.

The idea of random linear network coding is proposed in [3], which makes the transmission schemes simple. The authors proves that if the coefficients of the linearly coded packets are selected randomly and in a distributed fashion at each intermediate node, there is a high probability that the generated coded packets will be linearly independent. As a result, random linear network coding suffices to achieve a flow rate very close to the capacity of the multicast problem. The authors in [9] take an algebraic look at the issue of network capacity, and derive a useful algebraic model of linear network coding.

In linear network coding, the coded packets are linear combinations of the original packets over a finite (Galois) field. A main advantage of linear network coding is that, any linear

combination of the coded packets is also a linear coded packet. In random linear network coding, the coefficients of the coded packets is randomly selected over the finite field. In this type of coding, each coded packet has a form of $\sum_{j=1}^{m} \alpha_j \times p_j$. Here, p_j is an original or a linearly coded packet. Also, the coefficients of the coded packets is shown as α_j.

Random linear network coding has two main advantages. First, since the coefficients are selected randomly, random linear network coding is appropriate for distributed systems, such as large communication networks. Second, similar to fountain codes [10]–[13], the source node can generate an unlimited number of coded packets using random linear network coding. The source node keeps transmitting the coded packets until the destination nodes receive m linearly independent coded packets. In order to decode the received coded packets, the destination nodes need to use a Gaussian elimination algorithm to solve a system of linear equations. In this way, reliable transmission can be provided without the need to feedback messages about the received packets. The destination nodes only need to transmit a single acknowledgment message when they are able to decode the coded packets.

B. Network Coding Applications

Network coding was first proposed for wired networks to solve the bottleneck problem and maximize the throughput of the multicast problem. However, researchers have found a wide range of applications for network coding. It has been shown that network coding is even more attractive and useful in wireless networks. This is due to two important characteristics of wireless networks: the unreliability of the wireless links, and the wireless nature of the medium. The authors in [14] classify the applications of network coding into throughput/capacity enhancement, robustness enhancement, network tomography, and security. In the following subsections, we discuss these applications in more detail.

1) Throughput/Capacity Enhancement: As mentioned before, network coding was first proposed for wired networks to achieve the capacity for multicast application. By solving the bottleneck problem, network coding can increase the throughput in wired networks. Network coding is more attractive in wireless networks, which is due to the overhearing possibility in these networks. COPE is the first practical forwarding architecture using network coding. COPE uses overhearing among the nodes as an opportunity to enhance the throughput of wireless networks. The idea in COPE can be described with the example in Figure 3. Source nodes s_1 and s_2 want to transmit packets p_1 and p_2 to destinations d_1 and d_2, respectively. If we do not use network coding, 4 transmissions are required to transmit the packets to the destination nodes. Because of the broadcast nature of the network, node d_1 can overhear packet p_2, which is broadcasted by node s_2. Also, node d_2 can overhear the transmission by node d_1. As a result, if the relay node r transmits a combination of packets $p_3 = p_1 \oplus p_2$, each destination node can XOR the overheard packet from the source nodes with p_3, and retrieve packets

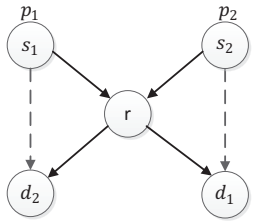

Fig. 3. Network coding in wireless networks, COPE method.

p_1 and p_2. In this way, one transmission is saved at the relay node and the throughput of the network is increased.

The authors in [15]–[17] study the problem of one-hop reliable transmission. The authors use feedback messages to provide reliability. The receiver nodes transmit feedback messages to report the received packets. The source node uses XOR coding in the retransmissions to reduce the amount of required transmission. In this way, each coded packet can deliver multiple lost packets to the different receivers. As a result, the throughput of the system is increased.

2) Robustness Enhancement: Any transmission on a wired or wireless link is subject to failure. Using feedback messages is the most common way to report the received packets to the senders and ask to retransmit the lost packets. Automatic repeat request [18] is frequently used to provide reliable transmissions. The main drawback of the ARQ method is the overhead of feedback messages. This overhead becomes a major challenge in multicast problems. Because of this overhead, most of the multicast applications do not use the ARQ method. In order to reduce the overhead of feedback messages, hybrid [19], [20] methods are proposed, which combine forward error correction [21]–[25] with the ARQ method.

Network coding can serve as an erasure correcting code, which is an effective way to provide reliable transmissions. As mentioned before, the source node can keep transmitting random linear network coded packets, until the destination nodes receive a sufficient number of coded packets. After a successful decoding, the destination nodes transmit an acknowledgment message to stop the source node from transmitting more packets. Using this scheme, there is no need for feedback messages. The reason is that, each packet contributes the same amount of information to the destination nodes, and the source node does not need to know which exact packets are received or lost.

Network coding can be also used to provide fault tolerance in storage systems. For this purpose, network coding can be applied on the original files, and coded packets can be stored on several distributed data storages. In order to provide fault tolerance, the total amount of stored coded packets on the set of the data storages should be more than the number of original packets. In the case that some of the data storages fail, unless the set of coded packets that are stored on the other data storages is more than or equal to the number of

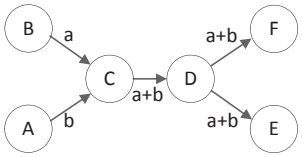

Fig. 4. Link loss rate inference [14], [32].

original packets, the original data can be recovered.

3) Protocol Simplification: Applying network coding on the source data can simplify many protocols. For example, a major challenge in peer-to-peer (P2P) networks [26]–[28] is content distribution and tracking the location of the stored data on different peers. In order to retrieve the original content, different parts of the file need to be collected from different peers, which need a reference table to show the parts that are stored on each peer. Network coding simplifies tracking the stored data [29], since each coded packet contributes the same amount of information to the users. As a result, we just need to know how much content is stored on each peer node. Moreover, in content distribution problems, many optimization problems cannot be solved in polynomial time, such as the problem in [30], [31]. When network coding is applied on the content, some of these optimizations become similar to a flow optimization problem, which can be modeled as a linear programming optimization, and be solved in polynomial time.

4) Network Tomography: Network coding also can be used to infer the characteristics of a network. Network coding is not necessary for network tomography, but it can improve the accuracy and reduce the complexity of network tomography [14]. Network coding can be used in link loss rate inference and topology inference. Consider the network in Figure 4, which is from [14], [32]. Nodes A and B transmit packets a and b simultaneously. Node C receives the packets, and transmits $a+b$ to node D. Then, node D relays the received packet to nodes E and F. If node D receives packet a, it means that packet b is lost on link BC. If node E receives packet $a+b$ and F receives nothing, it implies that the loss happened on link CD. By repeating the transmissions and gathering data from the nodes, we can find the loss rate of each link.

5) Security: Network coding can be used to provide secure transmissions against eavesdropper attack. Assume that a source node has n packets to transmit, and the eavesdropper can collect k transmitted packets. If we do not use network coding or encryption methods, the eavesdropper will receive useful information from the transmitted data. Now assume that the source node performs random linear network coding on the source data, and transmits coded packets. In the case that k is less than n, the eavesdropper will not be able to decode the coded packets and retrieve the original data. As a result, network coding can help us protect the transmitted data. In Section V, we discuss the applications of network coding in providing security in more detail.

TABLE I
CLASSIFICATION OF NETWORK CODING APPLICATIONS.

Network coding application	Types
Throughput/Capacity enhancement	Distributed storage Content distribution Layered multicast Wireless networks
Robustness enhancement	Erasure correcting code Fault tolerance
Protocol simplification	Content distribution P2P system
Network tomography	Link loss rate inference Topology inference
Security	Defense against eavesdropping Secret key sharing

C. Network Coding Classification

Network coding methods can be classified from different aspects. From one aspect, network coding can be classified into intra-session and inter-session network coding. Intra-session network coding is among the packets of the same flow. The source node mixes the packets to be transmitted to a single or multiple destination nodes, and depending on the protocol design, the intermediate nodes might recode the packets. This type of network coding is useful in providing reliable transmissions and in content caching on storages. This type of network coding is also useful in secure data transmission. The example in Figure 1 is intra-session network coding. In contrast, inter-session network coding is performed among the packets of different flows. The example is Figures 2 and 3 are inter-session network coding, which can reduce the number of transmissions and increase the network throughput. From another point of view, network coding can be classified as stateless and state-aware network coding.

1) Stateless Network Coding Protocols: The stateless network coding protocols do not rely on the network state information, such as topology information (neighbors of each node) and the packets in the buffer of the nodes, to decide when and how to mix the packets at each intermediate nodes [33], [34]. These types of coding can also be referred to as global network coding, since the decoding is typically performed only at the destination nodes.

2) State-Aware Network Coding Protocols: In state-aware network coding protocols, each node needs partial or full information about the state of the network, such as the packets in the buffer of its neighbors and the network topology. This information is used to construct a network code that is decodable by the neighboring nodes. Typically, each intermediate node decodes the received decoded packets before computing and transmitting a new coded packet. Then, depending on the state of its neighbors and their buffer, the node transmits a coded packet that is decodable by its neighbors. Since the coding and decoding is performed hop-by-hop, this type of coding can be called local network coding.

III. SECURITY CHALLENGES

Security challenges are not new and specific to network coding. However, the packet mixture makes the network

coding protocols more vulnerable against some of the security challenges. Moreover, some of the protocols that use network coding need exchanging control messages between the nodes, which make the proposed security methods for the traditional networks useless for network coding. In this section, we discuss some of the major security threats in network coding protocols. We also classify the threats and explain the difference between the threats in the case of traditional networks and network coding protocols.

A. Byzantine Attack

In Byzantine attack, which is also called Byzantine fabrication attack, wrong control messages are created by the malicious nodes. These wrong messages can be the header of the packets, or individual packets. The malicious nodes might change the routing information in the header of the packets, send wrong information about the packets in the buffer of the nodes, and make false acknowledgments.

Byzantine attack is a threat to both of the stateless and state-aware network coding protocols. In state-aware protocols, the coding at the intermediate nodes is performed based on the state of the neighbors, such as the packets in the neighbors' buffer and the connections (overhearing) among the neighbors of the coding node. The attackers can disrupt the normal operation of the network by sending wrong information about the state of the nodes. Also, in the case of stateless protocols, the attackers can change the header of the coded packets, which contain the coding vectors and routing information.

B. Pollution Attack

Pollution attack is sometimes referred to as Byzantine modification and pollution attack. In pollution attack, the malicious nodes modify and change the packets that should be transmitted, or inject fake packets to the network. This attack also exits in the store-and-forward methods. However, because of the packet mixture which happens in the intermediate nodes, pollution attack is more serious when network coding is applied. If the polluted packets are not discarded, they can be mixed with the clean packets and pollute the whole network.

C. Traffic Analysis

Another type of attack in networks is traffic analysis, in which the attacker nodes monitor the transmissions in the network in order to find the source and destination of the packets and the network topology. In applications such as military applications, it is important to hide the source and the destination of the packets. Otherwise, the enemy might attack these nodes. For traditional networks, anonymous routing methods, such as onion routing [4], [35], [36], can be used. However, onion routing is not applicable in the networks that are enabled with network coding operations. The main reason is that, the coding operations that happens at the intermediate nodes have conflict with the encryption/decryption that is performed in onion routing [37].

D. Eavesdropping Attack

An eavesdropper can wiretap one or several wire links, or overhear the wireless transmissions in order to retrieve sensitive data, such as passwords or confidential messages. The work in [33] classifies the eavesdroppers to *nice but curious* and *wiretapping* nodes. The nice but curious nodes, which can be refereed to as non-malicious nodes, are well-behaved in the sense of communication protocol, but they may try to extract information from the data flows that pass through them [33]. As we will see later, random linear network coding can provide protection against nice but curious nodes. Since the packets in random linear network coding are encoded, a curious nodes cannot get meaningful information without having access to a sufficient number of coded packets to be able to use Gaussian elimination and decode the coded packets. In contrast with the nice but curious nodes, the wiretapping nodes (malicious nodes) have access to a subset of communication links. These nodes are more capable than the nice but curious nodes, as they have access to more packets. In the extreme case, they have access to all of the transmitted packets. As a result, concealing the original data from these nodes is harder and more critical.

A classification of the eavesdropper nodes based of their behavior is presented in Figure 5(a). Typically, a nice but curious nodes cannot have access to all of the packets, since it does not try to capture the packets. However, a wiretapping node might have access to a set of packets, or all of the transmitted packets in the network. From another perspective, an eavesdropper can be an external or an internal node, which is shown in Figure 5(b).

In the case of stateless (global) network coding, eavesdropper attack is less critical. The reason is that, typically, random linear network coding is applied in global network coding. As a result, the eavesdropper cannot decode the coded packets and retrieve the original packets until it has access to a sufficient number of coded packets. However, in the case of state-aware (local) network coding, the coding is local, and each intermediate node performs decoding before recoding the packets and transmitting them. As a result, network coding cannot conceal the data, and evesdropping attack becomes more serious in the case of local network coding [34].

E. Classification of the Attacks

A classification of security attacks in network coding is shown in Table II. In the following subsection, we discuss classification of the threats.

1) Passive vs. Active: From one perspective, we can classify the security threats into passive and active attacks. Passive attacks do not interrupt the normal operation of the system, such as data transmissions and data storage. However, the malicious or the curious nodes overhear and store the transmitted data. The eavesdropping attack is a passive attack, in which the eavesdropper tries to get information from the transmitted data by the other nodes in the network. Another example of a passive attack is the traffic analysis attack. In passive attacks, confidential data might be revealed to the nodes that are not authorized to receive it. The nodes that participate

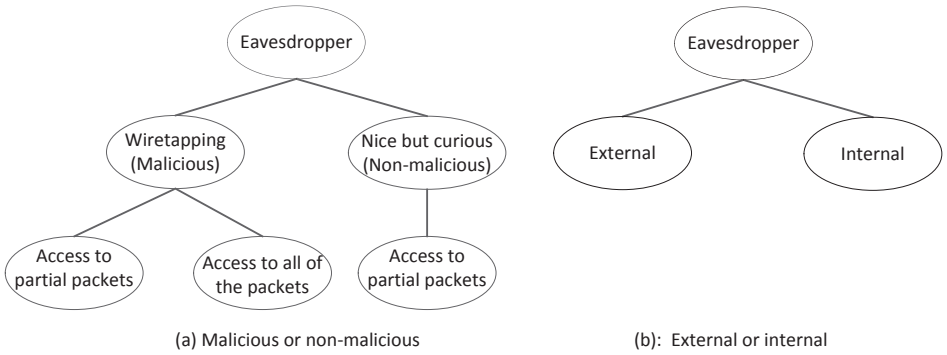

Fig. 5. Classification of eavesdroppers.

TABLE II
CLASSIFICATION OF THE ATTACKS IN NETWORK CODING.

Attack	Active	Passive	External	Internal	Effect of NC
Byzantine	✓		✓	✓	Challenge
Pollution	✓		✓	✓	Challenge
Traffic analysis	✓		✓	✓	Challenge
Eavesdropping		✓	✓	✓	Opportunity

in the passive attacks might be external nodes, which do not belong to the network, or internal nodes. They can also be malicious or just curious nodes.

In contrast with the passive attacks, the active attacks disrupt the normal operation of the network, and try to fail the system. The active attacks can be done by external or internal malicious nodes. The malicious nodes can modify the transmitted data, transmit fake data, or send incorrect control messages. Pollution and Byzantinism attacks are active attacks.

2) External vs. Internal: The attacker nodes can be external nodes, which means they are not a part of the communication network. The external nodes might overhear the transmitted packets or transmit fake messages and packets to the other nodes. In contrast, the attacker nodes might be internal nodes, which are a part of the network. In the case of eavesdropping attack, the internal nodes can be nice but curious nodes.

3) Effect of Network Coding: Applying network coding might be a security challenge. In other words, network coding might make the network more vulnerable to the security attacks. On the other hand, network coding can be used as a tool to provide security. In the case of Byzantine fabrication, pollution, and traffic analysis attacks, network coding is a challenge to the network security. However, network coding can help us to conceal information from unauthorized nodes.

IV. SECURE NETWORK CODING

In this section, we discuss the methods that are proposed to make the network coding protocols secure. In more detail, we study the defense mechanisms against Byzantine, pollution, and traffic analysis attacks.

A. Defense Against Byzantine and Pollution Attack

The methods that are proposed to defend against Byzantine and pollution attacks can be classified as end-to-end error correction and misbehavior detection methods [33]. The advantage of end-to-end error correction methods is that, the error detection and correction only happens at the destination nodes. As are result, the intermediate nodes do not need to change their normal operations once an attack happens in the middle of the transmission [33]. The advantage of misbehavior detection methods is that, the attacks can be detected in the early stages, and the resources of the network will not be wasted for transmitting polluted and corrupted packets.

Cryptographic schemes can be used to find the misbehavior of the nodes. For this purpose, different types of digital signatures and encryptions can be used. The work in [38] proposes to using homomorphic hash functions to detect polluted packets in file distribution systems, such as P2P systems. In general, the computation complexity of applying hash functions is high. The authors in [39] propose to using secure random checksum to detect the polluted packets, which has less computation complexity compared to homomorphic hash functions. In [40], [41], a homomorphic signature-based authentication method is proposed, but its computation complexity is high.

The author in [42], [43] proposes a signature scheme, which is designed for random linear network coding. They consider the file as a vector. The work uses the fact that all of the valid random linear network coded packets that are transmitted in the network are vectors that should belong to the subspace spanned by the set of vectors from the original file (vector). Based on this fact, a lightweight signature scheme is proposed, which can easily verify the clean packets, and it is hard for a node to generate a fake vector that passes the test.

The authors in [44] derived mathematical relations in linear network coding, and proposed a key pre-distribution based tag encoding scheme, which can protect the network against pollution attach by tagging pollution attacks. They also quantitatively analysis their proposed method and compared it with other schemes. In [45], a digital signature scheme has been proposed that can detect the polluted packets. Moreover, the authors propose a scheme that can detect a malicious node.

The authors in [46] propose a rateless and pollution attack resilient network coding method for multicasting application.

Many peer-to-peer systems use network coding to increase the throughput of their system and simplify their content distribution protocol. However, this makes a challenge as some of the node might propagate polluted packets in the system. The authors in [47] consider a peer-to-peer streaming system, and show pollution countermeasures that make a peer-ro-peer system that uses network coding resilient to pollution attacks. They modeled the diffusion of the polluted packets in the network. Based not their analysis, the packets that are received earlier by a node are less likely to be polluted. Moreover, the chance that nodes can recover a clean generation increases for short generations. Following these observations, the authors propose a coding scheme where nodes draw packets to be coded according to their age in the input queue. Also, they use a decoding scheme that is able to detect the reception of polluted packets early.

B. Defense Against Traffic Analysis

Onion routing [4], [35], [36] is a well-known method to defend against traffic analysis in traditional networks. However, onion routing cannot be applied directly in the protocols that use network coding. The authors in [37] propose a method called ANOC (Anonymous network coding), which is a modification of onion routing that works with network coding. In the following, we use an example from [37] to briefly describe the idea in onion routing and the reason why onion routing fails when network coding is enabled.

Consider the network in Figure 3. Assume that two nodes U_1 and U_2, which are connected to routers s_1 and d_1, want to establish a session. In onion routing, U_1 sends a request for a connection to router s_1. Then, s_1 find a path to router d_1. Node s_1 selects two random session keys sk_{r_1} and sk_{d_1} for nodes r and d_1, respectively. Also, s_1 creates a layered data structure, which is called onion. In this case, the onion is $\{\{sk_{d_1}, U_2\}_{uk_{d_1}}, sk_{r_1}, d_1\}_{uk_{r_1}}$. Here, $\{.\}_k$ represents the encryption using public key k. Moreover, uk_{r_1} and uk_{d_1} are the public keys of nodes r and d_1, respectively. Node s_1 sends the onion to node r. Node r decrypts the onion, and retrieves session key sk_{r_1}, next-hop d_1, and the onion $\{sk_{d_1}, U_2\}_{uk_{d_1}}$, which is embedded in the received onion. Node d_1 receives the onion and decrypts it. Node d_1 will find that U_2 is the last node in the route, and forwards the connection request to it, and the session is created. After this step, data transmission is performed with symmetric-key encryptions.

Assuming that nodes U_3 and U_4 are connected to nodes s_2 and d_2, the same process can be used to establish a session between nodes U_3 and U_4. However, instead of sk_{r_1}, sk_{r_2} is used. The problem arises when coding is applied at node r. Nodes d_1 and d_2 can overhear the messages sent by nodes s_2 and s_1, respectively. However, since messages are encrypted with session keys sk_{r_1} and sk_{r_2}, d_1 and d_2 cannot decode them.

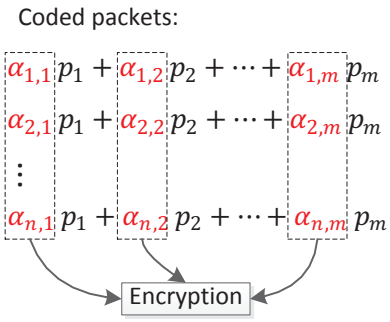

Fig. 6. Encrypting coefficients of coded packets instead of encrypting the original packets.

V. Applications of Network Coding in Providing Security

In the previous section, we discussed the defense schemes against Byzantine, pollution, and traffic analysis attacks, which make network coding secure. In this section, we discuss the applications of network coding in providing security. These applications serve as a defense against eavesdropping attack and secret key exchange.

A. Eavesdropping Attack

Linear network coding is a natural way to conceal data from eavesdroppers. As in linear network coding, the packets are mixed and encoded together; an eavesdropper cannot decode the coded packets until it has access to a sufficient number of coded packets. However, it does not guarantee that the eavesdropper cannot decode and get partial information.

1) Secure Data Transmission: The authors in [48] propose a low-complexity cryptographic security mechanism that exploits random linear network coding. The idea of the paper is to encrypt the coefficients of the network coded packets instead of the original data, as shown in Figure 6. The size of the coefficients of the coded packets is much less than that of the original data. As a result, the amount of the data that should be encrypted reduces dramatically, which reduces the encryption and time complexity. The source node performs random linear network coding on the original data, and encrypts the coefficients of the coded packets using a secret key, which is known only by the destination node. Recall that in random linear network coding, the intermediate nodes need to recode the received coded packets. However, the coefficients of the packets that are transmitted by the source node are encrypted. An intermediate node cannot recode the packets without knowing the coefficient of the coded packets, since in the recoding process, the coefficients should be modified. In order to solve this problem, the authors use two sets of coefficients. The first set of coefficients, which are called locked coefficients, are introduced by the source node for the encryption purpose. The second set of coefficients are introduced by the intermediate nodes and they are for the purpose of data transmission.

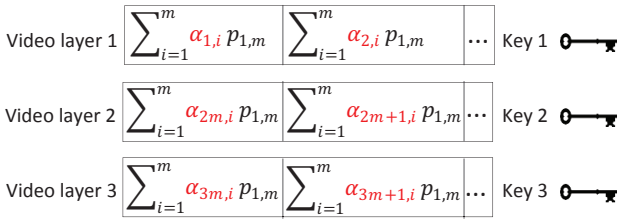

Fig. 7. Encrypting coefficients of coded packets in the case of multi-layer videos.

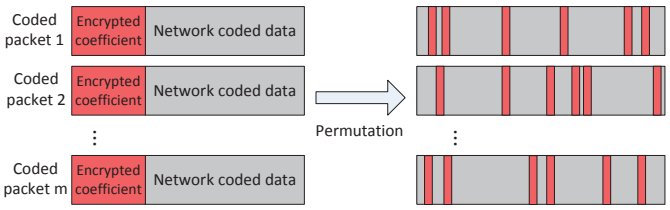

Fig. 8. P-coding scheme [58], [59]. Combining network coding and permutation to provide security.

The authors in [49] extend the idea of coding coefficients of coded packets to the case of multi-resolution videos [50]–[54]. In a multi-resolution video, which is also called multi-layer video, the video is divided to a base layer and a set of enhancement layers. The base layer is necessary to watch the video with a low quality. However, the enhancement layers can boost the quality of the received video. Multi-layer videos are useful for a variety of applications. For example, in video multicasting, the channel quality of the receivers might be different. Using a multi-layer video, the receivers can watch the video with a quality corresponding to their channel quality. Also, in the case of video broadcasting, the users can subscribe and pay for the service with their desired quality. The idea in [49] is to encrypt the coefficients of each video layer with a different key. The users only know the keys that are correspondent to the layers that they subscribed to. As a result, they cannot decrypt the coefficients of the other layers, which are required to decode the coded packets. The encryption scheme is shown in Figure 7.

In [55], the authors perform a trade-off between transmission cost and security. In their model, each link is associated with a cost and the probability that the eavesdropper can wiretap the link. A source node has a set of packets to transmit to a set of destination nodes. The data is linearly coded at the source node. In order to conceal the source data from the intermediate nodes, they are transmitted through disjoint paths. The objective of the work is to find the amount of the data that should be transmitted through each link such that a function of the total cost and network vulnerability is minimized. The authors formulate the problem as an optimization problem, and propose two heuristics to solve it.

In [56], the authors study the threat posed by intermediate nodes in a wired networks. They assume that the nodes are nice but curious and comply with the transmission protocol, but they might also be eavesdroppers. They investigate the security potential provided by network coding, and take an algebraic look at security. In order to understand the interplay between network topology and security against eavesdropping attack, the authors analyze the achievable level of algebraic security in directed acyclic graphs. In this graph, each eavesdropper has access to a limited number of coded packets.

In [57], the authors propose a new theoretic model for security, called weakly secure. A system is called weakly secure, if no meaningful information about the source messages can be retrieved from the intercepted packets by eavesdropper. This is in contrast with Shannon security [57], in which the adversary should not get any information about the source messages based of the intercepted packets. The authors proof that there exists a secure network code that can achieve the capacity of multicast. They also calculate the probability that an eavesdropper can get meaningful information about the source packets in the case that random linear network coding is applied in the network.

A lightweight encryption scheme, called P-coding, is proposed in [58], [59], which is based on network coding. The main idea of the paper is to combine random linear network coding with permutation of the symbols of the packets. In symbol permutation, the symbols that form a packet are permuted, which means their locations in the packets are changed. Only the source and the destination nodes know the permutation function. In P-coding, first network coding is applied on the original packets. Then, the permutation is performed on the network coded packets. Combining permutation and network coding together results in enhancing the security of P-coding. The coding scheme in P-coding is shown in Figure 8. Moreover, since the complexity of the permutation function and network coding are relatively low, P-coding is a lightweight encryption method. The paper also presents theoretical analyses on the security of P-coding.

2) Secure Data Storage: The problem of providing trusted storage over untrusted network is addressed in [60]. It is assumed that a user has a large file to store on a set of untrusted data storages. An eavesdropper has access to a set of data storages, but not to all of them. In order to conceal the data from the eavesdropper, the authors propose to apply random linear network coding on the data, and storing the coded packets on the data storages.

The authors extend their work in [61] to consider reliability in addition to security. An eavesdropper has access to a limited number of data storages. As a result, if the eavesdropper cannot download enough coded packets, it cannot decode and retrieve the original data. On the other hand, in order to make the distributed storage robust against storage failures, the authors propose to add equal redundancy on each data storage. In the case that a data storage fails, another data storage will be added to the system, and the linear combination of the stored packets on the other storages will be stored on a new storage. The authors also analyzed the security of their proposed secure

data storage method.

In [62], a secure and fault tolerant data storage system has been proposed. In this system, each data storage is subject to eavesdropping and failure with known probabilities. In order to make the system robust against failure, redundant data needs to be stored on the data storages. More redundancy makes the system more robust against storage failures. However, more redundancy results in more vulnerability against eavesdropping, since there is a higher chance that an eavesdropper has access to a sufficient number of coded packets. For this reason, the authors propose an optimization technique to perform a trade-off between fault tolerance and security of the system.

B. Secret-Key Exchange

A fundamental requirement of the most of cryptographic security methods is distributing secret keys in a secure manner. The authors in [63] show that network coding is an effective and efficient tool in providing secret key sharing. They propose a scheme that is based on XOR coding and provides a low-complexity secret key sharing among sensor nodes. The main idea of the method is very similar to the example in Figure 2. There are two nodes A and B, which want to share their secret keys with each other. Also, node C helps them to share their secret keys. The secret keys of nodes A and B should not be revealed to S. The proposed method contains two phases: prior and after sensor node deployment. The two phases are as follows [33], [63]:

(a) Prior to sensor node deployment:

- A large pool P of keys K_i and their identifiers are generated.
- A different subset of the keys in P and their identifiers are randomly selected and stored on each sensor node. Each of these random keys are served for stabilizing a connection in phase 2.
- The list of the identifiers of the keys in P and an encrypted version of their correspondent key $K_i \oplus R$ is stored on helper S. Here, R is a random key.

(b) After sensor node deployment:

- The helper node S transmitted a hello message for neighborhood discovery.
- Each sensor node that is in the transmission range of node S replies with a key identifier.
- Node S uses the key identifier to find the encrypted key in the list of the keys. Node S combines encrypted keys of each two sensor nodes, A and B, to conceal R. In this way, they result will be XOR of the keys of nodes A and B.
- Node S transmits the result.
- Nodes A and B can retrieve the keys of each other by combining (XOR operation) the received message from S with their own keys.

The steps of phase 2 (after sensor node deployment) are illustrated in Figure 9.

In [64], [65], the problem of efficient dissemination of the shares of a secret to the nodes of a network is addressed.

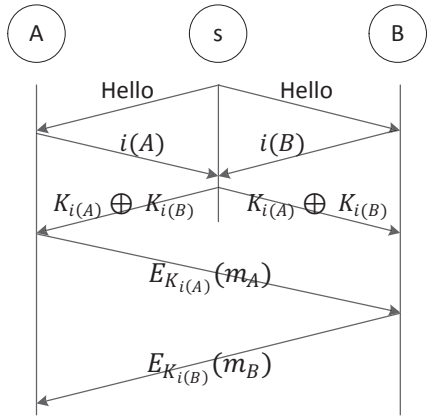

Fig. 9. Secret key distribution using network coding [33], [63]. Here, $i(A)$ and $i(B)$ are the identifiers of nodes A and B, respectively. Also, m_A and m_B are the messages that should be transferred securely from nodes A and B to each other. The encrypted messages with keys $K_{i(A)}$ and $K_{i(B)}$ are represented as $E_{K_{i(A)}}(.)$ and $E_{K_{i(B)}}(.)$, respectively.

The objective of the work is to use Shamir's secret sharing method [66], and transmit the secret shares from a source node to a set of participant nodes in a multuihop fashion. Shamir's (n, k) secret sharing method considers $(n+1)$ entities, which consists of a dealer and n participants. The dealer wants to share a secret s among the n participants such that the shares of any k participants are sufficient to recover the secret s, and the aggregated data from any less than k nodes does not reveal any information about s.

In the case that there is a wire connection between each participant and the dealer, the dealer can easily pass the secret of each participant to it. However, in the case of multihop networks, each secret needs to be protected along the path from the dealer to the corresponding participant. Otherwise, the other participants will know more than their corresponding secret keys. In order to address this problem, the authors in [64], [65] propose a systematic linear network coding, which encodes the secret keys of each participant in a special way. The main idea of the paper is to transmit the secret of each participant through multiple disjoint paths, such that each node does not receive enough encoded packets to retrieve the other nodes' secret keys.

VI. Conclusion

Primarily, network coding has been applied in wired networks to solve the bottleneck problem and achieve the multicast capacity of the wired networks. However, it has been shown that network coding has a wide range of applications in wired, wireless, and storage systems. From the perspective of network security, network coding can be a challenge and an opportunity of the networks and Cyber-Physical Systems at the same time. Some of the existing attacks in traditional networks, such as pollution attack and Byzantine attack become more serious when network coding is applied in the network. Moreover, the well-known onion routing which has been pro-

posed to defend against traffic analysis in traditional networks cannot directly be applied for the protocols that use network coding. On the other hand, mixing the packets in network coding provides a natural way to conceal the source data from an eavesdropper. Any nodes, including eavesdroppers, need a sufficient number of coded packets to be able to decode an retrieve the source packets.

In this survey, we summarized the research on security aspects of network coding. We started this survey with an introduction and background on network coding. We discussed the different applications of network coding and provided a classification of the network coding protocols. The security attacks are discussed in this work, and some of the solutions to make the network coding protocols robust against these attacks are presented. Moreover, we summarized the applications of network coding in providing network security, which includes defense against eavesdropper attack and secure secret key sharing.

ACKNOWLEDGMENT

This work is supported in part by NSF grants CNS 149860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, ECCS 1128209, and CNS 1138963.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

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

[4] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Communications of the ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM*, 2006, pp. 243–254.

[6] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM SIGCOMM*, 2007.

[7] D. Koutsonikolas, C. Wang, and Y. Hu, "CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *IEEE INFOCOM*, 2010, pp. 1–9.

[8] P. Ostovari, J. Wu, and A. Khreishah, "Network coding techniques for wireless and sensor networks," in *The Art of Wireless Sensor Networks*, H. M. Ammari, Ed. Springer, 2013.

[9] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782– 795, Oct 2003.

[10] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.

[11] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[12] P. Cataldi, M. Shatarski, M. Grangetto, and E. Magli, "Lt codes," in *IIH-MSP'06*, 2006, pp. 263–266.

[13] D. J. MacKay, "Fountain codes," *IEE Proceedings- Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[14] T. Matsuda, T. Noguchi, and T. Takine, "Survey of network coding and its applications," *IEICE transactions on communications*, vol. 94, no. 3, pp. 698–717, 2011.

[15] L. Lu, M. Xiao, M. Skoglund, L. Rasmussen, G. Wu, and S. Li, "Efficient network coding for wireless broadcasting," in *IEEE WCNC*, 2010, pp. 1–6.

[16] L. Lu, M. Xiao, and L. Rasmussen, "Relay-aided broadcasting with instantaneously decodable binary network codes," in *ICCCN*, 2011, pp. 1–5.

[17] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.

[18] H. Djandji, "An efficient hybrid arq protocol for point-to-multipoint communication and its throughput performance," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1688–1698, 1999.

[19] B. Zhao and M. Valenti, "Practical relay networks: a generalization of hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7–18, 2005.

[20] ——, "The throughput of hybrid-ARQ protocols for the gaussian collision channel," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1971–1988, 2001.

[21] G. C.Clark and J. B. Cain, *Error-correction coding for digital communications*. Springer, 1981.

[22] S. Lin and D. J. Costello, *Error control coding: Fundamentals and Applications*. Prentice-hall Englewood Cliffs, NJ, 2004.

[23] W. Ryan and S. Lin, *Channel codes: classical and modern*. Cambridge University Press, 2009.

[24] S. L. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: When is ecc energy-efficient?" *EURASIP Journal on Wireless Communications and Networking*, vol. 17, no. 2, pp. 29–29, 2006.

[25] M. Vuran and I. F. Akyildiz, "Error control in wireless sensor networks: a cross layer analysis," *ACM Transactions on Networking, IEEE*, vol. 17, no. 4, pp. 1186–1199, 2009.

[26] G. Fox, "Peer-to-peer networks," *Computing in Science & Engineering*, vol. 3, no. 3, pp. 75–77, 2001.

[27] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-peer Networking and Applications*, vol. 1, no. 1, pp. 18–28, 2008.

[28] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and challenges of peer-to-peer internet video broadcast," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 11–24, 2008.

[29] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *IEEE INFOCOM 2005*, vol. 4, 2005, pp. 2235–2245.

[30] S. Pawar, S. El Rouayheb, H. Zhang, K. Lee, and K. Ramchandran, "Codes for a distributed caching based video-on-demand system," in *IEEE ASILOMAR*, 2011, pp. 1783–1787.

[31] H. Zhang, M. Chen, A. Parekh, and K. Ramchandran, "A distributed multichannel demand-adaptive p2p vod system with optimized caching and neighbor-selection," in *SPIE*, 2011, pp. 81 350X–81 350X.

[32] C. Fragouli and A. Markopoulou, "A network coding approach to overlay network monitoring," in *Allerton*, 2005.

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

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

[35] D. Goldschlag, M. Reed, and P. Syverson, "M. reed and p. syverson and d. goldschlag," *IEEE Journal on selected Areas in Communications*, vol. 16, no. 4, pp. 482–494, 1998.

[36] P. Syverson, D. Goldschlag, and M. Reed, "Anonymous connections and onion routings," in *IEEE SP*, 1997, pp. 44–54.

[37] P. Zhang, C. Lin, Y. Jiang, P. P. Lee, and J. Lui, "ANOC: Anonymous network-coding-based communication with efficient cooperation," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1738–1745, 2012.

[38] C. Gkantsidis, P. Rodriguez *et al.*, "Cooperative security for network coding file distribution." in *IEEE INFOCOM*, vol. 3, 2006, p. 5.

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

[40] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," *International Journal of Information and Coding Theory*, vol. 1, no. 1, pp. 3–14, 2009.

[41] D. C. Kamal, D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *CISS*, 2006.

[42] K. Han, T. Ho, R. Koetter, M. Medard, and F. Zhao, "On network coding for security," in *IEEE MILCOM*, 2007.

[43] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," in *IEEE ISIT*, 2007, pp. 556–560.

[44] 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, no. 1.

[45] T. Shang, F. Huang, T. Peng, and J. Liu, "A deep detection scheme against pollution attacks in wireless inter-flow network coding," in *CSNT*, 2015, pp. 102–106.

[46] W. Huang, T. Wang, X. Hu, J. Jang, and T. Salonidis, "Rateless and pollution-attack-resilient network coding," in *IEEE ISIT*, 2015, pp. 2623–2627.

[47] A. Fiandrotti, R. Gaeta, and M. Grangetto, "Simple countermeasures to mitigate the effect of pollution attack in network coding-based peer-to-peer live streaming," *IEEE Transactions on Multimedia*, vol. 17, no. 4, pp. 562–573, 2015.

[48] J. P. Vilela, L. Lima, and J. Barros, "Lightweight security for network coding," in *IEEE ICC*, 2008, pp. 1750–1754.

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

[50] M. Shao, S. Dumitrescu, and X. Wu, "Layered multicast with inter-layer network coding for multimedia streaming," *IEEE Transactions on Multimedia*, vol. 13, no. 99, pp. 353–365, 2011.

[51] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *ACM CCR*, 1996, pp. 117–130.

[52] M. Kim, D. Lucani, X. Shi, F. Zhao, and M. Médard, "Network coding for multi-resolution multicast," in *IEEE INFOCOM*, 2010, pp. 1–9.

[53] N. Shacham, "Multipoint communication by hierarchically encoded data," in *IEEE INFOCOM*, 1992, pp. 2107–2114.

[54] M. Effros, "Universal multiresolution source codes," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2113–2129, 2001.

[55] J. Tan and M. Médard, "Secure network coding with a cost criterion," in *IEEE WiOpt*, 2006, pp. 1–6.

[56] L. Lima, M. Médard, and J. Barros, "Random linear network coding: A free cipher?" in *IEEE ISIT*, 2007, pp. 546–550.

[57] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *IEEE NetCod*, 2005.

[58] P. Zhang, Y. Jiang, C. Lin, Y. Fan, and X. Shen, "P-coding: secure network coding against eavesdropping attacks," in *IEEE INFOCOM*, 2010, pp. 1–9.

[59] P. Zhang, C. Lin, Y. Jiang, Y. Fan, and X. Shen, "A lightweight encryption scheme for network-coded mobile ad hoc networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 9, pp. 2211–2221, 2014.

[60] P. F. Oliveira, L. Lima, T. T. Vinhoza, J. Barros, and M. Médard, "Trusted storage over untrusted networks," in *IEEE GLOBECOM 2010*, 2010, pp. 1–5.

[61] ——, "Coding for trusted storage in untrusted networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1890–1899, 2012.

[62] P. Ostovari and J. Wu, "Fault tolerant and secure distributed data storage using random linear network coding," in *WiOpt*, 2016, pp. 1–8.

[63] P. F. Oliveira, R. A. Costa, and J. Barros, "Mobile secret key distribution with network coding," *Bernoulli*, vol. 1, p. 2, 2007.

[64] N. B. Shah, K. Rashmi, and K. Ramchandran, "Secure network coding for distributed secret sharing with low communication cost," in *IEEE ISIT*, 2013, pp. 2404–2408.

[65] N. Shah, K. Rashmi, and K. Ramchandran, "Distributed secret dissemination across a network," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1206–1216, 2015.

[66] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.