Universidade de Aveiro Departamento de
2017 Electrónica, Telecomunicações e Informática,

Alireza Esfahani

Esquemas de Segurança contra Ataques de Poluição em Codificação de Rede sobre Redes sem Fios

Security Schemes Against Pollution Attacks in Network Coding over Wireless Networks

Programa de Doutoramento em Telecomunicações das Universidades do Minho, Aveiro, e Porto

Universidade do Minho

Universidade de Aveiro

Universidade do Porto
U.PORTO


Universidade de Aveiro Departamento de
2017 Electrónica, Telecomunicações e Informática,

Alireza Esfahani

Esquemas de Segurança contra Ataques de Poluição em Codificação de Rede sobre Redes sem Fios

Security Schemes Against Pollution Attacks in Network Coding over Wireless Networks

Apoio financeiro da FCT com a
referéncia SFRH/BD/102029/2014.

o júri / the jury

presidente / president

Carlos Fernandes da Silva
Professor Catedrático da Universidade de Aveiro (por delegação do Reitor da Universidade de Aveiro)

vogais / examiners committee

Emmanouil Antonios Panaousis
Senior Lecturer of Cyber Security and Privacy
School of Computing, Engineering, and Mathematics
University of Brighton

Simon John Shepherd
Fellow of the Royal Society of Medicine
School of Engineering
University of Bradford

Aníbal Manuel de Oliveira Duarte
Professor Catedrático
Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

Alexandre Júlio Teixeira Santos
Professor Associado com Agregação
Departamento de Informática
Universidade do Minho

Jonathan Rodriguez Gonzalez (Orientador)
Professor Associado Convidado
Departmento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

acknowledgements

Although most likely the acknowledgments is the most fun part of a Ph.D. thesis to read, it is one of the hardest part to write. Actually, pursuing a Ph.D. degree is a challenging and exciting journey, and like in all journeys, the people that you travel with, make the experience much more enriching and enjoyable. Now, at the end of my Ph.D., I would like to acknowledge here the people who have been greatly influential in my graduate life and accompanied me during this journey. I have been recalling all of these years that I spent at Minho, Aveiro, Porto universities, and IT as a Ph.D. student, and researcher, respectively. These few paragraphs are an attempt to express my deepest gratitude to all those who made such an exciting experience possible.

First and foremost, my deepest gratitude goes to my adviser, Jonathan Rodriguez, for not only his help and advices as a Ph.D. supervisor, but also for that he was always being to me a source of inspiration and motivation in all aspects of life. I have learned a lot from Jonathan about conduction research, managing a research group, and combining personal and professional life successfully. I would also like to thank my co-advisor, Prof. Jose Carlos Neves, for his always wise words, most generous help, advice and being there at all times. Moreover, having a very nice, kind, responsible, and supportive character, working under their supervisions made my Ph.D. study a great experience full of absolute joy and fun.

I am indebted to Georgios Mantas for all of his kind and valuable help and support during my study. Although he was not officially my co-adviser, he treated me like one of his students, always available to answer my questions or discuss about various problems. He always patiently listen to me and helped me solving my problems. His constant support (both technical and non-technical) and encouragement were invaluable.

I am deeply grateful to the jury of my Ph.D. defense (composed by Professor Carlos Fernando Da Silva, the president of the jury, Professor Emmanouil Panaousis, Professor Simon Shepherd, Professor Manuel Oliveira de Duarte, Professor Alexandre Júlio Teixeira Santos) for giving me the honor of having them in my dissertation committee. I also thank them (and Jonathan) for carefully reading the thesis.

Based on the collaboration in SMARTVISION project, I was hosted by University of Madeira, during the first ten months of my Ph.D. I would like to thank University of Madeira, and also Alberto Nascimento for providing such a warm, supportive and friendly environment I had there which made it a place fun to work.

During my Ph.D. I was lucky to be integrated in the CODELANCE project and I could benefit from insightful discussions that taught me several aspects of secure network coding. I thank all the members of the CODELANCE project for these discussions, that inspired several improvements in the work present in this work.

I am very grateful to all 4TELL group member who made my Ph.D. time memorable. In particular, a special thank goes to Claudia Barbosa, who I could always count on her help, and support.

Above all, I cannot express enough my gratitude for my parents and parents-in-law for all that they have done for me in these five years. All of the achievements in my life were possible to their unconditional love, patience and support in every decision I make. They helped me through hard times during my Ph.D. work, and their Skype and phone calls to motivate me and retain my mental sanity.

Finally, I would like to thank my wife, for her constant support, understanding, trust and help. Her unending words of support were integral in pushing me through the toughest stages of my Ph.D. Thank you for believing in me at all times, even when I did not.

To MAHSA for all her love, help, and support.

Palavras-chave codificação de rede, aleatório codificação linear de rede, XOR codificação de rede, ataque a poluição de dados, ataque poluição Tag, redes sem fio

Resumo O assunto abordado nesta tese prende-se em como obter segurança eficiente contra ataques por poluição, explorando a estrutura de Codificação de Rede (CR).

Recentemente, tem havido interesse crescente no uso de técnicas de codificação de rede com o intuito de aumentar a robustez e o ritmo de transferência de dados de redes e reduzir o atraso nas redes sem fios, em que uma rede de esquema baseado na codificação aproveita a natureza aditiva dos sinais sem fios, permitindo que dois nós transmitam simultaneamente para o nó de retransmissão. No entanto, as redes sem fio activadas para CR são susceptíveis a ameaça segurança severa, conhecida como ataque por poluição de dados, onde um nó malicioso injecta na rede poluída (i.e., corrompidos) pacotes que impedem os nós de destino de descodificar correctamente. Devido à recodificação nos nós intermediários, de acordo com o princípio fundamental da CR, os pacotes poluídos propagam-se rapidamente nos outros pacotes legítimos corrompidos, levando a um desperdício de recursos de rede. Assim, uma série de investigações vêm sendo dedicadas a mecanismos de combate aos ataques de poluição dados. Os esquemas baseados em Código de Autenticação de Mensagem Homomórfica (MAC) são uma solução promissora de combate a ataques por poluição dados. No entanto, a maior parte destes são sensíveis a um novo tipo de ataque por poluição, chamado de ataque de poluição de etiqueta (Tag Attacck Pollution), onde um nó adversário modifica aleatoriamente etiquetas anexados ao final dos pacotes transmitidos.

Portanto, nesta tese, primeiro propomos um esquema de autenticação mensagem homomórfica baseado no código, proporcionando resistência contra ataques por poluição de dados e ataques por poluição de etiqueta em redes sem fio baseados em esquemas NC-XOR. Para além disso, propomos quatro esquemas baseados na codificação de mensagens de autenticação homomorficas, que fornecem resistência contra ataques por poluição de mensagens e de etiquetas na Codificação de Rede Linear Aleatório (RLNC). Nossos resultados mostram que os esquemas que propostos são mais eficientes em comparação com outros esquemas imunes à poluição de etiqueta em termos de complexidade, de overhead de comunicação e de overhead de armazenamento da chave.

Keywords Network coding, Random linear network coding, XOR network coding, Security, Data pollution attack, Tag pollution attack, Wireless networks

Abstract The topic of this thesis is how to achieve efficient security against pollution attacks by exploiting the structure of network coding.

There has recently been growing interest in using network coding techniques to increase the robustness and throughput of data networks, and reduce the delay in wireless networks, where a network coding-based scheme takes advantage of the additive nature of wireless signals by allowing two nodes to transmit simultaneously to the relay node. However, Network Coding (NC)-enabled wireless networks are susceptible to a severe security threat, known as data pollution attack, where a malicious node injects into the network polluted (i.e., corrupted) packets that prevent the destination nodes from decoding correctly. Due to recoding at the intermediate nodes, according to the core principle of NC, the polluted packets propagate quickly into other packets and corrupt bunches of legitimate packets leading to network resource waste. Hence, a lot of research effort has been devoted to schemes against data pollution attacks. Homomorphic Message Authentication Code (MAC)-based schemes are a promising solution against data pollution attacks. However, most of them are susceptible to a new type of pollution attack, called tag pollution attack, where an adversary node randomly modifies tags appended to the end of the transmitted packets.

Therefore, in this thesis, we first propose a homomorphic message authentication code-based scheme, providing resistance against data pollution attacks and tag pollution attacks in XOR NC-enabled wireless networks. Moreover, we propose four homomorphic message authentication code-based schemes which provide resistance against data and tag pollution attacks in Random Linear Network Coding (RLNC). Our results show that our proposed schemes are more efficient compared to other competitive tag pollution immune schemes in terms of complexity, communication overhead and key storage overhead.

Table of contents

List of Abbreviations

AES	Advanced Encryption Standard	13
ARQ	Automatic Repeat Request	23
CPSs	Cyber Physical Systems	45
DES	Data Encryption Standard	12
DHP	Diffie–Hellman Problem	57
DLP	Discrete Logarithm Problem	57
DoS	Denial of Service	49
FEC	Forward Error Correction	23
HHF	Homomorphic Hash Function	56
HMAC	Homomorphic Message Authentication Code	56
ICT	Information and Communications Technology	1
IoT	Internet of Things	61
KDC	Key Distribution Center	64
LNC	Linear Network Coding	14
MANETs	Mobile ad hoc Networks	45
NC	Network Coding	73
PGP	Pretty Good Privacy	12
PKIs	Public-Key Infrastructures	56
RLNC	Random Linear Network Coding	117
SSL	Secure Sockets Layer	12
SG	Smart Grid	45
TCP	Transmission Control Protocol	12
WNs	Wireless Networks	115
WSNs	Wireless Sensor Networks	61
WWW	World Wide Web	45

List of Symbols

δ	Security parameter
$\mathbb{F}_p$	Symbol in finite field
$\underline{\mathbf{u}}_i$	Native packet
c	Number of compromised nodes
l	Number of MACs (tags)
$l^{'}$	Number of Dual MACs (tags)
m	Number of native packets
n	Number of symbols in each native packet
p	Finite field size
q	Security parameter
SV	Secret Value
α_i	Coefficient
$\mathbf{u}_i$	Coded packet
$\mathcal{K}'_S$	A second keys set assigned to the source node
$\mathcal{K}_S$	A first keys set assigned to the source node
$t'_{\mathbf{u}_i,l}$	A Dual MAC (tag)
$t_{\mathbf{u}_i,l}$	A MAC (tag)

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

> "You only live once, but if you do it right, once is enough."
>
> Mae West

We start by giving a brief overview of Network Coding (NC) concepts used throughout this thesis. Next, we outline the need for a secure NC approach in Wireless Networks (WNs). Afterwards, we discuss the motivation and objectives of the thesis. Then, the organization of this thesis is presented. Finally, we finish the chapter by providing an overview of the main contributions of the thesis.

1.1 Introduction

There is a strong demand in modern societies for pioneering Information and Communications Technology (ICT) services that will support modern social infrastructure. In the new urban environment model, technology is called to support citizens in terms of new paradigms of services, tested and proved to facilitate everyday life. With the appearance of new technologies (i.e., smart homes, smart cities, Internet of Things (IoT), smart factories, smart phones, and etc.) and applications (i.e., Facebook, Twitter, Instagram, and etc.), the request for "being connected" has been increased dramatically. This implies an increasing demand for higher throughput and more efficient use of communication networks and developing new techniques in the fields of wireless communication, which can be used as a base for creating smart services that would serve people's everyday life in modern societies.

Moreover, security concept in communications network of these technologies and applications is a crucial aspect of modern digital infrastructure. There are many aspects to security, addressing a range of natural adversaries and malicious threats. Also in communication networks, one of the important issues is that data remains confidential impervious to malicious or accidental eavesdropping. Not only should communications be resistant to link failures, unintentional interference, and natural imperfections in the transmission medium, but also it is necessary to provide reliable, authenticated communication in the presence of malicious tampering (where an adversary intelligently modifies transmitted signals), spoofing (adversary attempts to impersonate a legitimate user), and jamming (adversary injects random noise to disrupt communication). Furthermore, data integrity should be ensured.

Recent research efforts have shown that WNs can benefit from NC technology in terms of bandwidth, robustness to packet losses, delay and energy consumption [5–15]. However, NC running over multi-hop networks can be vulnerable to various information security threats such as eavesdropping and data injection/modification, also node compromising and traffic analysis creating a need for robust security mechanisms that guarantee the security of data distribution. Existing mechanisms rely on simple network coding operations to provide a robust and low complexity solution. Nevertheless, the presented solutions not only are still vulnerable to network pollution attacks but also have a lot of overhead, so these solutions are opening the door for new research initiatives towards "secure" network coding solutions. In order to appropriately position the key motivators and objectives for this thesis, we provide an overview of NC that provides the benefits of NC, as well as providing the reader with an insight towards the current vulnerabilities of secure NC-enabled wireless network.

1.2 Overview of NC

Ahlswede and his colleagues in [1], about 16 years ago, officially introduced the notion of network coding. In contrast to traditional store-and-forward networks, the incoming packets at the intermediate nodes in NC-enabled networks are not only routed or replicated but also be recoded. In traditional store-and-forward schemes, each intermediate node has a set of routing information such as a routing table that helps the node to find which paths are the best for relaying a received data to one of the sink nodes in the network. In other words, each intermediate node, based on the address of the final sink nodes and the information available in its routing table, finds the best and cost effective paths. However, the policies which allow the node to find the best paths depend on the employed routing protocol. Moreover, regardless of the particular applied data routing protocol and the process based on which the cost function is defined, in traditional store-and-forward schemes the incoming data is only replicated and copied on one or more of the outgoing paths at a given intermediate node. Therefore, using traditional store-and-forward methods, the content of each data packet remains untouched during its travel through the network unless noise or malicious nodes injects some change to it. Nevertheless, in NC networks, each node in the network has the ability to encode its incoming data in order to generate its outgoing data. Hence, for each node, NC uses an encoding function f_v to generate the output packets by taking as input the packets arriving at the node. We demonstrate the network coding operation at the node v as:

$$(y_1^v, \ldots, y_z^v) = f_v(x_1^v, \ldots, x_d^v) \tag{1.1}$$

where parameters z and d are the number of packets departing from and arriving at the node v respectively.

Figure 1.1 demonstrates how the capacity of a network is improved when the NC method is applied instead of the traditional store-forward method. In this demonstration, there are two nodes A and B, who transmit their packets a and b to each other through an intermediate node R, respectively. In the traditional store-forward method, A transmits its packet to R and after that B transmits its packet to R. Afterwards, R transmits the received packet from A to B and the received packet from B to A. On the other hand, in NC method, A and B transmit their packets a and b in similar way to the traditional store-forward method. However, R transmits only the XOR result of the two packets, $a \oplus b$, to both A and B simultaneously. This capability provides one transmission step less for NC method compared to the traditional

store-forward method. The two main approaches of NC are XOR network coding and Random Linear Network Coding (RLNC), which are discussed and explained in details in Chapter 2.

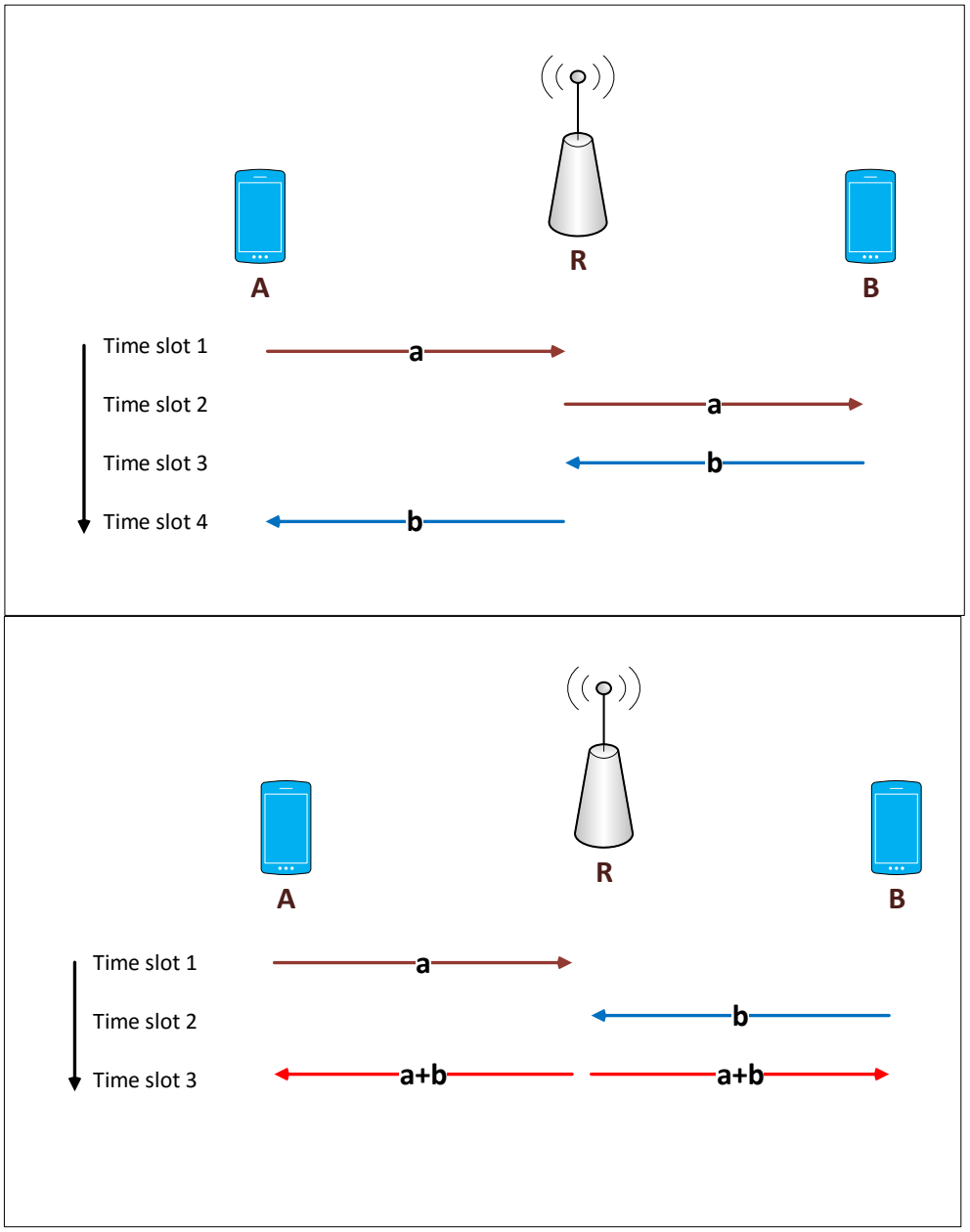

Figure 1.1: Traditional store-forward scheme and Network Coding examples of 3 nodes. To exchange messages a and b between nodes A and B, these nodes need to route their packets through node R. As it is shown on top, the traditional store-forward scheme would require four transmissions. However, if R is allowed to perform network coding with simple XOR operations, as illustrated in the lower diagram, $a \oplus b$ can be sent in one single broadcast transmission. By combining the received data with the stored message, A and B which possess a and b can recover b and a respectively. Thus, network coding saves one transmission.

1.3 Secure Network Coding

The heterogeneous and dynamic nature of wireless communications can raise many security challenges. More precisely, NC-enabled wireless networks can be the target of a wide spectrum of security attacks including eavesdropping attacks, Byzantine attacks, and pollution attacks among others. In this sense, a lot of effort has been placed on secure NC schemes against these types of attacks.

Due to the main principle of NC, according to which the intermediate nodes employ coding operations to mix (i.e. recode) the incoming packets, the NC-enabled networks are vulnerable to a severe security threat known as pollution attack. According to this kind of attack, an adversary injects into the NC-enabled network polluted packets that propagate quickly into other legitimate packets and corrupt bunches of them because of the recoding process that take place at the intermediate nodes of the NC-enabled network. Consequently, pollution attacks lead to network resource waste. In this sense, detection of pollution attacks is of utmost importance for NC-enabled networks. Pollution attacks can be considered as two types: data pollution attacks and tag pollution attacks. In data pollution attacks, the adversary targets at modifying (i.e. corrupting) the content of the transmitted packet. In tag pollution attacks, the adversary targets at modifying the tags appended to the end of the packets. A tag is a piece of information appended to the end of the coded packet to ensure its integrity. If the tag polluted packet is not detected, it can travel multiple hops or even reach the sink node leading to waste of network bandwidth. The presentation of data and tag pollution attacks is shown in Figure 1.2, where a node s sends its packet, which includes the data and tags, to the node R.

More precisely, in this thesis, we focus on the subject of security in network coding and propose various schemes to provide data and tag pollution attacks resistance.

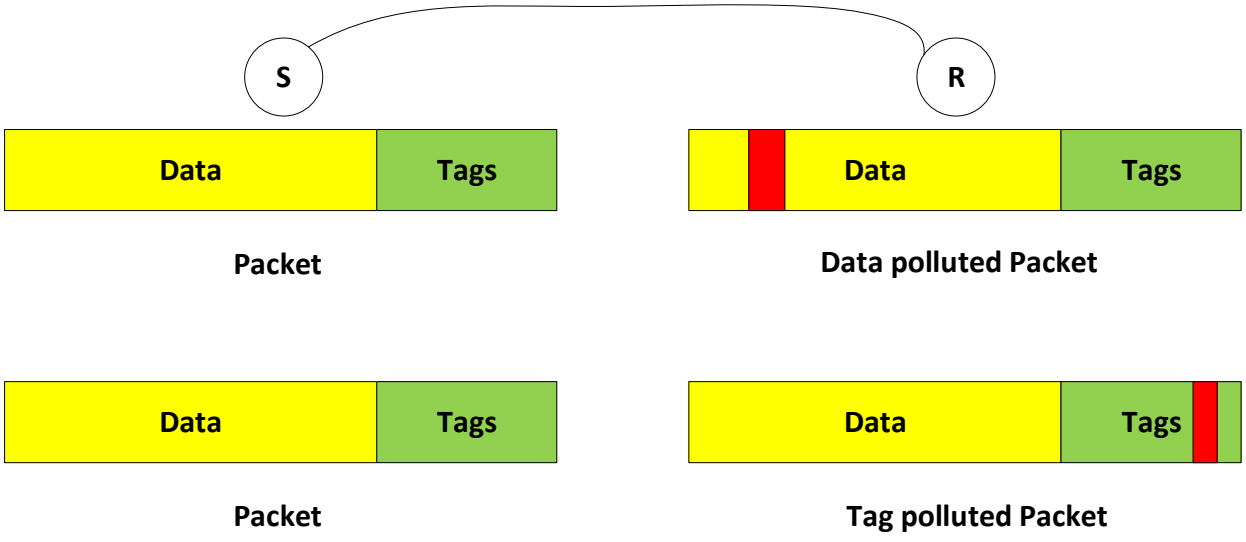

Figure 1.2: Data and tag pollution attacks illustrations.

We summarize the related work of secure NC schemes in Table 1.1. However, other related work specific to each topic can be found in each chapter.

Table 1.1: Secure network coding schemes

Attacks	Adversary target	Related work
Eavesdropping	monitoring information	[16]- [17]
Byzantine	selectively dropping packets, forwarding data packets through non-optimal or even invalid routes, generating routing loops, modifying or altering the message in transit, changing the header of packets	[18], [19], [20]
Pollution	**data pollution attacks:** the adversary targets at modifying (i.e. corrupting) the content of the transmitted packet	[2, 19–25]
	tag pollution attacks: the adversary targets at modifying the tags appended to the end of the transmitted packets	[3], [4], [26]

1.4 Motivation and Objectives

Wireless networks are characterized by low communication bandwidth, packet loss and power consumption constraints. To address these limitations, the application of Network Coding technology in wireless networks can be a promising solution since it can provide significant benefits including improved network capacity and robustness among others. However, the Network Coding-enabled wireless networks are vulnerable to various types of attacks that can compromise the security of the wireless networks. Based on that and the fact that the security is a critical factor for the success of wireless networks, we steer our focus on the mitigation of these types of attacks. In order to support secure data distribution over wireless multi-hop networks, the main objective of this PhD thesis is to design advanced security mechanisms to mitigate the impact of data corruption and protect transmission from adversaries or other intrusions. So motivated by the above observations, in this thesis we focus on the following objectives:

- **To propose a lightweight security scheme for XOR NC based systems**: a key enabler of Wireless Sensor Networks (WSNs) is a secure communications that needs to be robust and lightweight (in terms of complexity) to support end-to-end connectivity in highly dynamic wireless environments. Towards this direction, XOR NC is a promising solution for WSNs which have some limitations in terms of communication bandwidth, reliability and power consumption. Nevertheless, XOR NC is vulnerable to pollution attacks, where adversaries (i.e., compromised intermediate nodes) inject into the network corrupted packets that prevent the destination nodes from decoding correctly. This has, as a result, not only network resource waste but also energy waste at the intermediate nodes. In this sense, pollution attack constitutes a serious threat against WSNs (i.e., resource-constrained wireless networks) that should be addressed in order for WSNs to reap the benefits of XOR NC. To address this attack in XOR NC-enabled networks, a number of schemes have been proposed so far. But most of these schemes can detect the polluted packets only at the sinks. In this context, a key object of this thesis is to propose an efficient lightweight scheme for XOR NC-enabled networks which can detect the polluted packets before arriving to the sinks.

- **To propose efficient security schemes for RLNC based systems**: RLNC has been shown to be particularly beneficial for guaranteeing robust communication over dynamic wireless networks, even in the presence of disruptions such as corruption, packet loss, and congestion. However, similar to XOR NC, RLNC is susceptible to data and tag

pollution attacks, where an adversary (e.g., compromised intermediate node) pollutes the data and tag components of the packets that prevent the destination nodes from decoding it correctly. In this context, approaching an efficient security scheme for RLNC will be a key target. To address this security attack, MACs (Message Authentication Codes) have been proposed, but they use only a single domain tags to protect the integrity of the data that still leaves the packet vulnerable to pollution. In this context, a key objective is to extend this promising approach to investigate how we can provide robustness against both data and tag pollution.

1.5 Organization of the Dissertation

The remainder of the thesis is organized as follows:

Chapter 2: Principles of Security and Network Coding.
In this chapter, we present an overview of cryptography and security concept, as well as the mathematical techniques and notation that will be used throughout this thesis. Moreover, an overview and state-of-the-art analysis in the field of network coding is given. In order to position our work, we review two specific models of NC systems (i.e., XOR NC and RLNC), and the different metrics that can influence the NC system performance from a bandwidth, delay and computational overhead perspectives.

Chapter 3: Secure Network Coding: Countermeasures and Challenges
This chapter introduces briefly the concept of security requirements of the Network Coding-enabled WNs, as well as the vulnerability of Network Coding-enabled WNs to various types of attacks that can compromise their security. Based on that and the fact that the security is critical factor for the success of WNs, we steer our focus on the mitigation of these types of attacks. In particular, we intend to provide a basis for positioning our research efforts towards the deployment of the appropriate countermeasures against attacks targeting Network Coding-enabled WNs.

Chapter 4: Efficient MAC-based for Secure XOR NC.
In contrast to normal network coding operated over large finite fields, XOR network coding has gained an increasing number of applications for its simplicity, especially in wireless networks. Existing solutions to pollution attacks mostly can protect only the normal network coding, but a few of them are able to secure XOR network coding. This chapter proposes an efficient Message Authentication Code (MAC)-based scheme providing resistance against pollution attacks in XOR NC-enabled for WNs. Our proposed scheme makes use of a probabilistic key pre-distribution and MACs (tags). In our scheme, the source appends multiple MACs (tags) to the end of each native packet, where each MAC can authenticate only a part of the message, and the native packet authenticated by different MACs are overlapped.

Chapter 5: The Proposed Secure RLNC Schemes
Chapter 5 includes two parts. In part I, we propose two schemes that can provide resistance against pollution attacks. In the first proposed scheme, we present a homomorphic MAC-based scheme, called Dual-Homomorphic MAC (Dual-HMAC), for network coding-enabled wireless networks. The proposed scheme makes use of two types of tags (i.e., MACs and D-MACs) to provide resistance against data pollution attacks, and partially against tag pollution at-

tacks. Moreover, the second proposed scheme proposes an efficient Homomorphic Message Authentication Code-based scheme, called HMAC, providing resistance against data pollution attacks and tag pollution attacks in Network Coding-enabled Wireless Networks. In part II, we propose two more schemes. In the third scheme, we propose a null space-based homomorphic MAC scheme, for network coding-enabled wireless networks. Finally, in the last proposed scheme, we present a computationally efficient null space-based homomorphic MAC scheme, where due to its lightweight nature, it incurs minimal complexity compared to the other related schemes.

Chapter 6: Main Contributions and Future Work.
The conclusions of the thesis, along with the outline directions for future research in this field are summarized in this chapter.

1.6 Thesis Contribution

This PhD study mainly contributes towards providing lightweight security schemes that improve the robustness of several network applications. We aim to (i) achieve lightweight security using low-complexity procedures in XOR NC-enabled WNs, without adding to the complexity of protocols, and (ii) to achieve lightweight security using low-complexity procedures in RLNC-enabled WNs.

The main contributions and novelty of this PhD study lies in the following:

- We investigate existing techniques in the literature for network coding-enabled wireless networks. Therefore, we focus on a survey of existing NC-enabled WNs attacks that have potential a negative impact on the security of the whole network. Furthermore, we present mitigation approaches against the most common attacks, and position these categories. Our objective is to highlight the current limitations, and provide a basis on which to build more reliable and efficient defense mechanisms against known attacks for NC-enabled WNs.

- No such scheme was available in the open literature for XOR NC which can provide resistance against both data and tag pollution attacks; this thesis provides an efficient Message Authentication Code (MAC)-based scheme providing resistance against pollution attacks in XOR NC-enabled WNs. Moreover, we propose a technique for distributing the keys required for verification, which enables us to reduce tag pollution probability and not allow to a polluted packet to travel many hops. Furthermore, based on the proposed scheme and the key model applied in this scheme, we not only effectively speed up detection probability and time complexity of verification, but also save bandwidth overhead efficiency.

- We propose a novel homomorphic MAC scheme to address both data and tag pollution attacks for RLNC-enabled wireless networks. In this regard, existing schemes mainly focus on mitigating data pollution attacks, however, there exist a few related works [3], [4], and [26] which are also resistant against tag pollution attacks. However, these have had limited applications in wireless networks due to their high complexity. In this context, we extend the MAC scheme to the dual domain approach to protect against data and tag pollution. More precisely, we first introduce a scheme which uses a new idea of exploiting two types of MACs (tags). We call the second type of MACs as Dual MAC.

Furthermore, by adding a signature to the Dual MACs, we propose both data and tag pollution immune scheme. According to our scheme, the former tags (i.e. MACs and D-MACs) ensure the integrity of the packet and MACs and the latter one ensures the integrity of the D-MACs.

- In a different approach, we also propose the efficient homomorphic MAC scheme to promote secure NC communications. In this context, we exploit an efficient null space-based scheme that allows us to provide additional protection against tag pollutions from adversaries in contrast to current schemes that are either applied to linear NC, or to computationally complex. By using null space scheme, each node has not a fixed location in the coded packet, providing resistance against tag pollution attacks. In the proposed scheme, the tags are calculated based on null space properties, and then they are secretly swapped with the packet data. Hence, they are not appended to the end of the coded packet and thus, an adversary is not able to distinguish them from the packet data. Our results show that the proposed scheme is more efficient compared to the schemes proposed in [3] and [4], in terms of computational complexity without incurring additional communication overhead. To the best of our knowledge, the proposed schemes in [3] and [4] are the most competitive schemes in the literature for providing resistance against tag pollution attacks in RLNC.

The results of this study have resulted in eleven scientific works, four of which have been accepted and published in international peer-reviewed journals, these include:

- **Book Chapters**

  1. A. Esfahani, G. Mantas, D. Yang, A. Nascimento, J. Rodriguez, J. Neves, "**Toward Secure Network Coding-Enabled Wireless Sensor Networks in Cyber-Physical Systems**" in *Cyber Physical Systems: From Theory to Practice*, ISBN: 9781482263329, CRC-Taylor & Francis Group, USA, October, 2015.

- **Scientific Journal Papers**

  2. A. Esfahani, G. Mantas, and J. Rodriguez, "**An Efficient Null Space-based Homomorphic MAC Scheme against Tag Pollution Attacks in RLNC**" in IEEE Communications Letters, Vol. 20, No. 5, pp. 918 - 921, May, 2016.
  3. A. Esfahani, G. Mantas, J. Rodriguez, and J. Carlos Neves, "**An Efficient Homomorphic MAC-based Scheme against Data and Tag Pollution Attacks in Network Coding-Enabled Wireless Networks**" in Springer International Journal of Information Security, 2016.
  4. A. Esfahani, G. Mantas, H. Silva, J. Rodriguez, and J. Neves, "**An Efficient MAC-based Scheme against Pollution Attacks in XOR Network Coding-Enabled WBANs for Remote Patient Monitoring Systems**," EURASIP Journal on Wireless Communications and Networking, Vol. 2016, No. 1, pp. 1 - 10, April, 2016.
  5. A. Esfahani, D. Yang, G. Mantas, A. Nascimento, and J. Rodriguez, "**Dual-Homomorphic Message Authentication Code Scheme for Network Coding-Enabled Wireless Sensor Networks**," in Intrnl. Journal of Distributed Sensor Networks, Vol. Special issue, No. 15, pp. 1 - 10, January, 2015.

- **Scientific Conference Papers**

  6. A. Esfahani, G. Mantas, V. Monteiro, K. Ramantas, E. Datsika, and J. Rodriguez, "**Analysis of a Homomorphic MAC-based Scheme against Tag Pollution in RLNC-Enabled Wireless Networks**," in IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks - IEEE CAMAD, Guildford, United Kingdom, Vol. 2015, pp. 156 - 160, September, 2015.
  7. A. Esfahani, G. Mantas, J. Rodriguez, A. Nascimento, and J. Neves, "**A Null Space-based MAC Scheme against Pollution Attacks to Random Linear Network Coding**," in IEEE International Conference on Communications (ICC), London, United Kingdom, Vol. 2015, pp. 1521 - 1526, June, 2015 .
  8. A. Esfahani, D. Yang, G. Mantas, A. Nascimento, and J. Rodriguez, "**An Improved Homomorphic Message Authentication Code Scheme for RLNC-Enabled Wireless Networks**," in IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks - IEEE CAMAD, Athens, Greece, pp. 80 - 84, December, 2014.
  9. D. Yang, A. Esfahani, G. Mantas, and J. Rodriguez, "**Jointly Padding for Subspace Orthogonality against Tag Pollution**," in IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks - IEEE CAMAD, Athens, Greece, Vol. 2014, pp. 85 - 89, December, 2014.
  10. A. Esfahani, A. Nascimento, J. Rodriguez, and J. Neves, "**An Efficient MAC-Signature Scheme for Authentication in XOR Network Coding**," in IEEE International Symp. on Computers and Communications - ISCC, Madeira, Portugal, Vol. Workshops, pp. 1 - 5, June, 2014.
  11. A. Esfahani, A. Nascimento, D. Yang, and J. Rodriguez, "**A Mathematical Model for Improving Lightweight Security with Network Coding**," in IARIA International Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies - UBICOMM, Porto, Portugal, Vol. 2308-4278, pp. 159 - 162, October, 2013.

The organization of the thesis is illustrated in Figure 1.3.

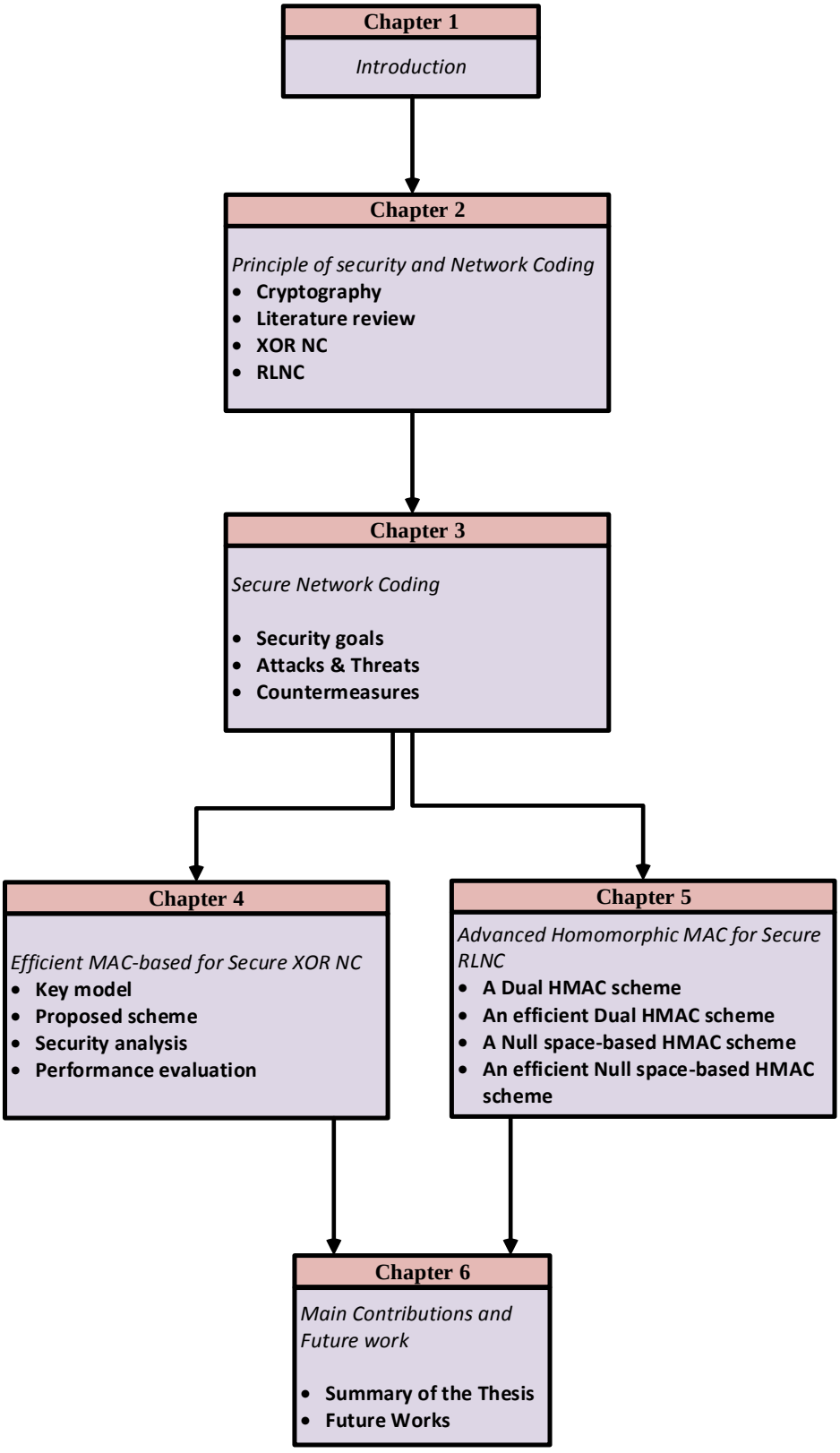

Figure 1.3: Thesis Organization.

Chapter 2

Principles of Security and Network Coding

"Science never solves a problem without creating ten more."

George Bernard Shaw

In this chapter, we first give a brief overview of classical cryptography followed by information-theoretic security. Then, previous works done in the field of Network Coding, as well as the frequently used notation and definitions that will be used throughout the thesis are introduced. Finally, a special emphasis is given to Random Linear Network Coding, where we end this chapter by describing the decoding probability of the random linear network coding schemes.

2.1 Security and Cryptography

Cryptography deals with the study of mathematical techniques related to information security aspects. In this regards, the authors in [27–29] proposed a complete discussion of cryptography where they categorize the goals of cryptography studies into five main groups: (*i*) *Authentication*, (*ii*) *Data confidentiality*, (*iii*) *Data integrity*, (*iv*) *Availability*, and (*v*) *Non-repudiation*. The first one, authentication, guarantees that two nodes entering into a communication authenticate each other. Data confidentiality prevents an unauthorized entity from accessing a data shared between authorized entities, while data integrity aims to prevent modification of communicated data by an untrusted party. Availability ensures that any resource is always available for any legitimate entity. Non-repudiation prevents an entity from denying previous activities. In this thesis, we will focus on security aspect of network coding-enabled wireless networks with relation to the data integrity.

To continue this section and introduce the terminology used thorough the thesis, we give a generic cryptographic protocol example for exchanging messages, as follows. There are two persons (i.e., Alice (the source) and Bob (the destination)) who want to communicate with each other through an wireless channel (it also could be as a wired, or fiber optic medium). The information that Alice wants to share with Bob is called the plaintext. The encrypted plaintext that Alice actually sends to Bob is called the ciphertext. The general idea behind

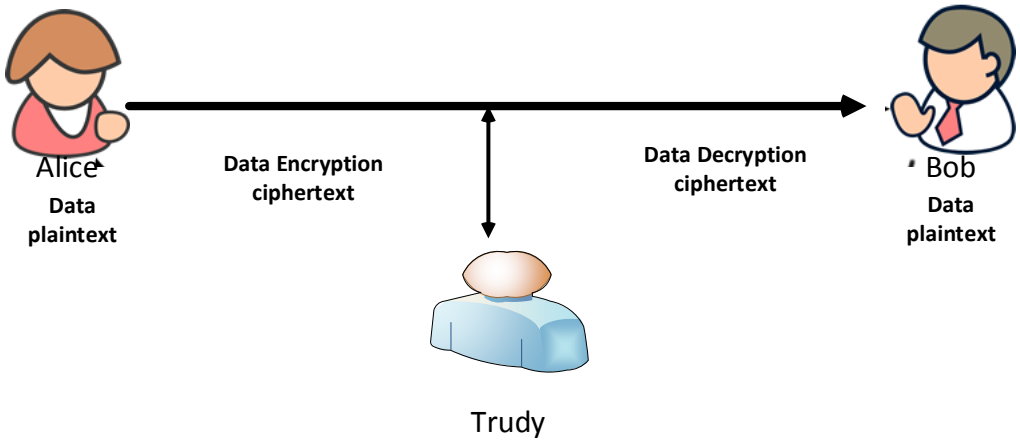

Figure 2.1: A general scenario of cryptography, where two legitimate parties (Alice and Bob) wish to protect the plaintext P from an adversary Trudy who acquires the ciphertext.

any cryptosystem is that Alice and Bob must share a secret key which is used to encrypt a message. Moreover, for maintaining the security framework between source and destination, cryptographic techniques allow a source to conceal data so that an intruder (i.e., Trudy acts as an intruder) can gain no information from the original data. Furthermore, the destination must be able to recover the original data from the disguised data. Figure 2.1 illustrates a general scenario which is used in cryptography concepts.

Generally, cryptography can be classified into symmetric-key cryptography and public-key (asymmetric-key) cryptography. In a symmetric cryptography, all participants share the same key which is used for both encryption and decryption. By using the same key and a decryption algorithm, a destination can recover the plaintext from the ciphertext. In addition, there are two broad classes of symmetric encryption techniques: block ciphers and stream ciphers [28]. The main idea in block ciphers is dividing the plaintext into some large (e.g. 128 bits) blocks, then encoding each block separately and decoding each block by participating only the last block and same key. Block ciphers are used in many secure Internet protocols like Pretty Good Privacy (PGP) (for secure e-mail), Secure Sockets Layer (SSL) (for securing Transmission Control Protocol (TCP) connections), and IPsec (for securing the network-layer transport) [28]. In other side, stream cipher divides the plaintext into small blocks (e.g. 1 bit) and encoding each block by participating many previous blocks and different keys. The two most popular symmetric cryptography standards are discussed briefly as follows:

- **DES (Data Encryption Standard)** The Data Encryption Standard (DES) has been adopted by the National Bureau of Standards in 1977. Data in DES are encrypted in 64-bit blocks and also it uses a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.

- **AES (Advanced Encryption Standard)** Because of the short key, DES could not satisfy the security requirements in practice and also considering to the very short

estimation time of cracking key in DES, in 2001, the National Institute of Standards and Technology (NIST) introduced the Advanced Encryption Standard (AES). The idea is replacement for DES as the approved standard for a wide range of applications. AES uses 128-bit blocks and can operate with keys that are 128, 192, and 256 bits long.

As mentioned earlier in this chapter, in symmetric encryption, all participants need to communicate with each other and have an agreement and share a secret key that this key is used for encryption and decryption. Thus, how to distribute and manage keys among legitimate participants in an efficient and secure way, is the main challenge of symmetric cryptography technique. However, public-key cryptography method is a form of cryptosystem in which encryption and decryption are performed using different keys: one a public key and one a private key. The public key can be published, but the private key should be kept secret. In public-key cryptography method, the idea is transforming plaintext into ciphertext by using a public key and an encryption algorithm. Using the private key and a decryption algorithm, the plaintext is recovered from the ciphertext. Suppose again the example described in previous section, where Alice wishes to send a data message to Bob (see Figure 2.2). Bob could share his public key to all nodes (legitimate nodes and attackers) and keep his private key to himself. Then, Alice would encrypt the plaintext P using Bob's public key pu_k and send the resulting ciphertext $Enc_{pu(k)}(p)$ to Bob. Finally, Bob uses his private key pr_k to decrypt the ciphertext. The key should be large enough for it to be computationally infeasible to derive the secret key from the public key via brute-force search. The three well-known representatives of public-key cryptography schemes are RSA, Diffie–Hellman key exchange and ECC [29].

- **Rivest-Shamir-Adleman (RSA):** RSA is one of the first practical public-key cryptosystems and is widely used for secure data transmission which introduced for the first time by Ron Rivest, Adi Shamir, and Len Adleman in 1977. This scheme has since that time reigned supreme as the most widely accepted and implemented general-purpose approach to public-key encryption. The RSA algorithm is a block cipher in which the plaintext and ciphertext are integers between 0 and $n-1$. A typical size for n is 1024 bits. The RSA algorithm involves four steps: key generation, key distribution, encryption and decryption. RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. The intention is that messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. However, it is shown that four possible approaches to attacking the RSA algorithm are as follows [29]:
  - Brute force: This involves trying all possible private keys.
  - Mathematical attacks: There are several approaches, all equivalent in effort to factoring the product of two primes.
  - Timing attacks: These depend on the running time of the decryption algorithm.
  - Chosen ciphertext attacks: This type of attack exploits properties of the RSA algorithm.
- **Diffie–Hellman key exchange (DHP):** This protocol enables two users to establish a secret key using a public-key scheme based on discrete logarithms. The protocol is secure only if the authenticity of the two participants can be established.

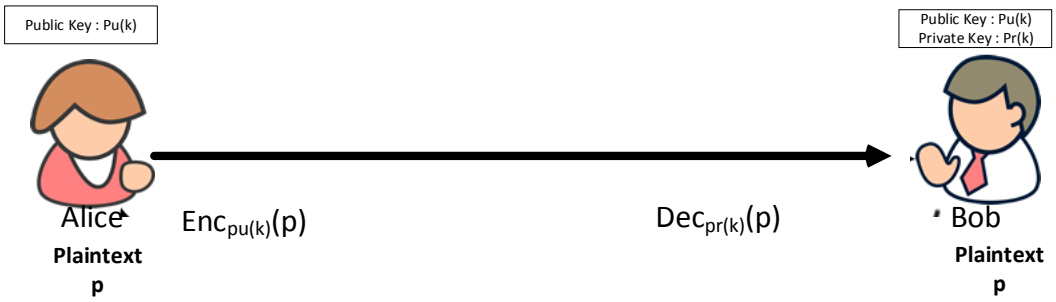

Figure 2.2: Public-key cryptography model.

- **Elliptic Curve Cryptosystem (ECC):** The key length for secure RSA use has increased over recent years, and this has put a heavier processing load on applications using RSA. In this regard, ECC is showing up in standardization efforts, including the IEEE P1363 Standard for Public-Key Cryptography. The principal attraction of ECC, compared to RSA, is that it appears to offer a smaller key size, thereby reducing processing overhead. Accordingly, the confidence level in ECC is not yet as high as that in RSA.

2.2 Network Coding

2.2.1 Related Works and Existing Schemes

The following brief review on some of the current works and existing schemes allows us to provide the framework for the proposed schemes, including general assumptions and the particular notations that are going to be used in this thesis.

According to the Max-flow Min-cut Theorem [30, 31], which proved that the max-flow min-cut capacity of a general multicast network can only be achieved by allowing intermediate nodes to mix different data flows, the first appearance of NC was proposed by Ahlswede *et. al* [1] in 2000. After the first appearance of NC, the authors in [16] demonstrated a model for a collection of subsets of wiretap channels for an NC system called wiretap network. Subsequently, a solution for error correction is presented in [32]. Furthermore, a different approach to NC proposed in [33], where an algebraic framework is derived to verify the feasibility of the multicast problem. Afterwards, the authors in [34] modified the Max-flow Min-cut Theorem to the problem of finding a point on an algebraic set. Moreover, the authors proposed the multicast problem in a Linear Network Coding (LNC) scenario [34]. They also investigated an upper and lower bound required for the alphabet size of the network codes. In this work, the authors explained the relation between a linear dispersion and a generic network coding. Additionally, they found a relation on the sizes of the base fields of the code. Moreover, the theory of codes on graphs was used to provide deterministic polynomial time algorithms and randomized algorithms for linear network coding in directed acyclic graphs [35]. They also realized a connection between linear network coding and linear system theory, especially with the theory of codes on graphs. A simple presentation of linear network coding scenario is

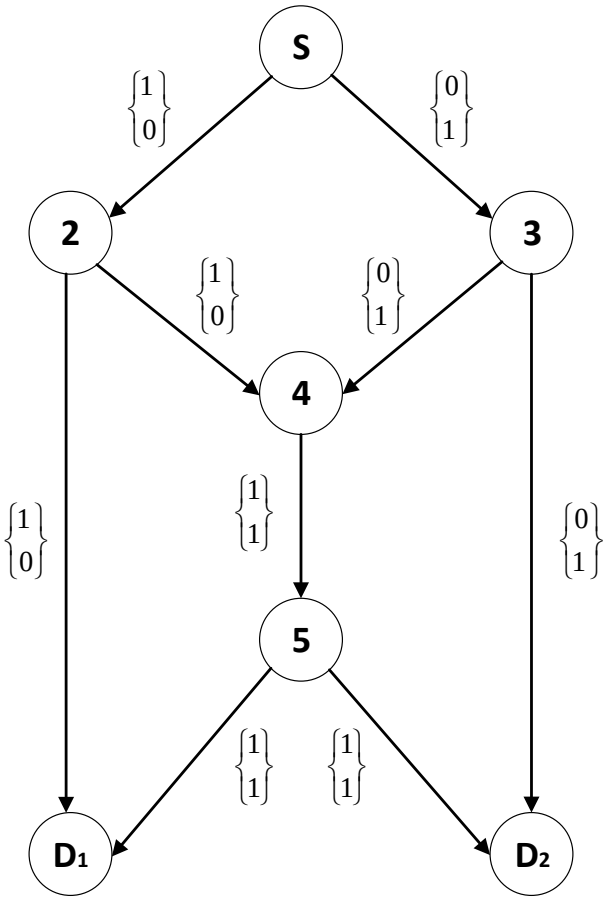

Figure 2.3: Illustration of linear network coding.

shown in Figure 2.3. The vector assigned to an outgoing link from each node is the linear combination of the corresponding vectors assigned to the incoming links. Let's suppose that there are two bits x_1 and x_2, where the source node aims to send them through the network to the destination nodes D_1 and D_2. These two bits can be written as a vector $[x_1 \quad x_2]$. Based on this scenario, $[x_1 \quad x_2] \times [1 \quad 0]^T$, which matches to x_1, is sent through the channel between the source and node 2. Furthermore, the node 4 receives $[x_1 \quad x_2] \times [1 \quad 1]^T$, which allows him to receive and forward $x_1 + x_2$ to node 5. Finally, at a destination node, the native bits x_1 and x_2 can be retrieved from the received vectors of each channel [36, 37].

Afterwards, a RLNC approach presented by Ho *et. al* in [18], where an original message is split into packets, called native packets, which are grouped into generations. Initially, the source encodes its native packets and then it multicasts them. Subsequently the intermediate nodes recode the incoming encoded packets by using independently and randomly coefficients from a finite field. Finally, the destination node is able to decode the incoming encoded packets when an efficient number of encoded packets of the same generation have been received. Decoding is only allowed for packets that belong to the same generation.

Due to its simplicity and distributed nature, RLNC quickly became the basis for several

network coding protocols such as Mobile ad hoc Networks (MANETs), peer-to-peer content distribution networks [6,38], wireless networks [39], and sensor networks [40]. Based on the idea proposed in [41], the main advantage of RLNC is increasing the robustness through exploitation of path diversity. RLNC has been shown to be particularly beneficial for guaranteeing robust communication over highly dynamic networks with minimal or even no control information, in the presence of disruptions such as congestion, corruption, packet loss, and changes of topology.

Several authors have evaluated the performance of network coding for content distribution networks and also studied the potential impact of RLNC in content distribution networks [6,38]. The authors in [6] proposed a scheme for content distribution of large files which shows that network coding not only improves the performance in terms of expected file download time, but also improves the robustness of the system. Furthermore, based on [38], since each node forwards a random linear combination independently of the information present at other nodes, its operation is completely decentralized. Moreover, there is high probability of obtaining a linearly independent packet in each time when each node collects a random combination of packets from a randomly chosen node. Thus, the problem of redundant transmissions is considerably reduced.

The comparison of the performance of network coding with traditional coding measures in a distributed storage setting, where each storage location only has very limited storage space for each file, is presented by Acedanski *et. al* in [42]. It is shown that RLNC performs well without the need for a large amount of additional storage space required at a centralized server. More precisely, the benefits of RLNC in wireless environments with limited resources, either due to mobility or battery rarity, are studied in [40], which proposes an algorithm aimed at reducing the overhead of probabilistic routing algorithms with applications in delay tolerant networks. Other RLNC based schemes that increase reliability and robustness while reducing the incurred overhead can be found in [43, 44].

In addition, several research studies have shown that network coding is suitable for other applications, such as multimedia streaming [45–47], and multimedia streaming over peer-to-peer networks which can reduce the network traffic, help control the server load and improve the playback quality [48–53].

Technically, the works done in the scope of secure network coding can be classified in two sets of passive attack and active attack [54]. In passive attacks, the attacker aims to only observe the content of data symbols that flow on the wiretapped channels. This means passive attackers only threaten the data privacy. Therefore, the content of each packet remains intact even though essentially the packet information has been disclosed to the attacker. Active attacks try to disrupt the normal network operation and may also alter, corrupt, or delete the data packets being exchanged over the network.

In [55], the authors provide a brief review on the security aspects of stateless and state-aware network coding. The set of possible attacks is categorized into two types of active and passive attacks and based on the type of network code and attack, different countermeasure solutions are reviewed in this reference. Another review on the security aspects of linear network coding, various attack scenarios, the existing solutions, and the future works is presented in [56].

2.2.2 Network Coding Protocols

According to the paper presented in [55], NC protocols can be categorized into the following two main types of protocols: stateless and state-aware NC based protocols. This categorization

is based on whether the NC protocol makes use or does not make use of the network state information which includes information related to the state of the network such as network topology, link cost, and node location.

2.2.2.1 Stateless NC Protocols

As mentioned earlier, stateless NC based protocols do not rely on network state information to perform coding operations such as the mixing of data packets. In stateless NC, coding operations can work properly even in the presence of dynamically changing topologies, such as in Mobile Ad-hoc Networks (MANETs). For instance, Random Linear Network Coding (RLNC) [18] is a distributed stateless NC protocol in which a node without any network state information can randomly and independently choose a set of coefficients to perform a linear combination of two or more received data packets and then send the resulted packet to its output links. Stateless NC protocols are more immune against different types of attacks compared to the state-aware NC based protocols because of their independence of the network state information [57]. However, due to this independence, these protocols not only need more sophisticated coding operations, but also the network codes are required to be selected from a sufficiently large field in order to guarantee that encoding and decoding operations can perform correctly in the source and sink nodes. This results in increasing the data overhead in the network.

2.2.2.2 State-aware NC Protocols

State-aware NC protocols require that each node obtains partial or full network state information. A very famous state-aware NC based protocol is COPE which is presented in [58]. This protocol uses local information and network state information to achieve improvements in terms of throughput and robustness. More specifically, COPE proposes an opportunistic network coding in order to reduce the number of transmissions. In state-aware NC protocols, the fact that each node has knowledge of network state information such as the other nodes' location, network topology and link state, leads to security vulnerabilities.

2.2.3 Notation, Definitions, and Assumptions

2.2.3.1 Finite Fields

All encryption algorithms involve arithmetic operations on integers. There is a need to work in arithmetic defined over a field, if one of the operations that is used in the algorithm is division. For convenience and for implementation efficiency, we would also like to work with integers that fit exactly into a given number of bits with no wasted bit patterns. That is, we wish to work with integers in the range 0 through $2^n - 1$, which fit into an n-bit word [29]. A *field* F, denoted by $\{F, +, \times\}$, is a set of elements with two binary operations, called *addition* and *multiplication*, such that satisfy the basic laws of arithmetic. Furthermore, a *finite field*, is a field with a finite number of elements. The characterization of finite fields shows that every finite field is of prime-power order. Furthermore, finite fields with the same number of elements are isomorphic and may therefore be identified. A finite field $\mathbb{F}_p$ (in this thesis, we denote interchangeably as $\mathbb{F}_p$ and as $GF(p)$), where $p = 2^n$ is represented by the set $\{0, 1, ..., p-1\}$. By setting $n = 1$, p is a prime number and this set should be closed under integer addition and multiplication modulo p. In encryption algorithms, we aim to provide a uniform mapping

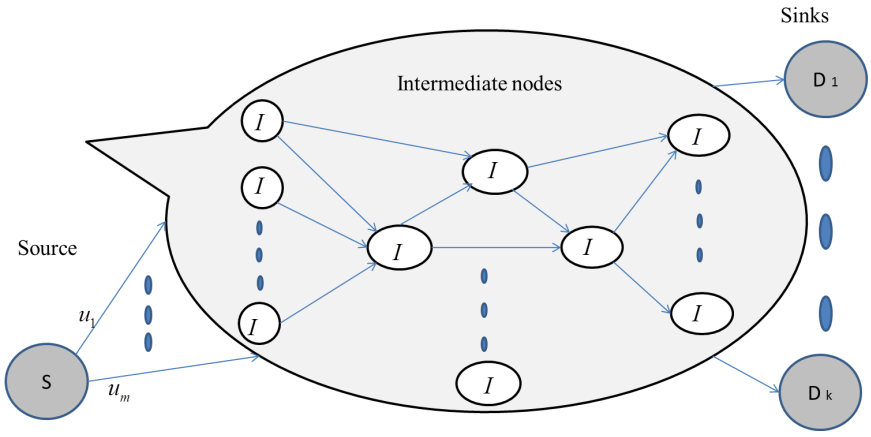

Figure 2.4: A general multicast network model.

to be cryptographically stronger. Thus, the finite fields of the form $\mathbb{F}_{2^n}$ are attractive for cryptographic algorithms. The entries of all multicast transmissions (where intermediate nodes linearly combine their incoming information symbols), vectors, and matrices considered in this thesis belong to a large enough finite field $\mathbb{F}_p$. For instance, $\mathbb{F}_2$ denotes the smallest finite field which contains two elements: 0 and 1, and the addition and multiplication tables shown in Tables 2.1 and 2.2 respectively.

+	0	1
0	0	1
1	1	0

Table 2.1: Addition in $\mathbb{F}_2$

×	0	1
0	0	0
1	0	1

Table 2.2: Multiplication in $\mathbb{F}_2$

2.2.3.2 Network Model

To study NC, we consider single-source multicast transmission on a network (see Figure 2.4), which is defined by a triple (G, S, R) consisting of the following components:

- *Acyclic and Directed multi-graph* G: We consider a pair of V, E as a directed acyclic graph where V represents the set of all nodes of G and E represents the set of all edges of G.
- *Source node* S: In our network model, we have a source S that transmits packets in multicast mode. The source is one of the nodes of V.
- *Non-source node* R: In our network model, we consider that the intermediate and destination nodes are included in a set of nodes which is defined as: $R = \{\forall x \in V - \{S\}\}$.

The two main approaches of NC are the XOR NC and RLNC. Two examples of these approaches based on the butterfly topology [1] are illustrated in Figure 2.5.a and Figure 2.5.b, respectively. Both of these examples include one source node (i.e., node 1), a few intermediate

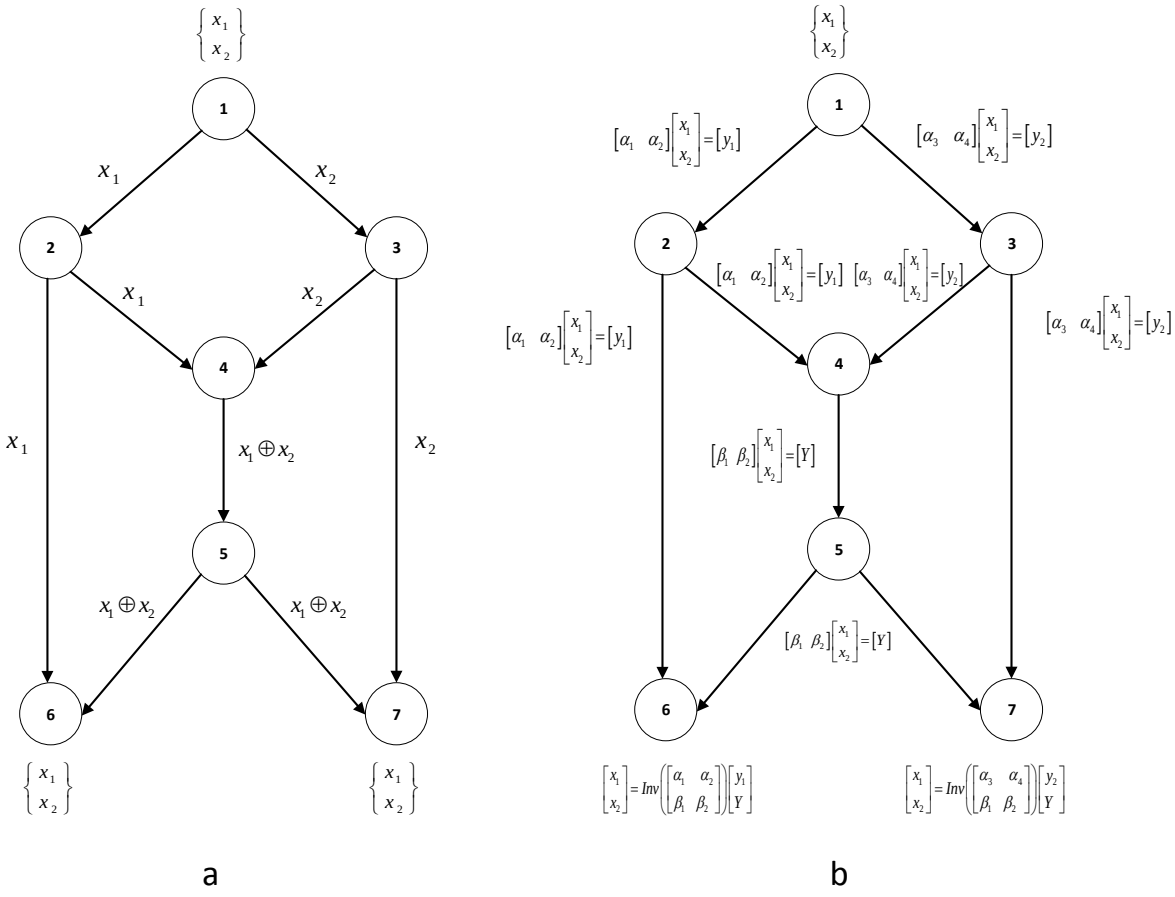

Figure 2.5: (a) XOR NC based on the butterfly topology and (b) RLNC based on the butterfly topology.

nodes (i.e., node 2, 3, 4 and 5) and two destination nodes (i.e., node 6 and 7). The source node wants to deliver the native packets x_1 and x_2 to both destination nodes through the intermediate nodes. In case of the XOR NC approach, node 4 XORing the two received packets (x_1 and x_2) from the two incoming nodes and sends the result of $x_1 \oplus x_2$ to node 5. Then, node 5 retransmits result of $x_1 \oplus x_2$ to the sink nodes. Finally, the two sink nodes decode the received result of $x_1 \oplus x_2$ and obtain x_1 and x_2, as it is depicted in Figure 2.5.a.

On the other hand, in the RLNC approach, node 4 creates the linear combination of x_1 and x_2. Then, the linear combination is transmitted to node 5, where the linear combination is recoded by using independently and randomly coefficients from a finite field. Afterwards, node 5 transmits the outcome to the sink nodes, where it is decoded and shown in Figure 2.5.b.

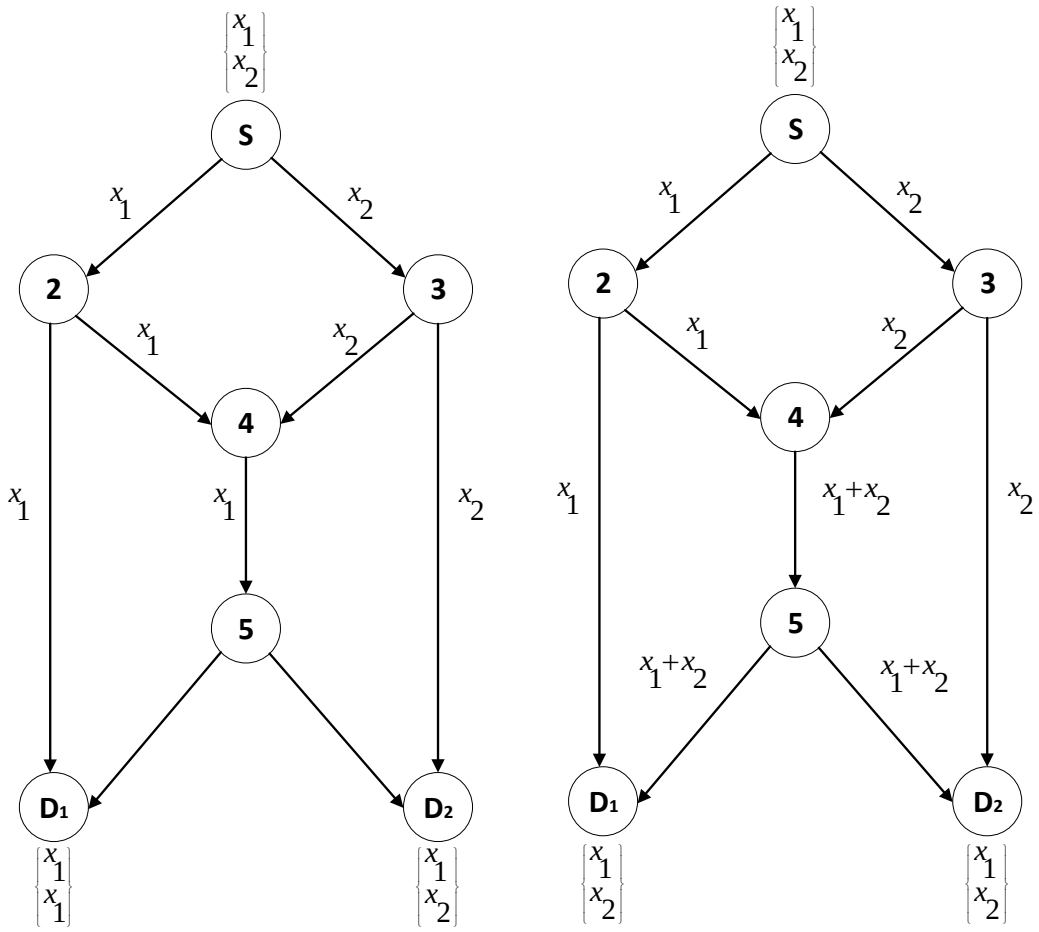

Figure 2.6: Canonical network coding example: node S multicasts bits x_1 and x_2 to nodes D_1 and D_2, respectively. If node 4 did not perform a simple encoding operation on the incoming bits x_1 and x_2, the max-flow min-cut capacity of 2 could not be achieved [1].

2.2.4 Benefits made by network coding

Recent research efforts have shown that wireless networks can benefit from network coding technique in terms of throughput, energy consumption, delay, and robustness to packet losses [1, 5, 34, 59–74]. These benefit characteristics are discussed as follows:

- ***Throughput***

  The following illustrates the idea behind the improvement in throughput improvement brought by network coding. This simple idea is illustrated in the canonical network coding example (known as the butterfly network), shown in Figure 2.6. Consider a simple setting, with a source node S, and two destinations D_1 and D_2. The source node S needs to deliver two bits, x_1 and x_2 respectively, to both destination nodes, using each network edge only once. Therefore, it sends bit x_1 to node 2, and bit x_2 to node 3, and each of these relay nodes (i.e., node 2 and node 3) further broadcasts the received bit. Consequently, destination D_1 and destination D_2 receive x_1 and x_2 respectively,

while relay node 4 obtains both bits. Based on the Max-flow Min-cut Theorem, the min-cut to each destination equals 2, and this cannot be achieved unless node 4 encodes x_1 and x_2 simultaneously. In traditional store-and-forward approaches, node 4 decides which of the two bits to forward. If for example it picks bit x_1, and further relay 4 broadcasts x_1 to both of the destinations, then D_1 obtains only bit x_1 (twice), and D_2 obtains both of the source bits. With network coding, the nodes are allowed to perform coding operations on the incoming data, thus node 4 forwards $x_1 + x_2$, which is then broadcasted by 5 to both D_1 and D_2. Consequently, receiver D_1 obtains x_1 from node 2 and $x_1 + x_2$ from node 5, which enables it to decode bit x_2 as well. Similarly, node D_2 decodes both bits as well. Thus, by means of network coding, the source S can deliver two bits simultaneously to both destinations.

The average throughput of network is anaylysed in [66], where a multicast model scenario is applied on either linear network coding scheme or random linear network coding scheme. The average throughput is the averaged rate that each sink node experiences. Their results have shown that network coding offers benefit in average throughput proportional to $\sqrt{n}$, where n is the number of sink nodes.

The gain in throughput efficiency made by network coding is not only constrained to the case of multicast, but also applicable to other traffic configurations, such as unicast transmissions [67]. As an example, COPE [39] is a technique that benefits from network coding in terms of throughput in wireless networks when the traffic is transmitting in unicast model. Furthermore, a theoretical analysis of the throughput gain in COPE-based network coding is presented in [67]. It is shown that network coding opportunities lead to higher end-to-end throughput when compared to no coding routing strategies.

- ***Energy efficiency***

  In [72], the minimum-energy multicast problem is studied by constructing network codes, where constitute a fully decentralized approach. The authors in [73] adopted the network coding approach to achieve minimum-cost multicast in both wireline and wireless networks. In this work, they applied distributed routing algorithms to solve the multicast subgraph optimization for fixed link capacities. They have shown that this has to be achieved through either deterministically or opportunistically scheduling transmitters' access to the wireless medium.

  Wu *et. al* proposed an alternative method for minimum-energy multicast in mobile ad hoc networks [75]. It is shown that the minimum energy multicast energy-per-bit can be obtained by performing network coding, but not traditional routing. A network coding scheme, called hybrid automatic repeat request with incremental redundancy (HARQ-IR) scheme, provides the energy efficiency of a two-way relaying system [76]. The proposed network coding scheme calculates the average energy consumption for signal transmission over block fading channel and concludes that network coding brings benefit in terms of energy efficiency to the two-way relaying networks.

  The work in [58] introduces a distributed protocol which supports multiple unicast flows efficiently by exploiting the shared nature of the wireless medium. Platz *et. al* apply linear network coding and random linear network coding schemes on ad-hoc networks [69]. The simulation results show that there are significant improvements in energy efficiency for these coded networks compared to the networks applying the traditional store-and-forward approach.

Since it is assumed that each transmission consumes the same amount of energy, the total energy is proportional to the number of transmissions. In this regards, the proposed scheme in [77] aims to minimize the number of transmissions. The energy efficiency under these assumptions are evaluated, showing that network coding can improve energy efficiency by a constant factor in fixed networks, and by a factor of $\log n$ in dynamically changed networks, where n is the number of nodes in the network [77].

- ***Delay***

  Network coding-enabled wireless networks may suffer from the delay which is caused from the encoding and decoding procedures [45–47, 78, 79]. It is shown that there is a trade-off between throughput efficiency and decoding delay [63]. More precisely, the more packets need to be encoded, recoded, and decoded to have a system with high throughput, which makes result in larger delay.

  The proposed model in [80] claimed that network coding can be used to reduce the delay in large files transmissions networks. They have shown that random linear network coding not only brings arbitrary gain in delay by scaling the system parameters, but also has the minimum delay compared to other non-coding schemes.

  In [74], firstly the relation between the average delay and the size of coding buffer is examined. Then, it is shown that network coding schemes enable a network to achieve the lower bound of the expected delay. Moreover, the upper and lower bounds of delay in multicast model is derived in [81]. The proposed scheme is based on an adaptive model which enables all destinations to find the maximum number of packets that can be encoded.

  The design of some real-time applications is considered in [82], where delay probability is investigated. In this regards, the two analyze models, Markov chain and brute-force, are considered, where a multicast model is used to apply RLNC over erasure channel. In the Markov chain analyze, their results show that even by choosing $GF(2)$, which is convenient for its low computational complexity, it bears the cost of a heavy tail in the delay distribution. However, in the brute-force analyze, the main idea is to select a large number of fixed erasure patterns and measure for every erasure pattern the delay of all possible encodings or sets of linear combinations of packet symbols. They have shown that their proposed model is feasible for small field size and small generation size, which yielding a characterization of the delay distribution under RLNC scenarios.

  In other work proposed in [78], the authors aim to examine the exact probability of decoding N linear independent packets among T received packets at a sink node. Their study shows that the number of received packet is proportional to the delay.

- ***Reliability***

  Previous works in network coding fields show that it is able to bring a more reliability in a network [83–92]. More precisely, in non-coded wireless network, several retransmissions (i.e., if a transmission fails due to the lost or dropped) of the native data packet from the source node are needed to retrieve it. Network coding is combined with retransmission techniques to improve the reliability of packets transmissions to multiple receivers [93–95].

  A proposed XOR-based coding scheme in [96] aims to combine the lost packets of different sink nodes. Their results show that the schemes who are using a dynamic network coding

benefit the smallest number of retransmissions. In this work, under different scenarios, the expected number of retransmissions for a packet to reach every sink node successfully is examined. Furthermore, the XOR-based network coding scheme, presented in [90], is applied to lossy networks to reduce the number of retransmissions. The main idea relies on encoding the lost packets of different nodes (sinks) and broadcasting the coded packet to all of them. They have shown that by using this idea and instead of transmitting individual lost packet separately, the total number of retransmissions can be reduced. Moreover, multiple sink nodes have the possibility to retrieve their lost packets in one transmission.

Fang *et. al* aim to find the optimal coding set of the lost packets so that the number of retransmissions is minimized [86]. This coding set is generated by a colouring-based heuristic algorithm. In this scheme, a packet-loss table, which contains the packet loss information of every sink nodes, is used to encode the lost packets.

Reference [97] proposed the network coding aided Automatic Repeat Request (ARQ). More specifically, a source node first finds the sink nodes who received the packets unsuccessfully, then, it employs network coding to broadcast a selected combination of those packets to the sink nodes. Due to the possibility of overhearing of all the packets, the number of retransmissions can be reduced. Furthermore, the reliability gain is characterized analytically in [87,98], where network coding is applied and compared with traditional error control protocols, such as ARQ and Forward Error Correction (FEC). More precisely, the authors, in the first proposed work [87], calculate the expected numbers of retransmissions by the source node under different error control protocols. Their numerical results show that the reliability gain made by network coding increases logarithmically. The authors prove this hypothesis in their latter work [98].

Improving the reliability by using opportunistic routing in wireless mesh networks when channel quality is poor is studied in [5, 99]. In MORE [5], a node creates a random linear combination of the packets, and does not decode the packets until they reach the destination. Consequently, the probability of successfully receiving and decoding packets at the sink nodes is raised and the reliability is improved as well.

2.2.5 The challenges in Network Coding

In this section, some challenges existing in the design of network coding schemes are summarized in two parts as follows:

- ***XOR Network Coding*** It is shown that XOR-based network coding schemes can benefit from low computational complexity, where encoding and decoding require only XOR operations. However, take into consideration that, an improper design of XOR coding may not only provide poor throughput, delay, energy efficiency, and reliability, but also bring negative effect to a network. This happens because of the lack of native packets in the sink nodes. For example, the coded packet $X_1 \oplus X_2 \oplus X_4$ is not a good coded packet to retrieve the native packets X_1, X_2, X_3 and X_4 which are transmitted by a source node to a destination node who has obtained only X_1, X_2. Thus, a network may suffer from decoding delay in order to collect a native packet (i.e., X_3, X_4). In addition, the number of native packets to be encoded affects the encoding and decoding delay because sufficient packets are required to be obtained before coding can proceed.

- ***Random Linear Network Coding***

  Based on the fact that real channels have errors and changing topology in the network, random linear network coding schemes are basically used for theoretical analysis [37]. However, there are several parameters which should be chosen carefully in real scenario of RLNC. The number of packets to be encoded is one of the important parameters. Several works have shown that this parameter can affect both throughput and delay of the network [39, 45, 47, 67]. Moreover, in RLNC schemes, we have to take into consideration that, other important parameter which can affect the overhead and also decoding probability is the size of finite field. It is shown that the larger finite fields enable the sink nodes to decode the more independent coded packets. However, it should be mentioned clearly, that increasing the finite field size brings more overhead.

Our summary and comparison of the existing challenges in network coding schemes is presented in Table 2.3.

Table 2.3: Comparison of existing challenges in NC

NC Model	Complexity	Reliability	Practical	Simulation [100]
XOR NC	low	low	easy	easy
RLNC	high	high	hard	hard

2.2.6 The principle of RLNC

Random Linear Network Coding is a distributed methodology for performing network coding, in which the source generates a linear combination of its original packets using random coefficients from a finite field. Furthermore, intermediate nodes perform this random linear combination using the coded packets previously received and those stored in the node's buffer with random coefficients. Each coded packet is sent along with the global encoding vector, which is the set of linear transformations that the original packet goes through on its path from the source to the destination. After transmission, when a destination node has obtained enough linearly independent coded packets, it can decode and recover the original packets by using Gaussian eliminations [18]. The operations of RLNC are summarized in Algorithm 1.

The key characteristic of RLNC is that coefficients are selected randomly over a finite field. This implies the transmission of coding vectors to the receivers by appending them as an overhead to the header of the data packets. Based on h and p, which are the number of information flows at the source node and the size of the finite field respectively, this overhead is calculated as $h \log p$ [101].

2.2.7 A General Model of RLNC

In our assumption, we consider three phases which are run by a source node, intermediate nodes, and destination nodes. First, a source node encodes the native data packets and transmits them into the network; next, an intermediate node recodes the incoming encoded packets and forwards them toward a destination node, and finally, the destination node decodes the incoming coded packets to retrieve the native data packets. Any packet and generation should follow the same three phases as illustrated in Figure 2.7. To continue this section, we consider that the source node sends g native data packets, where each of them is shown by size w, into the network. These three phases are described in details as follows:

Algorithm 1: Random Linear Network Coding Operations

main()
Initialization and encoding (**at source node** S):
The source S partitions each message into a number of packets (i.e., $u_1, ..., u_n$) for each generation. Then, it starts encoding the packets.
Recoding (**at intermediate node** V):
if Generation number of the received packet is same **then**
  Gaussian elimination is performed ;
  if the packet is non-innovative **then**
    it is discarded;
    else
      it is put into the buffer of node V sorted by generation number
  when there is a transmission opportunity, the node chooses all the packets that are in the buffer to make a random linear combination of them:

$$y = \sum_{i=1}^{h} c_i \mathbf{u}_i$$

  where each $c_i \in \mathbb{F}_p$ is a random coefficient chosen by each intermediate node.
Decoding (**at sink node** D):
if Generation number of the received packet is same **then**
  if Gaussian elimination is performed correctly **then**
    The sink node is able to retrieve the packets $u_1, ..., u_n$;
    else
      a decoding error occurs.

- **Source node:**

  First, the source node aims to generate g native data packets, where each of them can be defined as a vector $\mathbf{x_i} = [x_i^1, x_i^2, ..., x_i^w]$. $\mathbf{X}$ denotes the matrix of native data packet and is written as follows:

$$\mathbf{X} = \begin{bmatrix} x_1^1 & ... & x_1^w \\ & \vdots & \\ x_g^1 & ... & x_g^w \end{bmatrix}_{g*w} \tag{2.1}$$

  where each element of matrix $\mathbf{X}$ stands in a finite field $\mathbb{F}_p$. Then, the source node chooses a random matrix of coefficients $\mathbf{C}$ as follows:

$$\mathbf{C} = \begin{bmatrix} c_1^1 & ... & c_1^g \\ & \vdots & \\ c_g^1 & ... & c_g^g \end{bmatrix}_{g*g} \tag{2.2}$$

  An encoded data packet is denoted as following:

$$\mathbf{M} = \mathbf{C} \times \mathbf{X} = \begin{bmatrix} m_1^1 & ... & m_1^w \\ & \vdots & \\ m_g^1 & ... & m_g^w \end{bmatrix}_{g*w} \tag{2.3}$$

  Before transmitting, the source node concatenates the matrix $\mathbf{M}$ with the matrix of coefficients $\mathbf{C}$, which is called encoded matrix. The encoded matrix $\mathbf{B}$, which is represented in Equation 2.4, is sent to the intermediate nodes.

$$\mathbf{B} = [\, \mathbf{C} \mid \mathbf{X} \,] = \begin{bmatrix} \beta_1^1 & ... & \beta_1^{w+g} \\ & \vdots & \\ \beta_g^1 & ... & \beta_g^{w+g} \end{bmatrix}_{g*(w+g)} \tag{2.4}$$

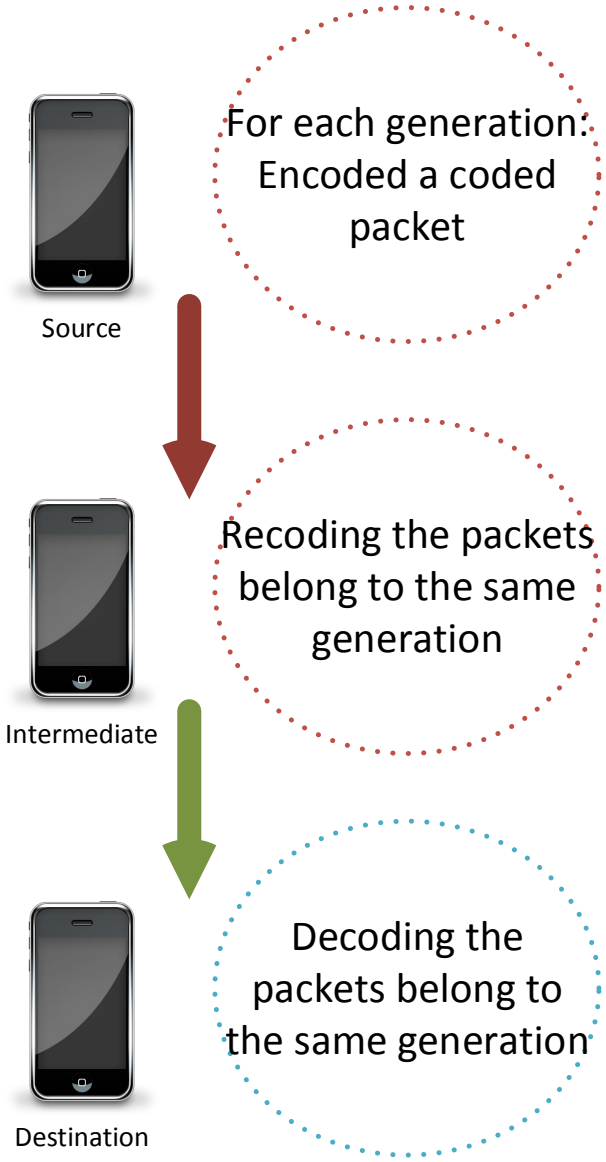

Figure 2.7: Three general phases in RLNC.

We also note that, each row of encoded matrix (e.g., $\mathbf{B_i} = [\beta_i^1 ... \beta_i^{g+w}]$) is called a codeword. To decode the native data packet, the destination node should receive at least g linear independent codewords.

- **Intermediate node:**

  Recoding operations, at the intermediate node, is started since it receives all g codewords from the source node. Then, the intermediate node makes a random matrix to generate

a coefficients matrix $\mathbf{D}$ as following:

$$\mathbf{D} = \begin{bmatrix} d_1^1 & \dots & d_1^g \\ & \vdots & \\ d_g^1 & \dots & d_g^g \end{bmatrix}_{g*g} \tag{2.5}$$

In the next step, recoding is done to make a new encoded packet as:

$$\mathbf{M}^{'} = \mathbf{D} \times \mathbf{B} = \begin{bmatrix} m^{'1}_{1} & \dots & m^{'w+g}_{1} \\ & \vdots & \\ m^{'1}_{g} & \dots & m^{'w+g}_{g} \end{bmatrix}_{g*(w+g)} \tag{2.6}$$

Finally, the new coded packet $\mathbf{M}^{'}$ is sent to the destination node. As mentioned in [102], several different policies for the recoding operations can be applied. Based on the policy which each intermediate node chooses, it starts buffering the input coming codewords. For instance, it may start the recoding process in each time interval or by having at least a number of codewords in its input buffer. Furthermore, the intermediate node may receive the codewords from different paths. Also, the intermediate node may mix codewords from different flows of data (called inter-flow or inter-session network coding), or it can recode and mix only the codewords of the same flow (called intra-flow or intra-session network coding) [103, 104].

- **Destination node:**

  The sink node should keep checking the rank of the total received codewords belonging to the same generation. It can immediately start the decoding when the sink node receives enough linear independent codewords (i.e., the rank of the received codewords, which is called innovative codewords, reaches to g). Take into consideration that each received codeword has two parts (i.e., header, which includes the coefficients, and trailer, which includes the codewords), the destination node forms the codewords and coefficients matrices by the following matrices:

$$\mathbf{T} = \begin{bmatrix} t_1^1 & \dots & t_1^w \\ & \vdots & \\ t_g^1 & \dots & t_g^w \end{bmatrix}_{g*w} \tag{2.7}$$

$$\mathbf{H} = \begin{bmatrix} h_1^1 & \dots & h_1^g \\ & \vdots & \\ h_g^1 & \dots & h_g^g \end{bmatrix}_{g*g} \tag{2.8}$$

  Based on linear combinations which the matrix $\mathbf{T}$ is created, the destination node needs to find the solutions, via Gaussian elimination method, for the following equation:

$$\mathbf{X} = \mathbf{H^{-1}} \times \mathbf{T} \tag{2.9}$$

Now, we give an example for RLNC approach to show the RLNC process in details, as follows. Note that all the arithmetic operations such as addition, multiplication, addition inverse and

multiplicative inverse are defined and executed over the field $\mathbb{F}_{2^3}$. Before giving the example and for simplicity, the mathematic in $\mathbb{F}_{2^3}$ is illustrated in following Tables:

Table 2.4: Addition in $\mathbb{F}_{2^3}$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

Table 2.5: Multiplication in $\mathbb{F}_{2^3}$

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	3	1	7	5
3	0	3	6	5	7	4	1	2
4	0	4	3	7	6	2	5	1
5	0	5	1	4	2	7	3	6
6	0	6	7	1	5	3	2	4
7	0	7	5	2	1	6	4	3

Table 2.6: Addition and multiplication inverse in $\mathbb{F}_{2^3}$

w	$-w$	w^{-1}
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1

To continue, we assume that the source node aims to send three native data packets $x_1 = 7, x_2 = 2$, and $x_3 = 1$. The source node generates the native data matrix as follows:

$$\mathbf{X} = \begin{bmatrix} 7 & \leftarrow x_1 \\ 2 & \leftarrow x_2 \\ 1 & \leftarrow x_3 \end{bmatrix}_{3*1} \tag{2.10}$$

Then, the coefficients are generated as follows:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3*3} \tag{2.11}$$

The concatenation of theses matrices is shown as follows:

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \end{array}\right] \tag{2.12}$$

By using a random coefficients matrix, a new coded packets matrix is generated by the source node as follows:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \leftarrow x_1 + x_2 + x_3 \\ 2 & 0 & 1 & 4 \leftarrow 2 * x_1 + x_3 \\ 2 & 1 & 0 & 7 \leftarrow 2 * x_1 + x_2 \end{array}\right] \tag{2.13}$$

We call each row of coded matrix x'_1, x'_2, and x'_3, respectively.

$$\begin{bmatrix} x'_1 = 4 \leftarrow x_1 + x_2 + x_3 \\ x'_2 = 4 \leftarrow 2 * x_1 + x_3 \\ x'_3 = 7 \leftarrow 2 * x_1 + x_2 \end{bmatrix} \tag{2.14}$$

Since an intermediate node receives this matrix, starts recoding as follows:

$$\left[\begin{array}{ccc|c} 5 & 3 & 1 & 1 \leftarrow x'_1 + 2 * x'_3 \\ 0 & 1 & 1 & 3 \leftarrow x'_2 + x'_3 \\ 0 & 2 & 3 & 7 \leftarrow 2 * x'_1 + x'_2 \end{array}\right] \tag{2.15}$$

The received coded packet can retrieve the native data packet via Gaussian elimination:

$$\left[\begin{array}{ccc|c} 5 & 3 & 1 & 1 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 1 & 1 \end{array}\right] \tag{2.16}$$

which leads to retrieve the native packet as:

$$\left[\begin{array}{ccc|c} 5 & 3 & 1 & 1 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 1 & 1 \leftarrow x_3 = 1 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 5 & 3 & 1 & 1 \\ 0 & 1 & 1 & 3 \leftarrow x_2 = 2 \\ 0 & 0 & 1 & 1 \leftarrow x_3 = 1 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 5 & 3 & 1 & 1 \leftarrow x_1 = 7 \\ 0 & 1 & 1 & 3 \leftarrow x_2 = 2 \\ 0 & 0 & 1 & 1 \leftarrow x_3 = 1 \end{array}\right] \tag{2.17}$$

2.2.8 Analysis of RLNC

Apart of corruption or lossy channel, there are two main reasons such as (i) *Insufficient rank* and (ii) *Insufficient receiving codewords*, in which the sink node may fail to fully retrieve the native packets. These cases affect the error decoding probability [102], throughput, delay, redundancy, coefficients overhead, and the time required to recover the native data packets. Moreover, the capability of a receiver to successfully decode the native data packets depends on the number of sinks and the size of finite field. This also introduces a trade-off between the decoding error probability and the complexity of the coding operations over the finite field. To continue, we present the simulation results related to the performance of the RLNC mechanism in two different scenarios. In the first scenario, we aim to find out the error decoding probability of the general model presented in Figure 2.7, where the effect of different number of packets and finite field size are studied. In the second scenario, we aim to discover the time required to recover successfully the native data packets, where the effect of different number of packets and finite field size are considered in this scenario too.

1. ***Error decoding probability***
   Based on the network model of RLNC described in Section 2.2.7, the lower bound on the probability that the RLNC scheme supports the multicast flow is given by following Theorem [18]:

   Theorem 1. *[18] Consider a multicast connection problem on an arbitrary network with a source, and a network code in which some or all network code coefficients are chosen uniformly at random from a finite field $\mathbb{F}_p$ where $p>d$ (d is the number of sink nodes), and the remaining code coefficients, if any, are fixed. If there exists a solution to the network connection problem with the same values for the fixed code coefficients, then the probability that the random network code is valid for the problem is at least $(1-d/p)^{\eta}$, where η is the number of links associated with random coefficients.*

   In case of an acyclic network, following the Theorem 1 in [18], the lower bound on the error decoding probability, is calculated as following:

$$p_e = 1 - p_s = 1 - (1 - d/p)^{|E|} \tag{2.18}$$

   Where d is the number of sink nodes, $|E|$ corresponds to the number of edges, and p is the finite field size.

   We have considered the several parameters which can affect the network performance as follows:

   - **Number of receivers:**

     We illustrate this probability for different number of sink nodes in Figure 2.8. As it is shown in this figure, by increasing the number of receivers in the network, the error probability increases. Moreover, increasing the number of edges also affect the error decoding probability.

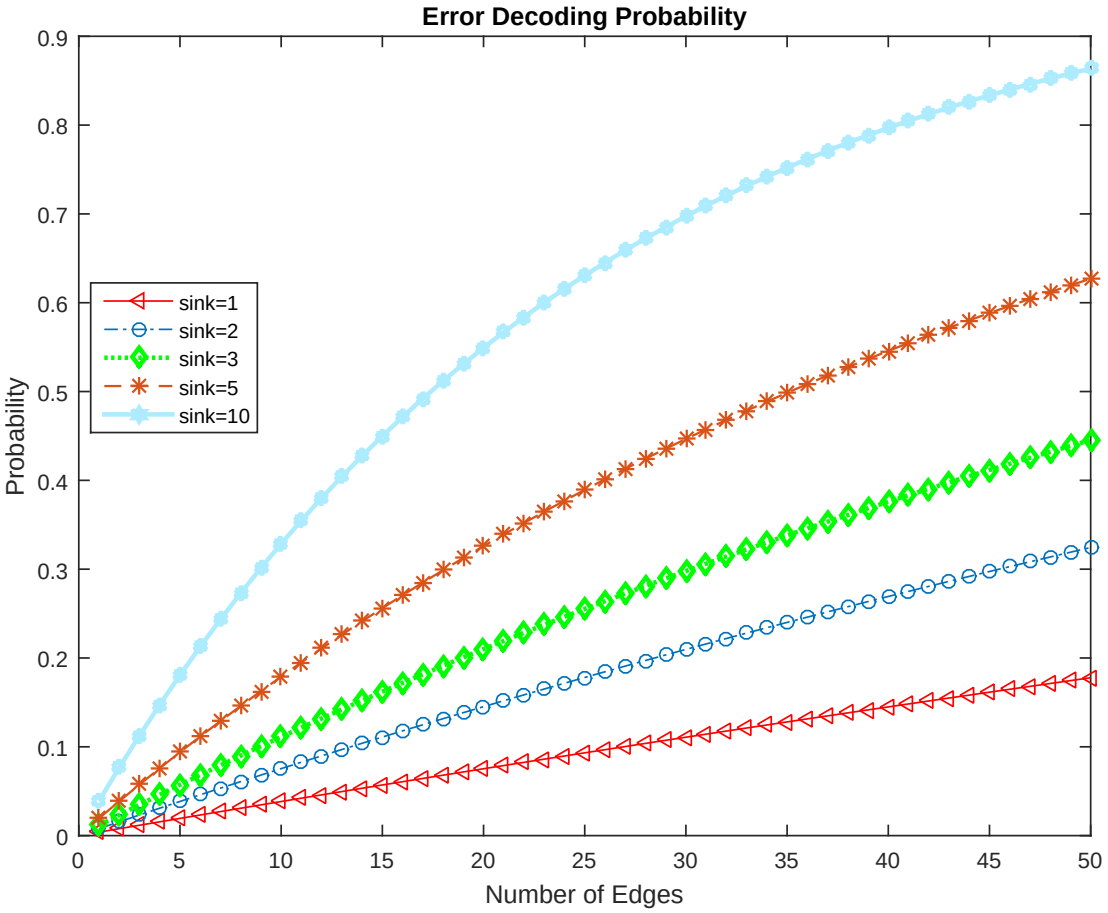

Figure 2.8: Error decoding probability for different number of destination nodes.

- **Finite field size:**

  The other representation of this probability is shown in Figure 2.9, where we show the effect of the error decoding probability based on different finite field size. Our simulation results show that, RLNC schemes which use a small finite field size (i.e., $\mathbb{F}_{2^3}$), has a larger error decoding probability in compared to the RLNC schemes who use a larger finite field size (i.e., $\mathbb{F}_{2^7}$ and $\mathbb{F}_{2^8}$).

- **Number of native packets:**

  In other simulation illustration, we show the direct effectiveness of the error decoding probability and the number of native packets. In this regard, in MATLAB, we generate 10 times, which is called sample id, several random numbers (i.e., native data packets) from different finite field size. In Figures 2.10, it is shown that by increasing the number of data native packets, a RLNC scheme can decode the coded packets with a lower probability, where different finite field size such as $\mathbb{F}_2$, $\mathbb{F}_{2^7}$, and $\mathbb{F}_{2^8}$ are applied respectively.

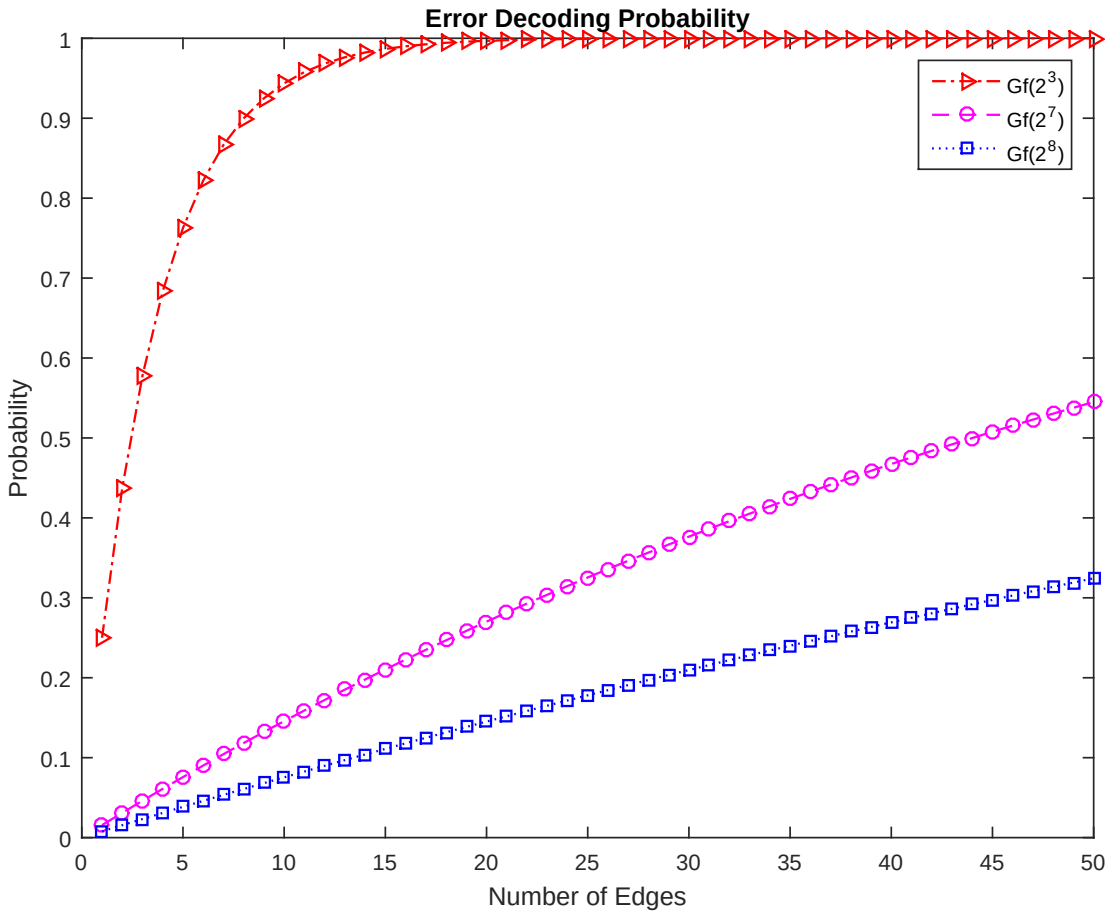

Figure 2.9: Error decoding probability for different finite fields.

Based on Figure 2.10, the successful decoding probability is very low in case of finite field $\mathbb{F}_2$. This is obvious which is related to the number innovative packets arrive at the sink nodes.

A much better successful decoding probability is shown in Figures 2.11 and 2.12. Based on these Figures, the successful decoding probability is almost same in the different finite fields ($\mathbb{F}_{2^7}$ and $\mathbb{F}_{2^8}$)

As a result, it is shown that successful decoding probability has a direct relation with the number of packets per each generation in RLNC schemes. It seems, a finite field such as $\mathbb{F}_{2^7}$ or $\mathbb{F}_{2^8}$ has a better successful decoding probability compared to $\mathbb{F}_2$. More precisely, by increasing the number of data packets, the finite field size should be large enough to enable a sink node to retrieve the packets successfully.

- **Finite field size:**

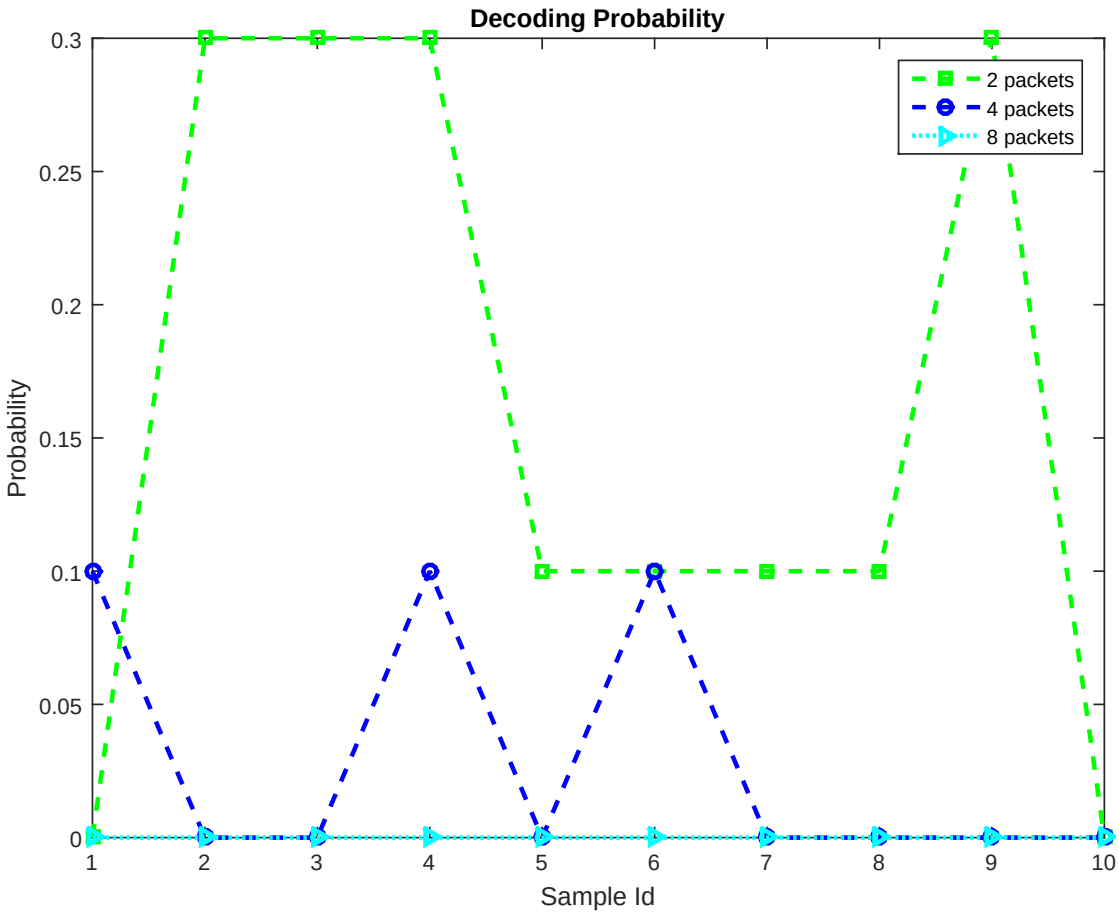

Figure 2.10: Successful decoding probability for different number of packets when the finite field size is $\mathbb{F}_2$.

In this section, we demonstrate three Figures 2.13, 2.14, and 2.15 which show the successful decoding probability of the RLNC mechanism versus the finite filed size. Increasing p, the finite field size, leads to a higher complexity for encoding, recoding and decoding; however, the number of packets can also increase the decoding probability. More precisely, we consider three scenarios which different number of packets (i.e., 2, 4, and 8) are used in each of them. We aim to calculate the successful decoding probability with different number of finite fields. In the first scenario, we assume that the source node generates only 2 packets over three different number of finite fields (see Figure 2.13).

In the second and third scenarios, we consider that the source node wants to transfer 4 and 8 packets, respectively. As mentioned earlier, the number of packets can alter the performance of a RLNC scheme; however, in the last simulation results it is shown that the finite field size can also affect the performance of the RLNC scheme. In other words, it should be mentioned that the number of native packets in each generation and also a finite field size, which are chosen by the source node, are two important factors which can lead to an increase in the RLNC performance.

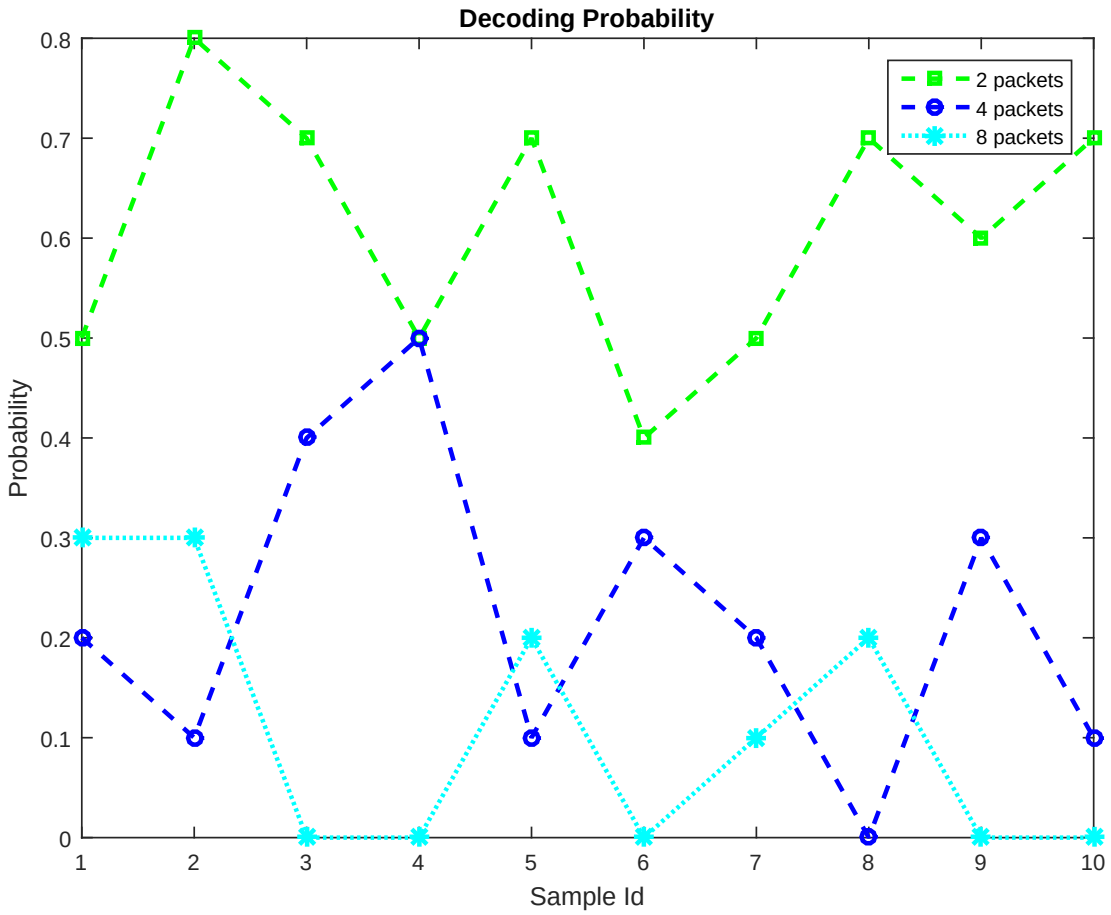

Figure 2.11: Successful decoding probability for different number of packets when the finite field size is $\mathbb{F}_{2^7}$.

More precisely, our simulation results show that the finite fields $\mathbb{F}_{2^7}$, and $\mathbb{F}_{2^8}$ can be used as practice, based on fact that they represent a better successful decoding probability (see Figures 2.14 and 2.15).

2. ***Time complexity***

To obtain a time required for decoding the data packets, we run our simulations on a 3.16GHz Intel Core 2 CPU, and 4 GB of RAM. In this thesis, a total time requirement is known as the whole time which is required from generating a packet by a source node, recoded by an intermediate node, and until finally decoded by a sink node. However, we also calculate three required times, which are used in our simulation in this thesis, such as encoded time, recoded time, and encoded time. The encoded time is considered as the required time for the source node to generate the data packets and encode them. A recoded time is considered as the time that an intermediate node needs to recode the data packets. Furthermore, a decoded time is used by the sink nodes to decoded the native data packets. For that reason, our simulation consists of two parts: (i) *total time requirement* and (ii) *encoded, recoded, and decoding time requirement*, where the

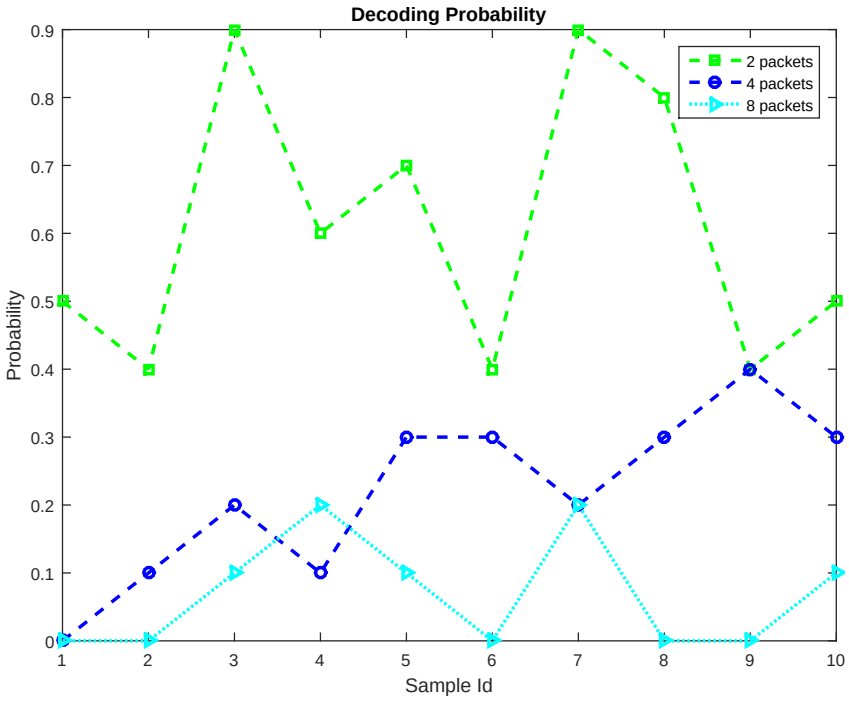

Figure 2.12: Successful decoding probability for different number of packets when the finite field size is $\mathbb{F}_{2^8}$.

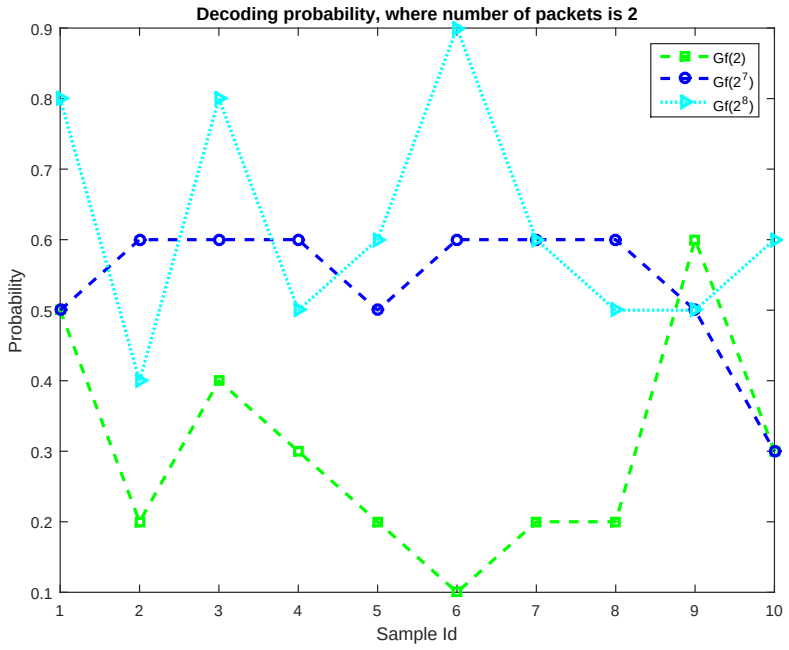

Figure 2.13: Successful decoding probability for different finite fields when the number of packets is 2.

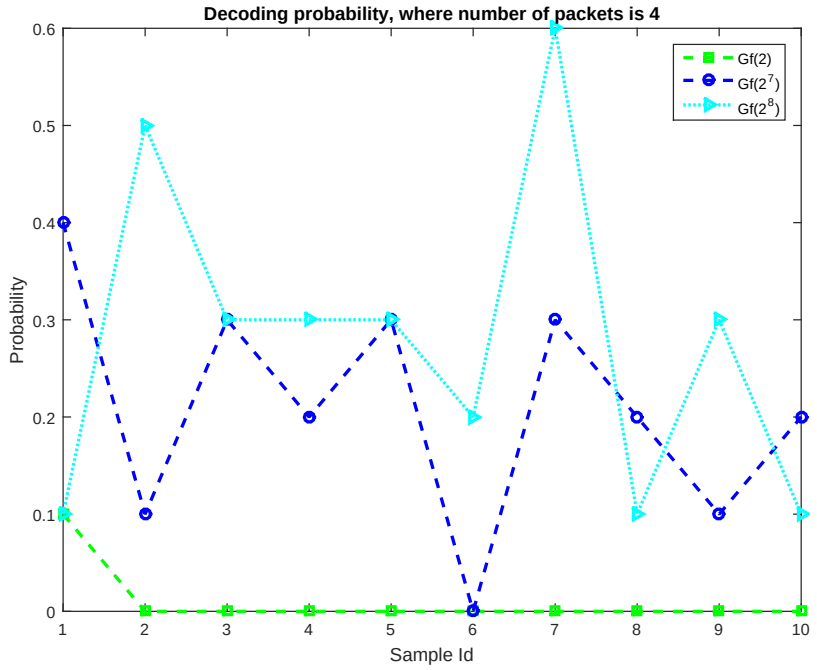

Figure 2.14: Successful decoding probability for different finite fields when the number of packets is 4.

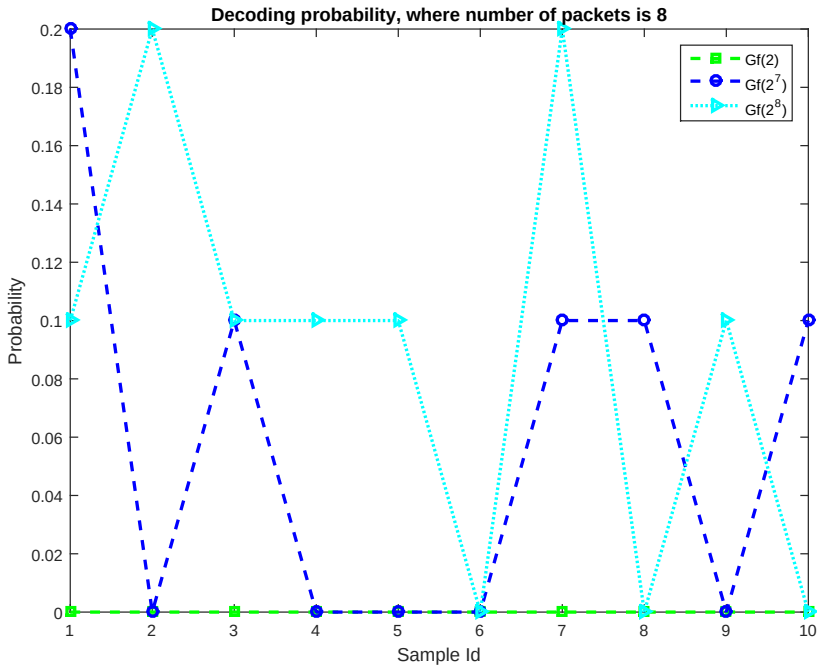

Figure 2.15: Successful decoding probability for different finite fields when the number of packets is 8.

following parameters are simulated respectively as follows:

(a) **Total time requirement:**

- **Finite field size:**

  In Figure 2.16, we illustrate the total time requirement where our simulation is run over a finite field $\mathbb{F}_{2^7}$. We also demonstrate time decoding requirement over a finite field $\mathbb{F}_{2^8}$ in Figure 2.17.
  Based on these Figures, it is completely clear that increasing the number of packets increases the time complexity.

- **Number of native packets:**

  In the following, we show the effect of the number of packets on the total time requirement. In this regard, Figures 2.18, 2.19, and 2.20 illustrate the time complexity when we run our simulation over three number of packets, 2, 4, and 8, respectively.

(b) **encoded, recoded, and decoding time requirement:**

- **Finite field size:**

  The encoded time, recoded time, and decoded time are illustrated in Figures 2.21, 2.22, and 2.23 respectively. In these simulations, we set the number of packets 4, and we show the results of different number of finite fields on each

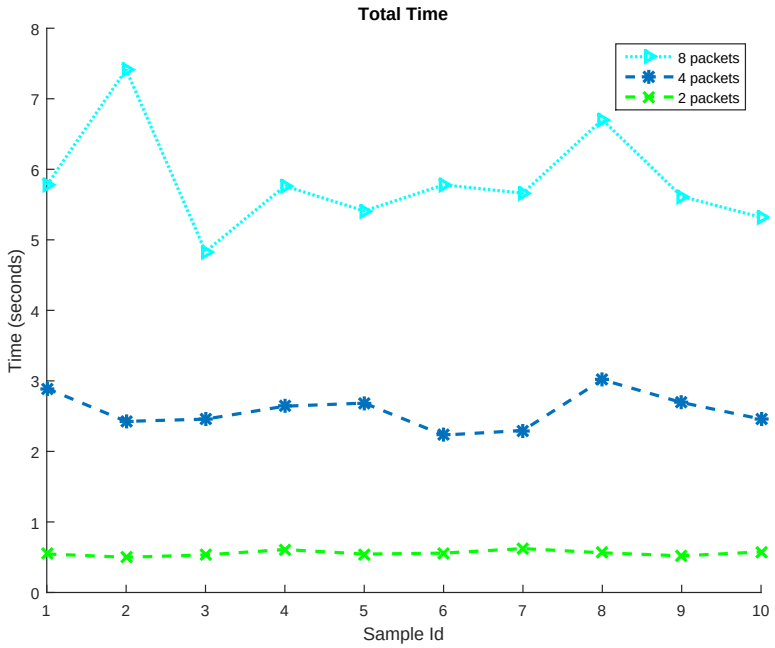

Figure 2.16: Required total time for different number of packets over $\mathbb{F}_{2^7}$.

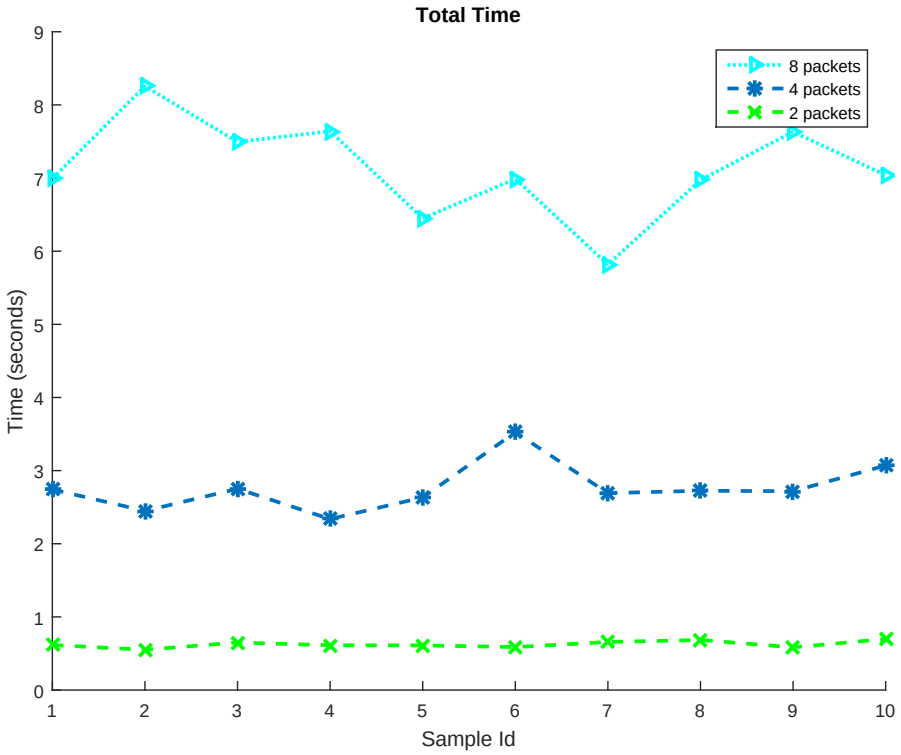

Figure 2.17: Required total time for different number of packets over $\mathbb{F}_{2^8}$.

encoded, recoded, and decoded time requirement.

- **Number of native packets:**

  To end this section, we show the effect of the number of packets on encoded, recoded, and decoded time requirement. In this regard, Figures 2.24, 2.25, and 2.26 illustrate the time complexity when we run our simulation over three number of packets, 2, 4, and 8, respectively.

From the complexity analysis of the RLNC scheme above, we come to the conclusion that it is necessary to know the number of packets and finite field size which are the two most important parameters that a source node should use to transmit its native data. In order to do that, based on the several practical network coding applications [36, 100], it seems that a RLNC scheme which uses a finite field $\mathbb{F}_{2^7}$ or $\mathbb{F}_{2^8}$ can benefit from advantages of network coding.

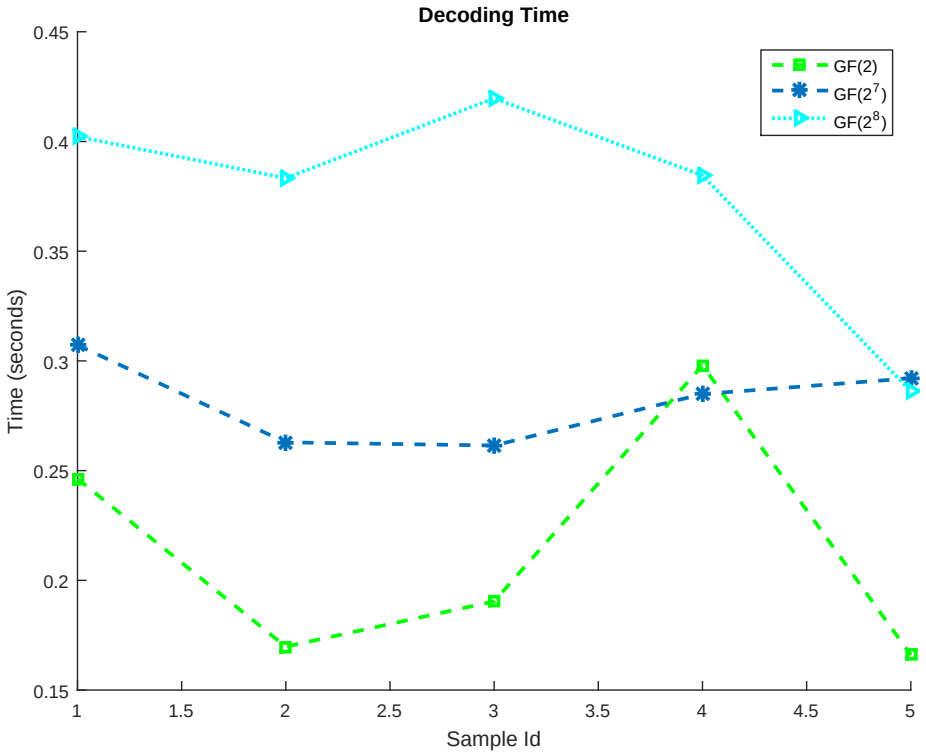

Figure 2.18: Required total time for 2 packets.

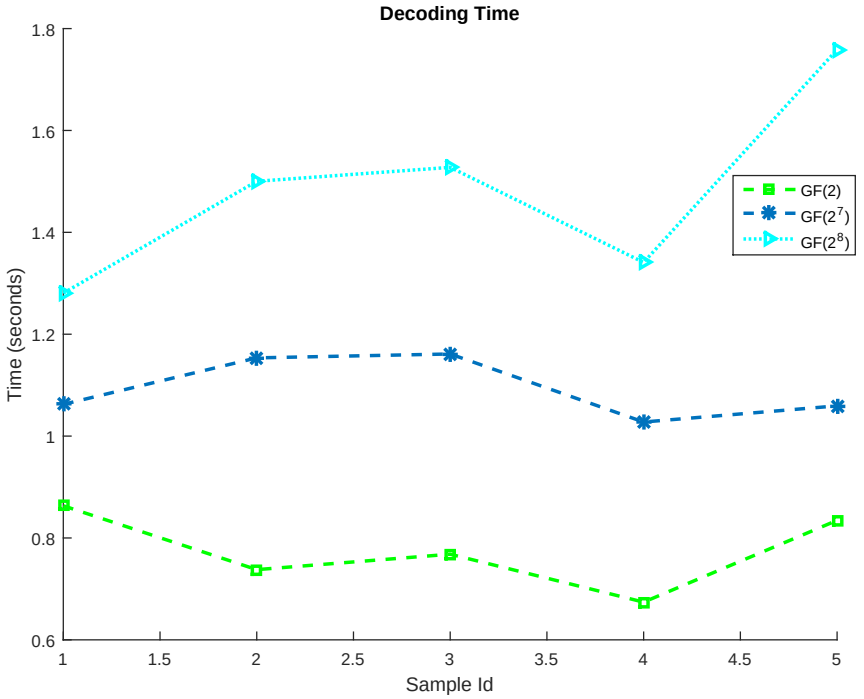

Figure 2.19: Required total time for 4 packets.

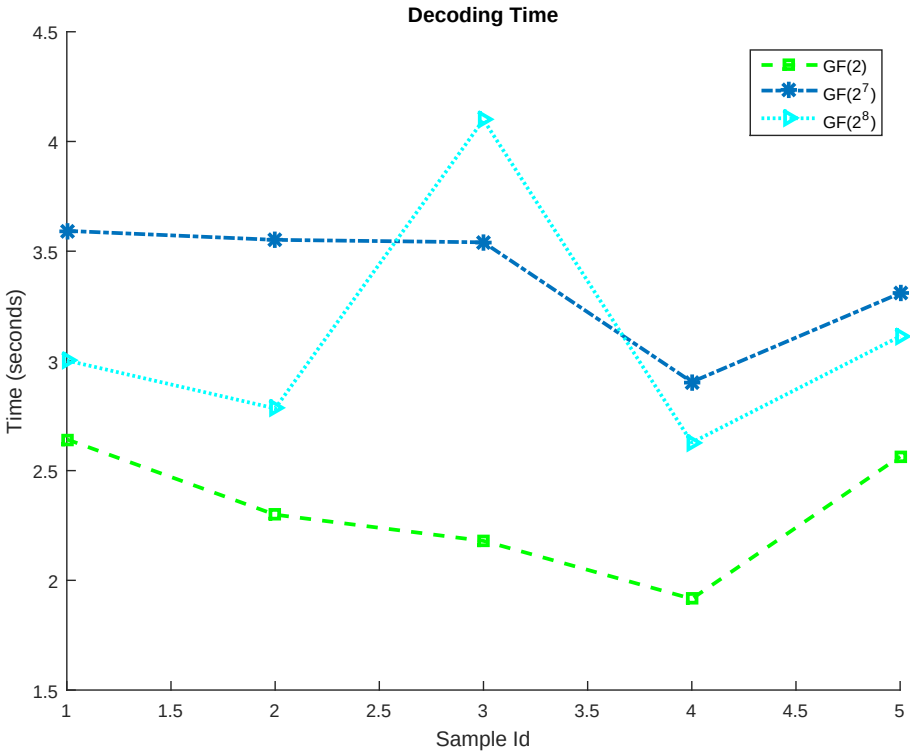

Figure 2.20: Required total time for 8 packets.

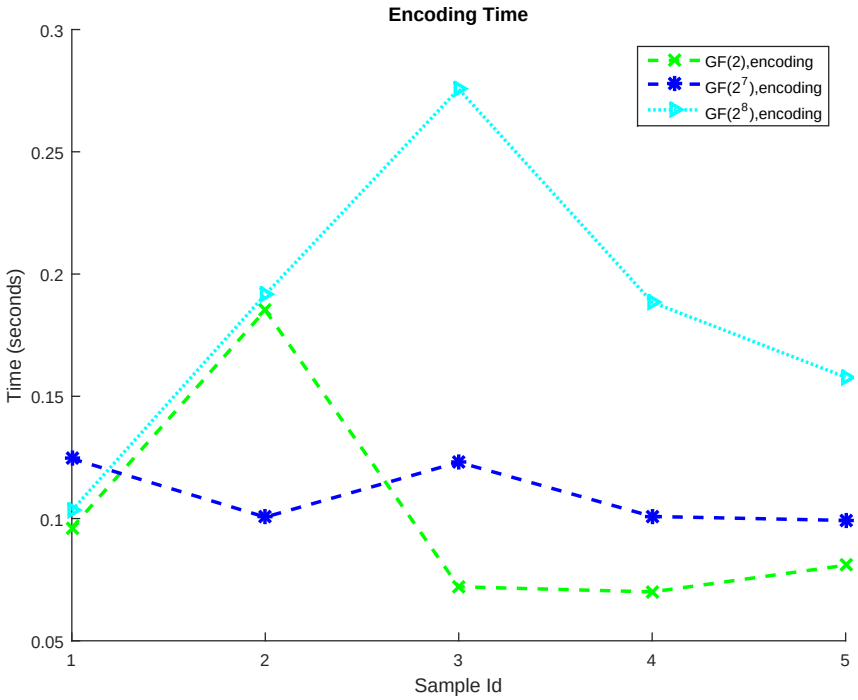

Figure 2.21: Required encoding time for 4 packets.

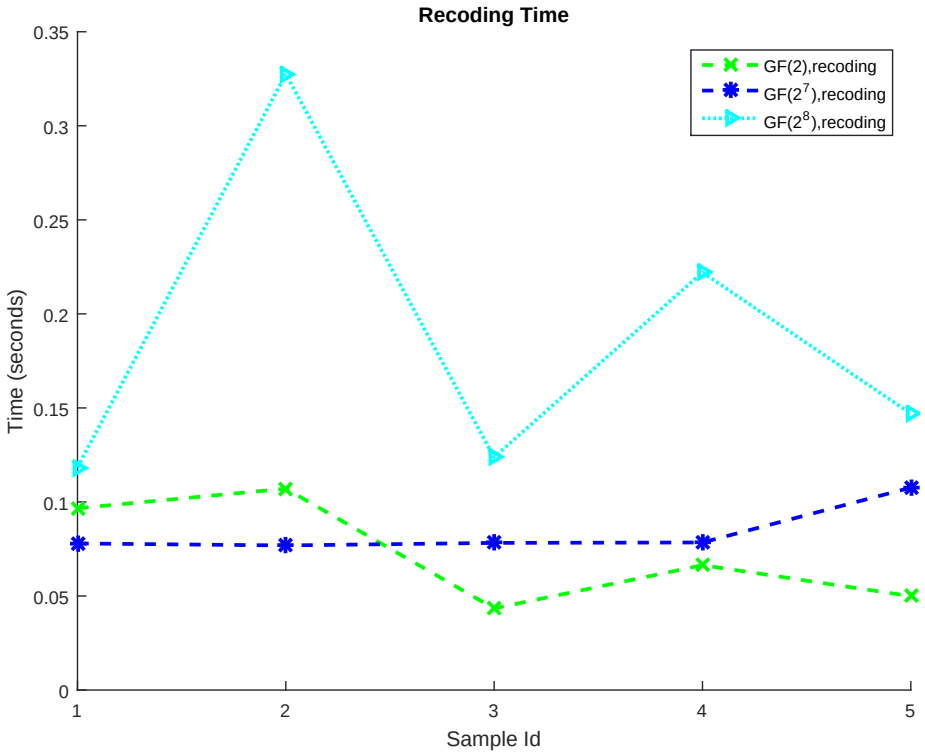

Figure 2.22: Required recoding time for 4 packets.

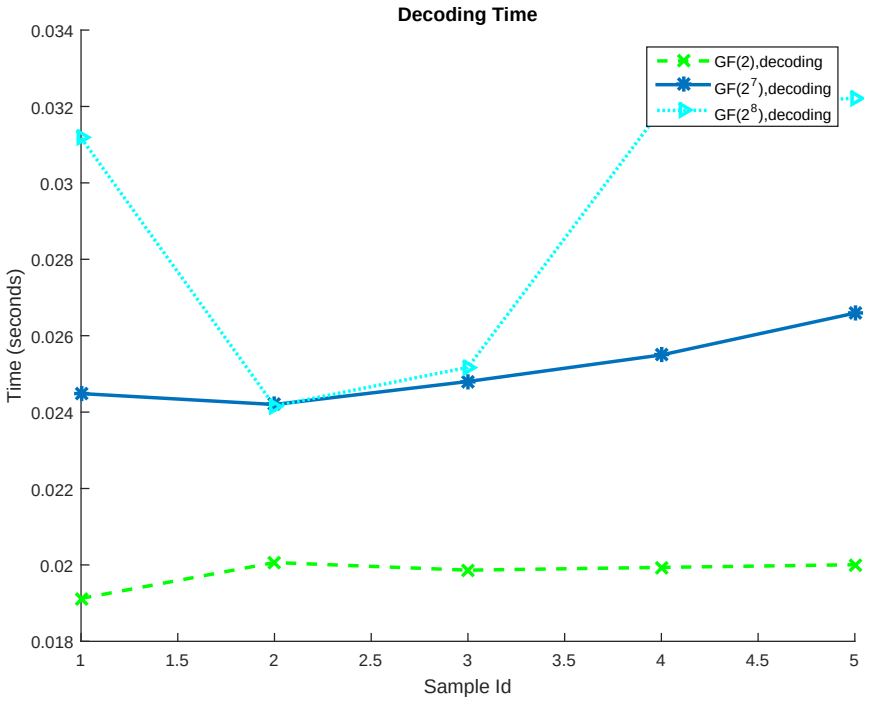

Figure 2.23: Required decoding time for 4 packets.

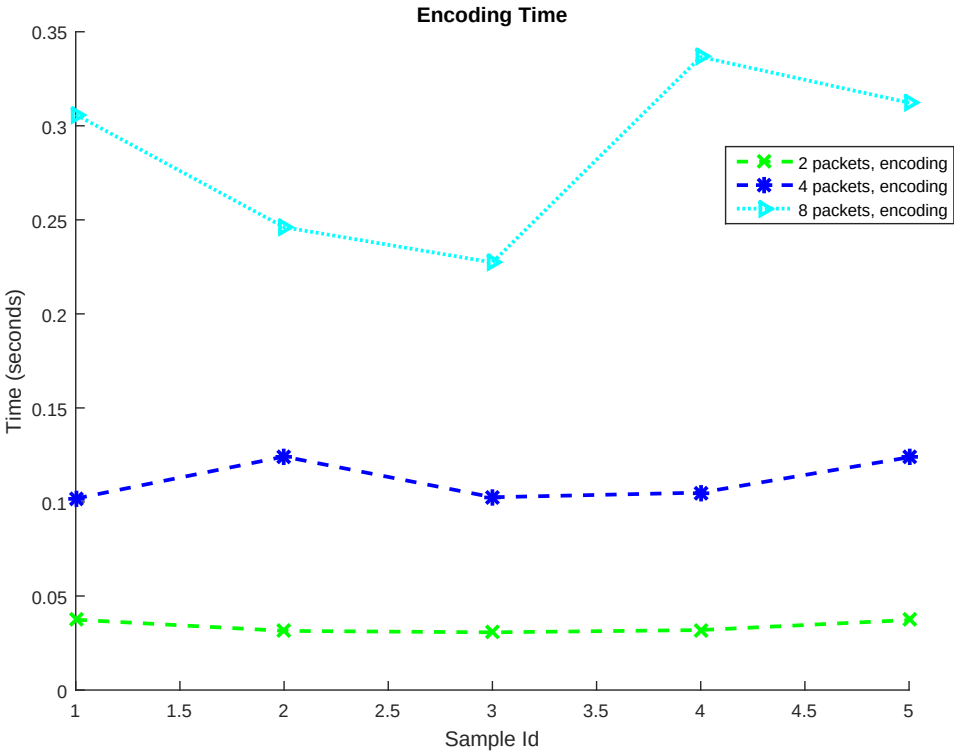

Figure 2.24: Required encoding time for different number of packets (2,4, and 8) .

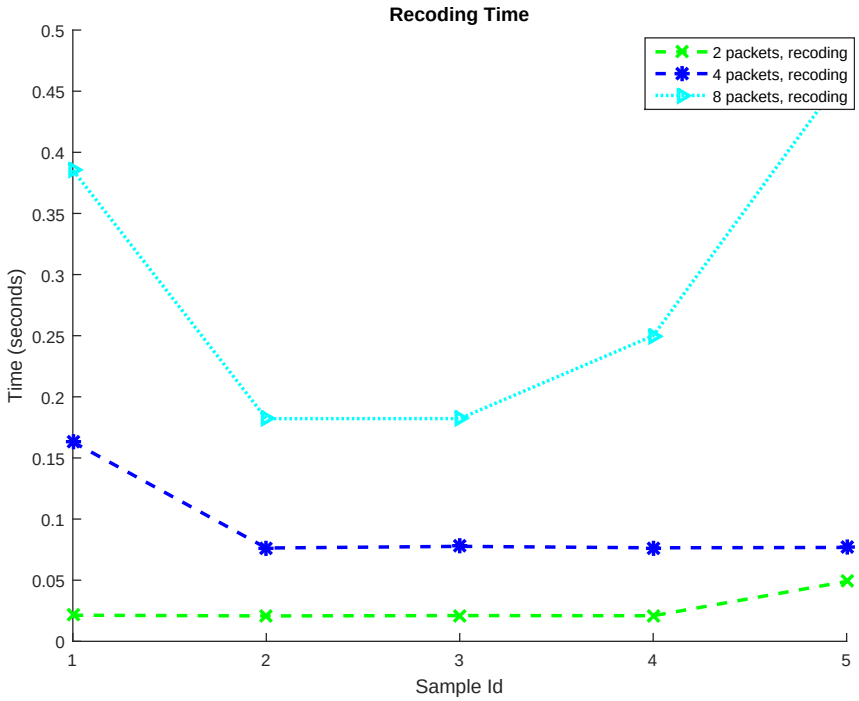

Figure 2.25: Required recoding time for different number of packets (2,4, and 8) .

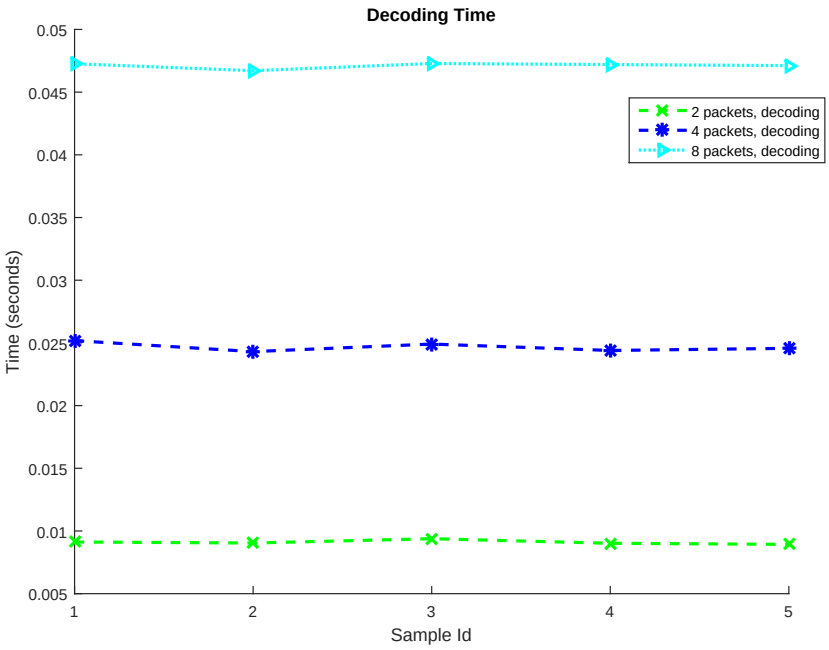

Figure 2.26: Required decoding time for different number of packets (2,4, and 8) .

2.3 Conclusion

In this chapter, we gave an overview of the main principle of cryptography and network coding. We also introduced the related works and existing schemes in the field of network coding. Then, we discuss the notation and assumption which we use in this thesis. The benefits and drawbacks of network coding were also discussed. Finally, we simulated the performance of a simple RLNC system. Our results have shown, that choosing parameters such as the number of packets in each generation and finite field size can influence many properties of the constructed RLNC, including but not limited to: throughput, delay, coefficients overhead, and decoding error probability.

Chapter 3

Secure Network Coding

> "In all secrets there is a kind of guilt, however beautiful or joyful they may be, or for what good end they may be set to serve. Secrecy means evasion, and evasion means a problem to the moral mind."
>
> Gilbert Parker

The Network Coding (NC) technique in Wireless Sensor Networks (WSNs), Wireless Networks (WNs), Smart Grid (SG), Cyber Physical Systems (CPSs) and Mobile ad hoc Networks (MANETs) can be a promising solution since it can provide significant benefits including improved network capacity and robustness. However, the Network Coding-enabled WNs are vulnerable to various types of attacks that can compromise the security of the network. Based on that and the fact that the security is critical factor for the success of wireless communications, we steer our focus on the mitigation of these types of attacks. In particular, we intend to provide a framework towards the deployment of the appropriate countermeasures against attacks targeting Network Coding-enabled WNs.

3.1 Introduction

We depend on Internet not only for our most business but also entertainments through WNs and it is proved that WNs provide us with infinite information about everything. So using WNs to access to the Internet and World Wide Web (WWW) have become a part of our daily life to find people, news, weather, sports, recipes. However, having a secure framework and assuring secure connection between devices and the Internet, or between devices become vital important. According to the basic concept of wireless networks and their behaviors, there are lots of security requirements and goals that should be considered in this kind of networks. Actually, the broadcast nature of wireless channel gives additional challenges for secure connection, because it is difficult to contain where that signal may travel. To address these challenges, the application of Network Coding technology in WNs can be a promising solution since it can provide significant benefits including improved network capacity and robustness among others. However, the Network Coding-enabled WNs are vulnerable to various types of attacks that can compromise the security of the networks that integrate them.

Based on that and the fact that the security is critical factor in wireless networks, our main contributions in this chapter are as follows:

- We review the security goals that we should consider in a wireless communications, and then we will introduce some of the most common threats and attacks that occur in a wireless network.
- We intent to provide a foundation for organizing research efforts towards the deployment of the appropriate countermeasures against attacks targeting Network Coding-enabled WNs.

Following the introduction, this chapter is organized as follows. In Section 3.2, a comprehensive review of the security requirements for NC-enabled WNs is provided. In Section 3.3, the main categories of attacks against NC-enabled WNs that can affect the security of the whole network are presented. In Section 3.4, countermeasures for attacks included into the defined categories in Section 3.3 are discussed. Finally, Section 3.5 concludes this chapter.

3.2 Security Requirements for Network Coding-Enabled Wireless Networks

NC-enabled WNs can be the target of many known and unknown security threats since many security breaches can raise in its environment due to its heterogeneous and dynamic nature. Hence, the following security properties are required to be assured in NC-enabled WNs: Authentication, Data Confidentiality, Data Integrity, Availability and Non-repudiation [105, 106].

- **Authentication:** Ensures that the origin of a packet delivered over a wireless communication channel is traceable. Thus, a receiving node is able to verify that the received message have indeed come from the legitimate sender node and not from a malicious actor. Furthermore, authentication guarantees that two nodes entering into a communication authenticate each other. In other words, authentication ensures that each communicating node is the one that it claims to be [29, 105, 107].
- **Data Confidentiality:** Prevents an unauthorized entity from accessing a message shared between entities authorized to access it. For example, this property allows only the legitimate nodes to access the content of the transmitted messages between them. Moreover, sensitive information located on the nodes such as private keys and identities must be protected from unauthorized access [105, 107].
- **Data Integrity:** Data integrity methods attempt at preventing modification of communicated/stored data by an unauthorized entity. Integrity is compromised when a malicious actor accidentally or purposely modifies or deletes data [105, 107].
- **Availability:** Ensures that any resource of a given NC-enabled WN is always available for any legitimate entity. Availability guarantees that the provided services of the NC-enabled WN are always available even in presence of internal or external attacks [105, 107].
- **Non-repudiation:** Prevents an entity from denying previous activities. For example, after completing the process of sending and receiving data, neither the sender later can deny that he sent the information nor the receiver can deny the reception [108].

3.3 Security Attacks for Network Coding-Enabled WNs

Taking into consideration the broadcast nature of the wireless channel and the security vulnerabilities of the NC technique, the satisfaction of the above-mentioned security requirements in NC-enabled WNs is a challenging task. A known classification of security attacks, which is defined both in X.800 and RFC 2828, includes passive attacks and active attacks [54].

Passive attacks only observe the communication without interfering in any way; in the other words they 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. As they do not compromise the operation of the network, detection of this type of attacks is difficult. On the other side, active attacks try to disrupt the normal network operation and may also alter, corrupt, or delete the data packets being exchanged over the network. Furthermore, the most active attackers can threaten different network layers (i.e., MAC layer, network layer, transport layer and application layer). Attacks targeting the NC-enabled WNs in different network layers are presented in Figure 3.1.

Some of the most destructive network attacks such as pollution attacks, Byzantine fabrication and modification attacks are caused by active attackers inside the network. Despite its benefits, NC-enabled networks are more susceptible to attacks than traditional store-and-forward one, since even a single number of corrupted packets can infect a large number of downstream nodes because the corrupted packets propagate via recoding. We consider two types of attackers; internal and external attackers. As internal attackers, we consider the nodes that have been compromised or the legitimate nodes that have been turned into malicious ones. Moreover, they have access to network resources and key material. However, the external attackers are limited to adversaries without having access to network resources; this includes eavesdropping, modification and replay of packets. However, our main purpose in this chapter is to study the common attacks which are possible for NC-enabled WNs. Specifically, our main idea here is to show how the specific characteristics of NC can be exploited by attackers.

3.3.1 Eavesdropping

The main goal of an eavesdropper is to read data traffic and obtain sensitive information (e.g., native packets, public keys, private keys, location, or passwords of other nodes). In this case, the attacker not only acts as a normal node but also tries to attain some sensitive information that should be kept secret during the communication (see Figure 3.2). In order to obtain this information, the eavesdropper node only listens to the message transmission in the broadcasting wireless medium [109]. Eavesdropper targets at degrading the confidentiality of the transmitted data.

3.3.2 Impersonation

This attack is a kind of active eavesdropping, where the attacker nodes in order to obtain the information send queries to the victim nodes by concealing themselves as friendly nodes. By this bogus authenticity, the attacker can introduce conflicting routes or routing loops, cause waste of nodes' power and portion the network (see Figure 3.3). State-aware NC protocols rely on network nodes which can be effected by this type of attack [55]. Impersonation targets to degrade the authenticity property in NC-enabled WNs.

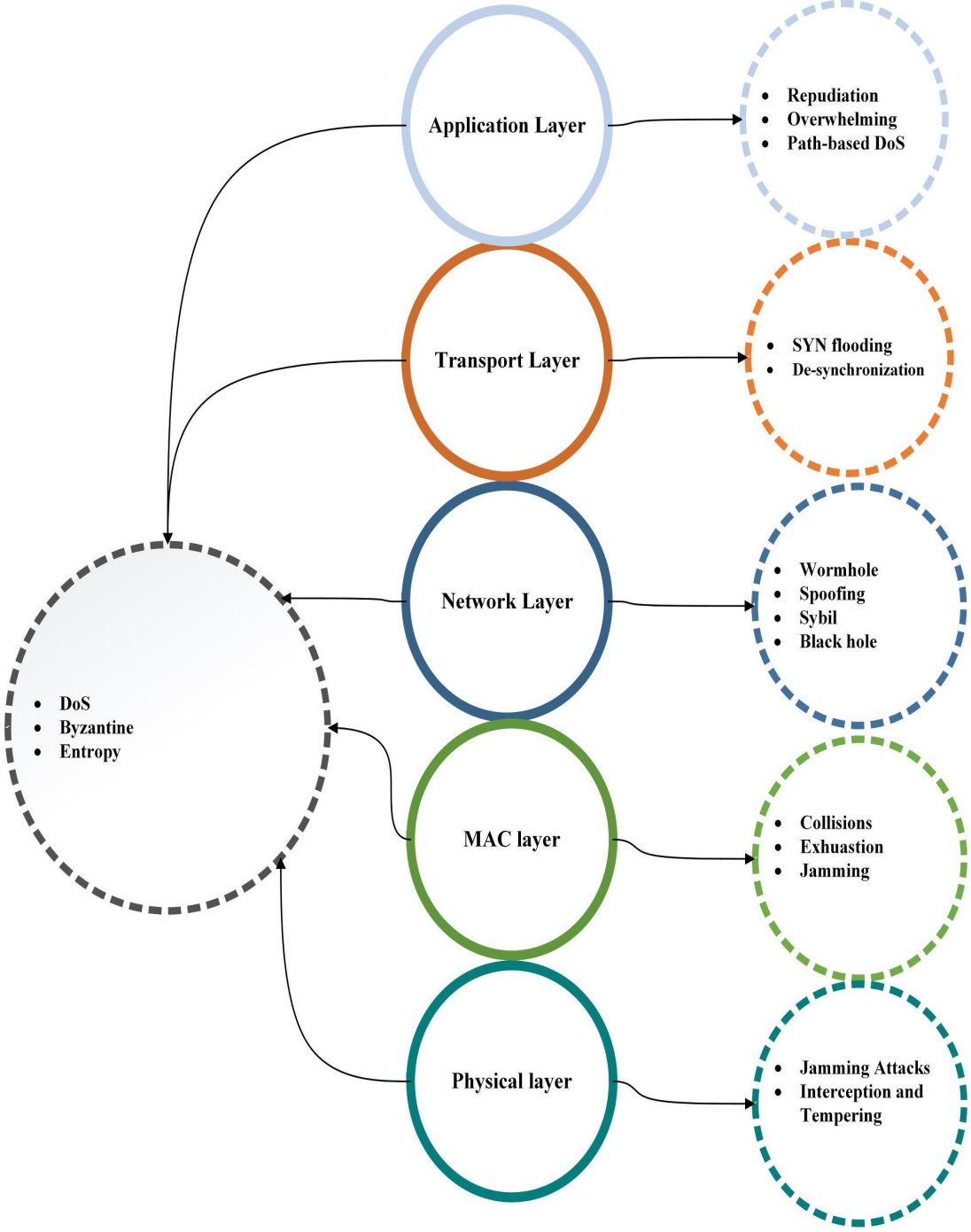

Figure 3.1: Attacks against NC-enabled WNs on different network layers. At the right side, we show the specific attacks for each layer, and at the left side, we show the common attacks targeting all the network layers.

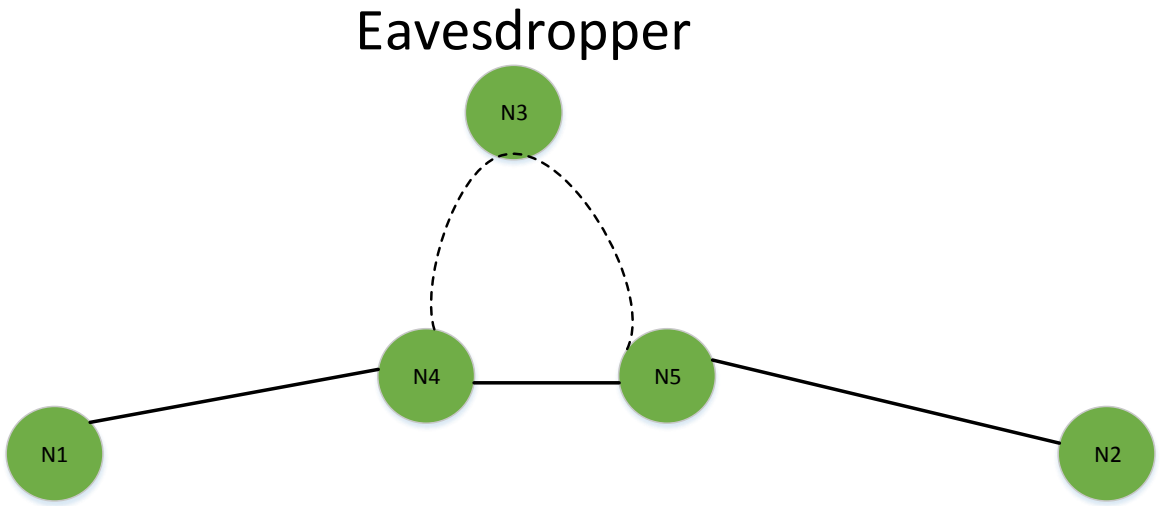

Figure 3.2: A simple eavesdropping attack behavior, where he tries to listen to the data which are available at the channel.

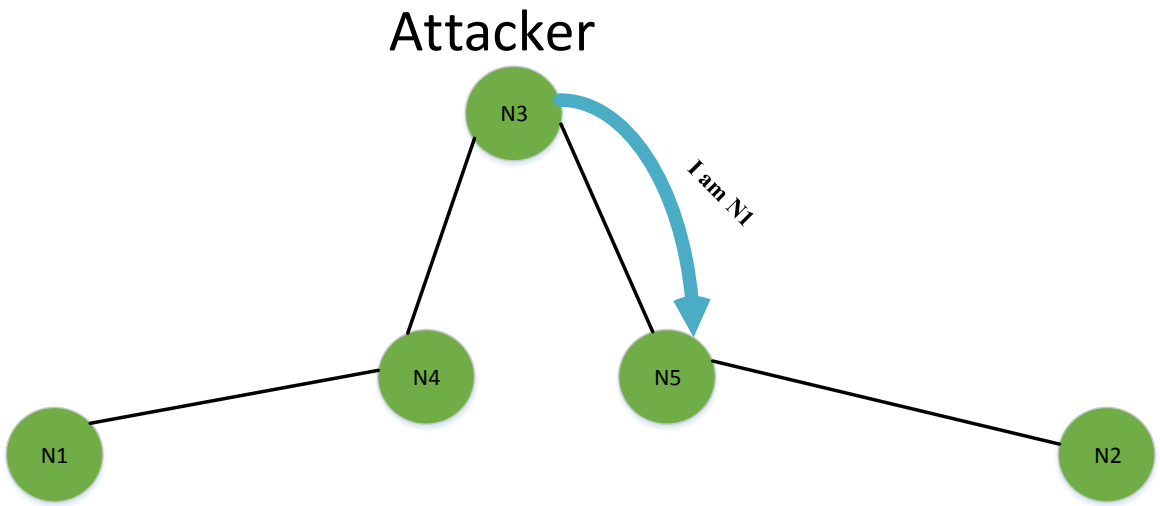

Figure 3.3: A simple impersonation attack behavior.

3.3.3 Byzantine Attacks

A Byzantine attack is an attempt to make the resources of a system unavailable to the legitimate users. More specifically, an adversary aims to prevent some services from functioning efficiently. Hardware failures, software bugs or resource exhaustion can lead to Denial of Service (DoS) attacks. Actually, the adversary in DoS attacks targets availability, which

ensures that an authorize node can access all the services of the network. In this attack, the malicious node may inject a high volume of junk packets into the network and this leads to waste of network resources like nodes' battery power, memory, and wired or wireless channels. Various types of Byzantine attacks work at different layers and affect differently the network such as jamming and tampering at the physical layer, collision and exhaustion at the link layer, black holes and routing table overflow at the network layer, SYN flooding and de-synchronization at the transport layer, and finally failure in the web servers at the application layer [110].

Byzantine attacks include a wide range of attacks such as selectively dropping packets, forwarding data packets through non-optimal or even invalid routes, generating routing loops, modifying or altering the message in transit, changing the header of packets, flooding its neighbors via injecting lots of corrupted blocks (pollution attack) or even clean but old and reparative packets (Byzantine replay attack) into the downstream victim nodes, and so on. The main objective of these attacks is to decrease the network throughput. In state-aware NC protocols, packet headers normally contain topology states and routing information, and in stateless NC protocols, headers normally contain the required decoding vectors [22]. Therefore, the Byzantine attacks are not only disruptive on state-aware NC protocols but also impose a negative impact on stateless NC. In NC-enabled WNs, the main categories of Byzantine attacks that may occur are the following:

3.3.3.1 Byzantine Fabrication (Pollution Attack)

Based on the nature of packet mixing in NC schemes, these schemes are vulnerable to a severe security threat known also as pollution attacks, where an adversary node injects corrupted packets into the network. So, by propagating a corrupted packet that an intermediate node uses during the coding process, all the packets that are coded and forwarded by the node will be corrupted as well. Moreover, Byzantine fabrication attacks 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. 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. We present a pollution attack in Figure 3.4. Pollution attack targets to compromise the data integrity property in NC-enabled WNs.

3.3.3.2 Byzantine Modification

In this type of attack, the adversary aims at modification of the transmitted data in transit. For instance, the attacker can send a lot of unnecessary messages to a victim node by changing the destination node native messages. More precisely, the adversary causes a destination node decodes the native packets incorrectly by changing the coded packets in transit in stateless NC protocols. State-aware NC protocols can be affected by changes on either the coded data packets [55].

3.3.3.3 Byzantine Replay Attack

A node sends the old authenticated previously transmitted messages to the network. Sending these old messages cause waste network resources, and eventually degrade the overall

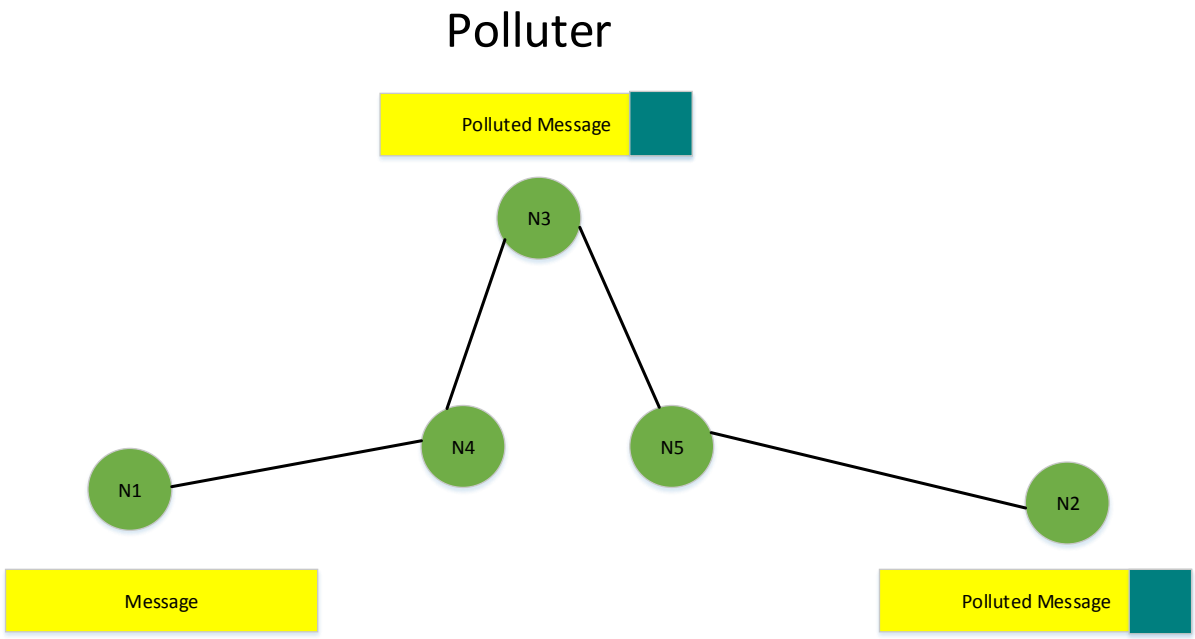

Figure 3.4: A Simple Byzantine Fabrication Attack.

throughput rate [111]. Stateless NC protocols can be affected in terms of NC gain and processing time by the injection of packets which are repeated into the information flow. State-aware NC protocols can be affected by sending the old messages to the specific victims in order to waste its resource as well. An example of a Byzantine replay attack is demonstrated in Figure 3.5.

3.3.4 Wormhole

Wormholes can constitute a serious threat in WNs, where two or more malicious nodes can collaborate and create a tunnel between two nodes. Afterwards, these malicious nodes record packets and retransmit them into the network through the tunnel [112]. The adversaries target at convincing the neighbor nodes that the two side of tunnel are in the same range and very close to each other. By using this strategy, an adversary may be able to send the messages to the other part of the tunnel in the network without being in the same range. The wormhole attack can impose more severe impact on the state-aware NC protocols, such as disrupting the route discovery process and taking the control of flows by leading them through the tunnel, in comparison with the impact on the stateless protocols. Figure 3.6 depicts the wormhole attack.

3.3.5 Entropy Attack

An adversary intends to generate valid but non-innovative packets that are valid linear combination which are stored or overheard at an earlier time by the adversary. In other words, the adversary node creates and distributes a non-innovative coded packet which is

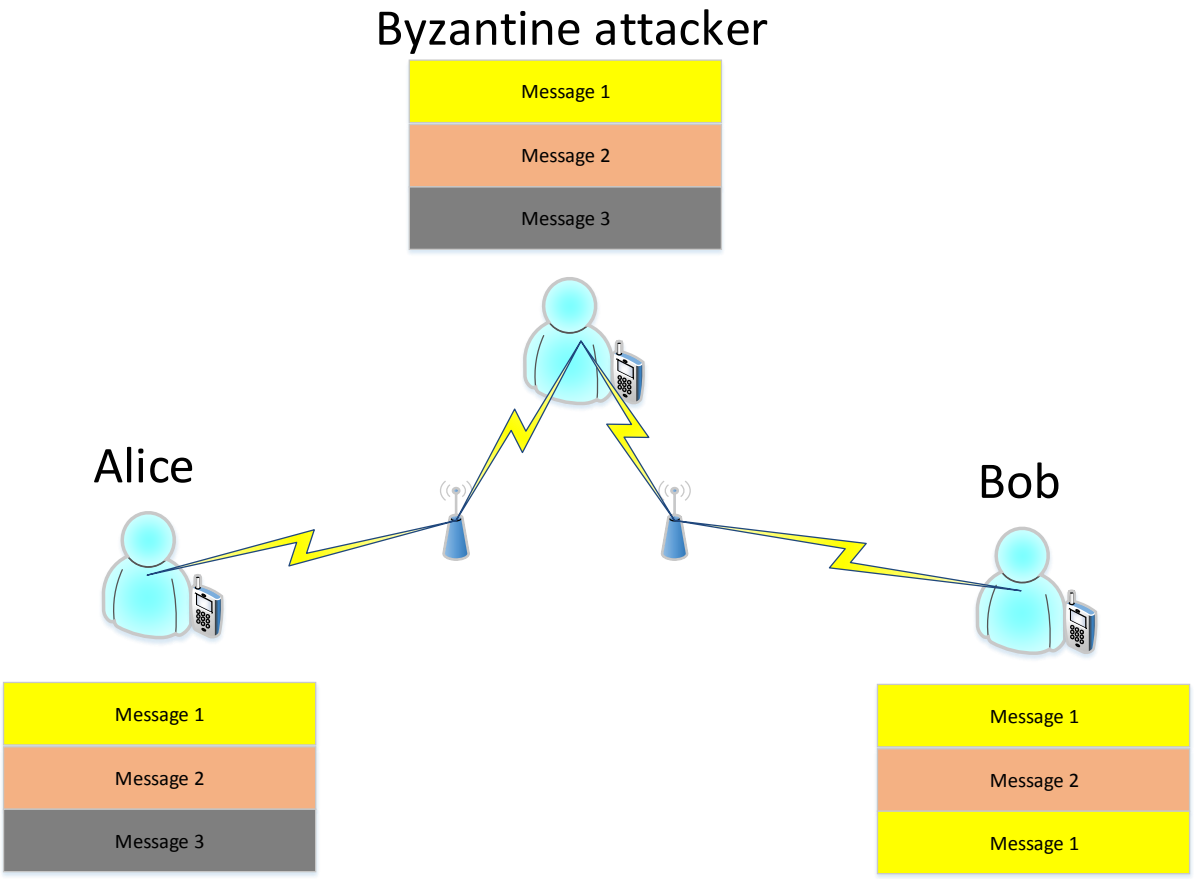

Figure 3.5: Byzantine Replay attack.

a non-random linear combination of coded packets such that the generated coded packet is linearly dependent with the previous coded packets stored at a downstream node. These valid but non-innovative packets decrease the decoding opportunities at sinks in both stateless and state-aware NC systems, cause waste of network resources, and degrade the overall throughput rate. In a pollution attack, a node creates and generates a coded packet which is not valid linear combinations of the native data. However, entropy attackers create and distribute coded packets that are valid linear combinations of the native data. As a result, a pollution defense is not helpful against an entropy attack because all of the defenses against pollution attacks rely on the fact that an invalid coded packet is not a valid linear combination [111].

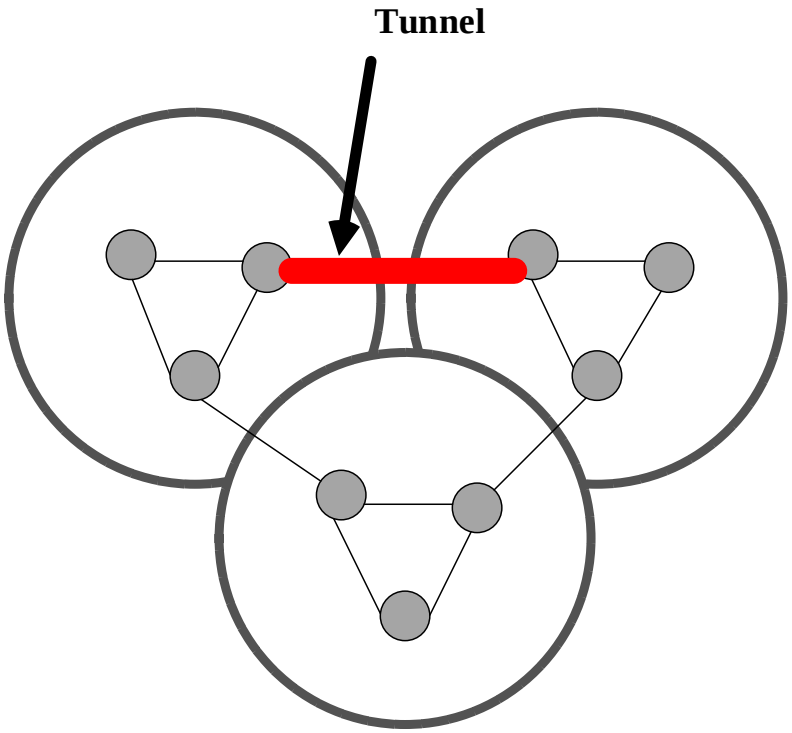

Figure 3.6: Wormhole attack.

3.4 Security Countermeasures for Network Coding-Enabled Wireless Networks

The fact that NC allows intermediate nodes to manipulate and mix incoming data packets from neighbor nodes as well as transmits data packets to neighbor nodes leads to open security issues in NC-enabled WNs. Therefore, the NC-enabled WNs are vulnerable to various types of attacks that can compromise the security of WNs. These types of attacks, which are presented in Section 3.3, are summarized into Table 3.1.

Table 3.1: Attacks against NC-enabled WNs

Type of Attack	**State-aware/Stateless NC**	**Target**
Eavesdropping	Both	Confidentiality
Impersonation	State-aware NC	Authenticity
Byzantine	Both	Integrity, Availability
Wormhole	Both	Confidentiality
Entropy	Both	Integrity

In this section, we particularly focus on the mitigation of eavesdropping, entropy, and Byzantine fabrication (pollution) attacks, which are among the most common attacks that

Table 3.2: Categories of the security schemes against Pollution, eavesdropping, and entropy attacks.

Type of Attack	Security Schemes		Related works
Byzantine Fabrication	Cryptography and Key Management	Homomorphic hash functions	[20, 113]
		Homomorphic digital signatures	[23, 24, 114]
		Homomorphic MAC	[25, 26, 115]
	Network Codes		[116–118]
	Cooperative Schemes		[20, 119–123]
Eavesdropping	Cryptography and Key Management		[124, 125]
	Network Codes		[14, 17]
	Cooperative Schemes		[55]
Entropy	Only a limited number of defense mechanisms - No any specific security scheme		[111, 126]

can be used against NC-enabled WNs and have a negative impact on the security of the whole networks. These attacks can result in the degradation of the performance gain of coded networks or even worse in the complete disruption of the whole network operation. However, these attacks have to be addressed in order for the NC-enabled WNs to reach their full potential [70]. Towards this direction, several security schemes, focused on NC-enabled WNs, have been recently proposed to handle these attacks since the traditional security mechanisms are infeasible solutions for the resource constrained NC-enabled WNs due to the large bandwidth for communication that they require. We classify the security schemes against Byzantine fabrication, eavesdropping, and entropy attacks into three main categories as follows: (i) cryptography and key management based schemes which are used by network coding as an extra line of defense through assigning proper keys or cryptographic functions; (ii) schemes that allow legitimate nodes to add redundancy to their information flows by applying an appropriate secure network codes algorithm; (iii) cooperative schemes which are able to detect compromised nodes and isolate them from the other legitimate nodes. This categorization of the security schemes against Byzantine fabrication, eavesdropping, and entropy attacks is also demonstrated in Table 3.2.

3.4.1 Countermeasures to Eavesdropping Attacks

3.4.1.1 Cryptography and Key Management based Schemes

The compromise of even a single node in WNs can reveal the secret keys and for that reason all its messages can be decrypted easily. Therefore, key distribution is a vital requirement of WNs. Moreover, the basic communication patterns of WNs differ from traditional networks' patterns since key-distribution techniques need to scale to networks with hundreds or thousands of nodes. A simple approach for distributing the keys in WNs is to pre-configure the network with a shared unique symmetric key for each node. However, we need to establish a lot of keys and distribute them to all nodes of the network. On the other hand, public-key cryptography (e.g., Diffie-Hellman key scheme) is another option which is useful in wireless networks. Besides, bootstrapping keys which are based on a trusted base station is another solution for key distribution [124]. In this case, all the processes related to the establishment and distribution of the keys are done by the base station and each node only needs to share a key with the

base station. However, this mechanism makes the base station a single point of failure and for that reason this security mechanism is considered as weak one.

3.4.1.2 Security via Network Codes

By exploiting appropriate network codes, it is possible to prevent eavesdropper from obtaining meaningful information from the eavesdropped data. Based on the use of appropriate network codes, there is the opportunity to mitigate a significant devastating effect of eavesdropping attack in network coding-enabled WNs. By specific definition of an eavesdropper model, which is described in [17], in network coding-enabled WNs there are the following three different security levels for network coding systems: Shannon security level, weak security level and computational security level. Preparing a network coding system, which supports the Shannon security level, is not practical as it is feasible only under ideal assumption. Moreover, for the weak security level, the authors show that it requires a sufficiently large filed size. Finally, by utilizing cryptographic approaches, the computational security can be achieved in practical NC-enabled WNs.

3.4.1.3 Security via Cooperative Mechanisms

In case of eavesdropping attacks, the authors in [55] studied a cooperative scheme with special emphasis on three different nodes in the NC-enabled WNs. First, they consider nice but curious nodes, which they do not ignore the data for which they are not the intended receivers. In the second type, the eavesdropper is able to wiretap a subset of network links. Finally, the third type of attacker is a worst-case eavesdropper who is given full access to all the traffic in the network

3.4.2 Countermeasures to Entropy Attacks

In contrast to the other treats and attacks which have been mentioned above, the entropy attacks have been emerged recently. Hence, only a limited number of defense mechanisms have been proposed for entropy attacks. In the context of entropy attacks, an attacker creates packets that contain information, which has been already known by the other sensors, and send them to the next nodes. The result is a severe degradation of the system performance.
An efficient symmetric-key based authentication scheme with network coding, called PMAC was proposed by Cheng et al. in [126]. In particular, their approach requires that any node can construct a valid MAC (tag) for any legitimate block it wants to upload. This idea can resist against entropy attacks.
The main research in this topic was done by [111], where the authors consider two local and global entropy attacks. The attack that produces coded packets that are non-innovative to local neighbor nodes is called local entropy attack. A main goal of attacker in global entropy attacks is that producing coded packets which seems innovative to local neighbor nodes but they are non-innovative to at least an outlying downstream node. The simulations in this work show the impact of their solution to the entropy attacks. Moreover, they present the difficulties in defending against a global entropy attack. However, developing an extremely efficient countermeasure mechanism against entropy attacks still is required and it is an ongoing research.

3.4.3 Countermeasures to Byzantine Fabrication (pollution) Attacks

3.4.3.1 Cryptography and Key Management based Schemes

Due to the main principle of NC, according to which the intermediate nodes employ coding operations to mix (i.e. recode) the incoming packets, the NC-enabled networks are vulnerable to a severe security threat known as pollution attack. According to this kind of attack an adversary injects into the NC-enabled network polluted packets that propagate quickly into other legitimate packets and corrupt bunches of them because of the re-coding process that take place at the intermediate nodes of the NC-enabled network. Consequently, pollution attacks lead to network resource waste. In this sense, detection of pollution attacks is of utmost importance for NC-enabled networks. To address pollution attacks, a number of cryptographic based schemes have been proposed in [19, 21, 24, 111, 127–130]. Specifically, the proposed cryptographic schemes include Homomorphic Hash Function (HHF) [20, 113] , Homomorphic digital signatures [23, 24], and Homomorphic Message Authentication Code (HMAC) schemes [25, 26, 115] for NC-enabled systems. However, these solutions have several drawbacks, when they are applied to NC-enabled WNs, due to the fact that they are computationally expensive and they cannot be supported by the resource constrained NC-enabled WNs. Moreover, trusted party schemes and Public-Key Infrastructures (PKIs) are used as solutions for ensuring security in NC-enabled WNs. In addition, the key pre-distribution schemes emerge as appropriate schemes for NC-enabled WNs that can afford less computation processes. Furthermore, according to [55] it is possible to design a secret key distribution scheme that requires only a small number of pre-stored keys by exploiting the benefits of network coding. Nevertheless, the proposed scheme ensures that shared-key connectivity is established with probability one and that the mobile node is provably oblivious to the distributed keys. Pollution attacks can be considered as two types: data pollution attacks and tag pollution attacks. These types of pollution attacks are further discussed as follows:

1. **Data Pollution:**

   In data pollution attacks the adversary targets at modifying (i.e. corrupting) the content of the transmitted packet. There are two main types of security schemes against data pollution attacks: cryptographic schemes ([19]- [2]) and information-theoretic schemes ([21], [22]).

   Cryptographic schemes: These schemes assume that the adversaries are confined in terms of computation process. The sources append additional verification information to the end of the coded packets. This allows intermediate nodes to verify the coded packets and filter out the polluted ones. This type of schemes includes the *homomorphic hashing schemes*, the *homomorphic signature schemes* and the *homomorphic MAC-based schemes.*

   - *homomorphic hashing schemes* are based on a homomorphic hash function which is applied by the source node to the packets. One of the well-known homomorphic hashing scheme is proposed in [19] by Krohn et. al. This scheme enables intermediate nodes to verify on-the-fly the transmitted packets. For that reason, they use homomorphism to compose multiple blocks inputs into a single value. Additionally, by supporting cooperation between legitimate nodes, authors in [20] extended the

previous work (i.e., [19]) and further reduced the expensive computation cost of hash functions. However, this scheme requires additional secure communication channels in order to transmit the calculated hash values to the intermediate nodes.

- *homomorphic signature schemes* appeared by Charles *et. al* in [131] for the first time. According to the proposed scheme in [131], the nodes are enabled to sign any linear combination of their incoming packets and produce a corresponding signature for each linear combination. Then, the homomorphic property of the signatures allows the next nodes to check the integrity of the content of the native packets. Another signature-based scheme was proposed by Yu et al. [24]. This idea is a kind of homomorphic signature function which resists against pollution attacks. In this scheme, the source signs its messages using its private key, and the forwarders verify the messages by using the public key. The authors believe that a hash collision is equivalent to solve a Discrete Logarithm Problem (DLP). In the proposed scheme in [24], any extra secure communication channel is not required. Moreover, the experimental results show that its verification efficiency is improved up to 10 times compared to the other related works (i.e. CJLs scheme [131]). Finally, the authors in [2] proposed a homomorphic signature scheme for both linear and XOR NC to prevent pollution attacks.

  The authors in [114] also present two signature schemes for network coding to prevent pollution attacks. The proof of their first scheme relies on the co-computational Diffie-Hellman problem which is slightly stronger in comparing to the Diffie-Hellman problem. Both two approaches are based on a single public key that can be used to sign many linear spaces.

  Zhao et al. [23] designed a signature scheme to maintain network coding properties for file distribution. In this scheme, the source calculates a signature from the spanned subspace and broadcasts this signature to all nodes. They use a standard public-key method and the verification is accomplished by checking the membership of encoded block in the spanned subspace. The authors proved that breaking this scheme is as hard as solving a Diffie–Hellman Problem (DHP).

  An efficient scheme for securing XOR network coding was presented for the first time in [2]. The authors studied the integrity of XOR coded-packets by proposing a messages authentication code scheme; however this scheme is vulnerable to tag pollution attacks. In particular, they assume that each node picks a fixed number of keys randomly from a large global key pool and a public key. Through meticulously managing the key pool size and the number of keys which each node picks, the authors assure that any two nodes have certain probability to find some shared keys. The source uses its keys to generate message authentication codes (tags) for its messages. Since the forwarders have the shared keys and thus they verify the MACs of received messages by using their shared keys.

- *homomorphic MACs schemes* rely on message authenticated information which is added at the end of the native packets. Agrawal and Boneh in [25] proposed the basic definition of homomorphic MAC. However, this idea is susceptible to tag

pollution attacks.

In [115], Kehdi *et. al* presented a homomorphic MAC scheme to detect data pollution attacks based on the subspace properties of RLNC, but it requires a high bandwidth overhead.

A practical defense scheme, called DART, against pollution attacks in network coding for wireless mesh networks was proposed in [132]. It is based on checksums which are very efficient to create and verify. Each node uses a checksum to verify only those packets that were received before the checksum itself was created. However, the delayed key distribution which is used in this scheme requires a clock synchronization of all the nodes.

Information-theoretic schemes. These schemes are characterized by considerable computation performance compared to the cryptographic schemes. However, they are not able to detect the polluted packets at the intermediate nodes of the network and also they can handle only a limited number of compromised nodes [21], [22].

2. **Tag Pollution:**

In tag pollution attacks, the adversary targets at modifying the tags appended to the end of the coded packets. There are few security schemes against tag pollution attacks. First of all, the authors in [26] proposed the RIPPLE scheme to counteract against the tag pollution challenge. This work is based on symmetric cryptography for NC authentication. RIPPLE allows a node to efficiently detect polluted packets and encode only the authenticated (i.e., verified) ones. However, the global synchronization among all nodes is the main issue of this scheme.

Furthermore, the authors in [3] propose the MacSig scheme which is tag pollution immune. The authors have presented a hybrid mechanism which enables the use of homomorphic MACs and one homomorphic signature. The basic idea is to pad each source packet with an extra symbol to make it orthogonal to a given vector. In addition, a signature is also appended at the end for resisting against tag pollution attacks. Moreover, a double-random key distribution method is proposed in [3]. However, a considerable bandwidth overhead is incurred due to the indexes of the keys added to the packets.

Finally, a key pre-distribution based tag scheme for practical NC has been proposed in [4]. By using this scheme, it enables all intermediate and sink nodes to verify the correctness of the received data messages. According to this key pre-distribution scheme, a KDC is responsible for distributing the keys. The KDC allocates secret key vectors at the source node to produce tags. Moreover, it assigns two secret key vectors to each node to check the correctness of its received packets. The authors believe that KEPTE can resist against tag pollution. The authors showed that this approach can be used as a practical network coding scheme.

3.4.3.2 Security via Network Codes

By exploiting appropriate network codes, it is possible to detect and correct corrupted packets, and there is opportunity to mitigate a significant devastating effect of Byzantine attacks (pollution attack) in network coding-enabled WNs. However, the mechanisms [116–118] that provide error detection and error correction may lead to some undesired problems. An error detection scheme creates monitoring overhead and there is a possibility to the sensors to correct the errors only after occurring pollution attacks, which may bring about epidemic disruptive problems for an NC system.

3.4.3.3 Security via Cooperative Mechanisms

Cooperative schemes that arise from network coding and can improve the verification efficiency. In these schemes, users not only cooperate to distribute content, but also cooperate to protect the rest sensors against adversary users in NC-enabled WNs. Moreover, the cooperative schemes for NC-enabled WNs are able to detect compromised nodes and isolate the adversary nodes [20, 119, 120]. In [20], one of the most famous cooperative schemes is presented. It is an efficient scheme to prevent Byzantine attacks against bogus alert messages based on secure random checksums. Another cooperative mechanism is a watchdog, which detects and removes adversaries by dynamically adjusting the routing paths [121]. It consists of the following three phases: monitoring, detecting, and isolating the malicious nodes. After monitoring and detection phases, nodes inform each other about these malicious nodes and finally they run the isolating phase. However, due to the significant overhead produced, this mechanism is not efficient for WNs.

The algebraic watchdog presented by Kim et al [122] have proposed a network in which the participants check their neighborhood locally to enable a secure global network. According to the homomorphism property of the used hash functions, all the intermediate nodes in the pollution attack model are able to verify the validity of the encoded packets on-the-fly prior to mixing them algebraically. As a result, the intermediate nodes can collaborate to verify the transit packets and the polluted packets will be dropped early and malicious nodes can be detected and isolated.

Le and Markopoulou introduced a homomorphic MAC scheme, called SpaceMac [123]. SpaceMac allows a node to verify the received messages if they belong to a specific subspace. Moreover, by using SpaceMac, the authors have proposed another cooperative scheme to detect and locate the adversary nodes.

3.5 Conclusion

Wireless networks are characterized by limitations that can affect the advancement of the wireless communications. These limitations are related to low communication bandwidth, packet loss and power consumption.

Towards this direction, the application of NC technology in several networks, such as CPSs and WSNs, leading to NC-enabled WNs, can be a promising solution. Recent research efforts show that NC technology can provide significant benefits to WSNs in terms of improvement in network capacity and robustness among others. However, due to the fact that NC technology

enables the intermediate nodes to manipulate and mix incoming data packets from neighboring nodes as well as transmit data packets between them, security issues are raised in NC-enabled WSNs.

Additionally, these security issues can be exploited by attackers to compromise the security of the wireless networks, which is a critical factor for their success. Therefore, in this chapter, we have focused on a survey of existing NC-enabled WNs attacks that can potentially have a negative impact on the security of the whole network. Furthermore, we have presented mitigation approaches against the most common attacks, and have positioned these categories. Our objective was to highlight the current limitations, and provide a basis on which to build more reliable and efficient defense mechanisms against these attacks for NC-enabled WNs.

Chapter 4

Efficient MAC-based for Secure XOR NC

> "In theory, theory and practice are the same. In practice, they are not."
>
> Albert Einstein

Wireless Sensor Networks (WSNs) have been recently used in a plethora of applications of many different areas, such as surveillance monitoring, environmental monitoring, traffic control, natural disaster prevention and e-health. However, WSNs are characterized by low communication bandwidth and power consumption constraints. Towards this direction, XOR Network Coding (NC) is a promising solution for WSNs. Nevertheless, XOR NC is vulnerable to pollution attacks, where adversaries (i.e., compromised intermediate nodes) inject into the network corrupted packets that prevent the destination nodes from decoding correctly. This has as a result, not only leads to network resource wasting, but also reduces the energy efficiency at the intermediate nodes. In this sense, pollution attacks constitute a serious threat against WSNs (i.e., resource-constrained wireless networks) that should be addressed. Therefore, in this chapter, we propose an efficient Message Authentication Code (MAC)-based scheme providing resistance against pollution attacks in XOR NC-enabled WSNs. Our results show that the proposed MAC-based scheme is more efficient compared to other competitive schemes for securing XOR NC against pollution attacks in resource-constrained wireless networks, in terms of communication bandwidth and computational complexity.

4.1 Introduction

Wireless Sensor Networks can support many important applications such as remote environmental monitoring and target tracking, and have gained worldwide attention in recent years as a part of the emerging Internet of Things (IoT) paradigm. Basically, the sensors in WSNs are small, with limited processing and computing resources. Thus, they inherit the limitations of WSNs in terms of communication bandwidth, reliability and power consumption that should be addressed. Thus, Network Coding (NC) can be a promising solution for WSNs, since it is a technique that can provide network capacity improvement and lower energy consumption [5–7]. More precisely, due to the lightweight complexity which XOR Network Coding (NC) has, it can be an encouraging solution for WSNs. Nevertheless, XOR NC is

vulnerable to pollution attacks, where adversaries (i.e., compromised intermediate nodes) inject into the network corrupted packets that prevent the destination nodes from decoding correctly. Even a small proportion of corrupted packets can quickly propagate into other packets via recoding, occurred at the intermediate nodes, and lead to infection of a larger number of packets. This has as a result not only network resource wasting, but also reduces energy efficiency of the nodes.

Therefore, in this chapter, we propose an efficient Message Authentication Code (MAC)-based scheme providing resistance against pollution attacks in XOR NC-enabled wireless networks. Our results show that the proposed MAC-based scheme is more efficient compared to the scheme proposed in [2], in terms of communication bandwidth and computational complexity. To the best of our knowledge, the proposed scheme in [2] is the most competitive scheme in the literature for securing XOR NC-enabled wireless networks.

Therefore, the main contributions of this chapter are as follows:

- We propose an efficient Message Authentication Code (MAC)-based scheme providing resistance against pollution attacks in XOR NC-enabled. Our proposed scheme makes use of a number of MACs which are appended to the end of each native packet.
- We discuss the communication overhead and computational complexity incurred by the use of our proposed scheme, as well as the comparison of these overheads with one the most competitive scheme [2] in the literature.
- We present the key distribution model which is adopted by our proposed scheme.

The remainder of this chapter is organized as follows. Section 4.2 provides the network model of the WSN scenario, where XOR NC is applied. In addition, in this Section, we present the adversary model of this XOR NC-enabled WSN, as well as the key distribution model which is adopted by our proposed scheme. In Section 4.3, our proposed scheme is provided. Subsequently, the security analysis of our proposed scheme is given in Section 4.4. In Section 4.5, the performance evaluation of the proposed scheme is analyzed. Finally, Section 4.6 concludes the chapter.

4.2 Models and Assumptions

In this section, we provide the network model of the WSN scenario, where XOR NC is applied. Moreover, we provide the adversary model of this XOR NC-enabled WSN, as well as the key distribution model which is adopted by our proposed scheme.

4.2.1 Network Model

Our XOR NC-enabled WSN can be modeled as a directed multigraph (S, I, e) which consists of the following components, as it is also shown in Figure 4.1:

- **Source node** S: We have a source S (i.e., a sensor node) which wants to multicast its messages. To achieve that, each message is divided into a sequence of packets and these packets are multicasted. Each packet consists of a number of codewords.

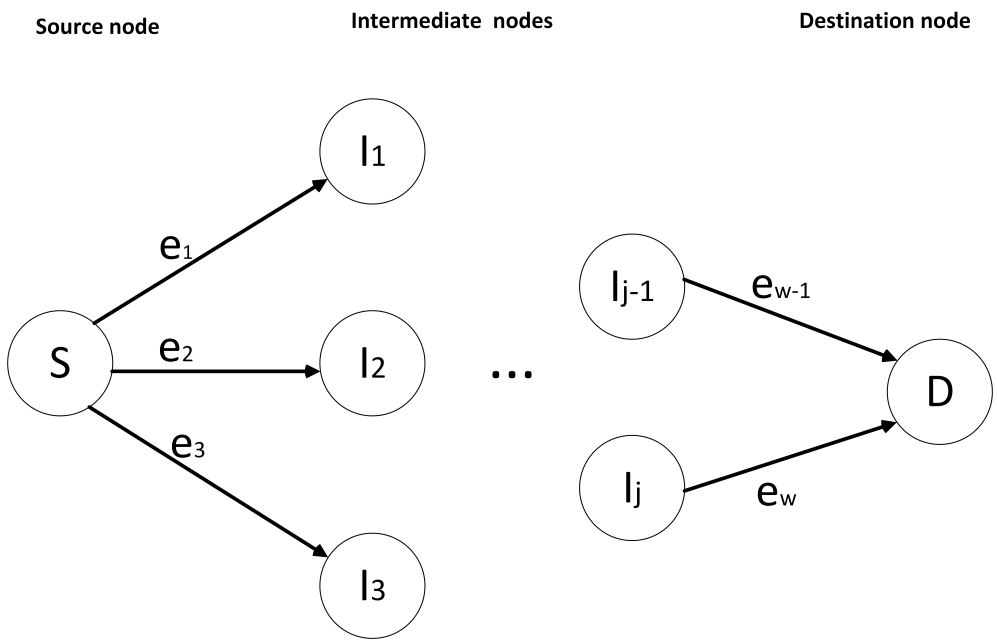

Figure 4.1: The Network Model of our XOR NC-enabled WSN.

- **Non-source node set** I: This set includes the intermediate nodes (i.e., relay nodes) and the destination node which recode and decode packets. In Figure 4.1, the set of non-source nodes is represented as: $I = \{I_1, ..., I_j, D\}$.

- **Link set** e: This set consists of all the links in the network. As a link is defined as the connection between each pair of two nodes. In Figure 4.1, the set of links is represented as: $e = \{e_1, ..., e_w\}$.

We consider two types of packets: native packets and coded packets. Native packets are packets generated at the source node. On the other hand, coded packets are packets which are encoded and recoded at the source and intermediate nodes. However, for simplicity, we refer to them as packets when it is not required to distinguish the native from the coded packets. At the setup phase and according to the assumption made by the most existing schemes [19], [126], the source divides each message into a sequence of native packets and partitions them into generations. In our model, the source $\mathcal{S}$ wants to send a number of native packets (i.e. $M_1, M_2, \cdots, M_n$) to the destination D. Each native packet M_i, which is divided into m codewords, can be represented as a row vector as follows:

$$M_i = (m_{i,1}, m_{i,2}, \cdots, m_{i,m}), \tag{4.1}$$

where $i = 1, \cdots, n$.

Each codeword stands in a finite field $\mathbb{F}_p$, where p is a power of prime number. Our model encodes the codewords over the field of size 2. Typically, each codeword is 256-bit long ($\lfloor \log_2 p \rfloor$).

Additionally, we denote the coded packet as E, which can be represented as following

$$E = \alpha_1 M_1 \oplus \alpha_2 M_2 \oplus \cdots \oplus \alpha_n M_n, \tag{4.2}$$

where $\alpha_i \in \{0, 1\}$ for $i = 1, \cdots, n$. The intermediate and destination nodes use the received coded packets E to verify the native packets.

4.2.2 Adversary Model

We assume that the source node and the destination node are always trusted and there is not any possibility to be forged. However, the intermediate nodes can be compromised. A compromised node can play the role of an adversary that is able to wiretap all the data packets that are transmitted over the network. The adversary's goal is to achieve pollution attack. Moreover, it is considered that the adversaries have limitations in computation power, and thus, they can only perform in polynomial-time.

4.2.3 Key Distribution Model

We assume that a set of keys are distributed to all participant nodes in a secure and authenticated manner by a Key Distribution Center (KDC) according to our key distribution model. In this Section, we provide a brief description of our key distribution model that we have adopted for our proposed MAC-based scheme against pollution attacks in XOR NC-enabled WSNs.

According to our key distribution model, if the keys are distributed properly, then no coalition of c compromised nodes can deceive another node. It is shown in [133] that in order to resist against c compromised nodes, the number of the keys at the source must be $c + 1$ times larger than the number of the shared keys at the relay/destination nodes. This model allows each verifier to obtain a number of shared keys in order to verify the coded packets. In our proposed scheme, this key distribution model assigns only one key to the intermediate and destination nodes so that they verify the received coded packet. Only one key is assigned to each intermediate and destination node, since in case that more than one key is assigned then there is the possibility to verify the coded packet more than once. However, this possibility needs more computational complexity.

4.3 Proposed MAC-based Scheme

In this section, we present the proposed MAC-based scheme against pollution attacks in XOR NC-enabled WSNs. Our proposed scheme is based on the secure XOR NC MAC-based scheme, defined by Yu et al. [2]. In this scheme, the source appends multiple MACs (tags) to each native data packet, where each MAC can authenticate only a part of the packet and the parts authenticated by different MACs are overlapped. Thus, multiple forwarders can collaboratively verify different parts of packet using the MACs with their own shared keys. It is shown that by carefully controlling the overlapping between the parts authenticated by different MACs, this scheme can filter polluted packets in a few hops with a high probability.

Our scheme is defined as a triple of probabilistic polynomial time (PPT) algorithms **(Setup, MAC, Verify)** as follows:

•**Setup:**

The KDC distributes the required keys to all nodes, the source node sets the security parameters and uses the assigned keys to generate the corresponding MACs (tags).

•**MAC:**

The source node calculates and appends a number of MACs (tags) to the end of each native packet.

•**Verify:**

Verification is based on the coded packets, the appended MACs (tags), and the shared keys. If verification succeeds, the received packets will be accepted and used for further recoding or decoding. Otherwise, the received packets are discarded.

The summary of notations used in this chapter is presented in Table 4.1

Table 4.1: Summary of Notations

Parameter	Meaning
n	The number of native packets per generation
m	The number of codewords of each native packet
l	The number of MACs
p	The size of finite field and each codeword
M_i	Each native packet
m_j	Each coded packet
h	The hash function
$K_{i,j}$	The key
$id_{i,j}$	The index of the key
r_j	The index of u codewords
u	The number of codewords used for each MAC
c	The number of compromised nodes

4.3.1 The Construction of the MAC-based Scheme

The construction of the proposed MAC-based scheme consists of the following:

- **Setup**

  Initially, the source node defines the number of MACs (tags) appended to the end of each native packet. This number of MACs is denoted as l. Then, the source node chooses randomly a value u which is the number of codewords that will be used in the generation process of each MAC. To define the codewords, that will be used in the generation of the l MACs, the source node makes use of l random integers $r_1, r_2, ..., r_l$, where $r_j \in [1, m]$ for $j = 1, ..., l$. These random integers are calculated based on a pseudo-random function. Each r_j represents the indexes of the u codewords that will be used in the generation process of each MAC. Also, a hash function $h : \mathbb{F}_p^u \rightarrow \mathbb{F}_p$, where $\mathbb{F}_p$ is a finite field of size p, is defined by the source node. This hash function will be used by the source node to

create the hash value of the u codewords. Then, based on the number of MACs (i.e., l), the KDC distributes l keys $k_1, k_2, ..., k_l$ to the source node. Finally, the KDC selects, based on the key distribution model, described in Section 4.2.3, one key from those l keys and distributes it to each non-source node.

- **MAC**

  The source node calculates l MACs (tags) and appends them to the end of each native packet. Each MAC is calculated as follows:

$$MAC_{i,j} = E(id_{i,j}, r_j, h_{i,j})_{k_{i,j}}, \tag{4.3}$$

  Where $E(.)$ denotes encryption using the key $k_{i,j}$, which is one of the l keys, distributed by the KDC, $i = 1, ..., n$ and $j = 1, ..., l$. In addition, $id_{i,j}$ denotes the index of this key, r_j is the index of u codewords, and $h_{i,j}$ is the output of the hash function h taking as input u codewords. More precisely, for XOR NC, the hash function h is defined as follows:

$$h_{i,j} = m_{i,r_{j,1}} \oplus ... \oplus m_{i,r_{j,u}}, \tag{4.4}$$

  where $i = 1, ..., n$, and $j = 1, ..., l$.

  Then, the source node transmits the following:

$$(M_i, id_{i,1}, MAC_{i,1}, ..., id_{i,l}, MAC_{i,l}), \tag{4.5}$$

  where $i = 1, ..., n$.

  We illustrate an example of the MAC generation in Figure 4.2.

- **Verify**

  Each non-source node (i.e., intermediate and destination nodes) holds a key which was distributed by the KDC as it has been mentioned in Setup phase. Using this key, the non-source node decrypts the corresponding MAC of the received coded packet. Thus, the non-source node obtains the indexes of the u codewords and the value of the hash function h, that was calculated in the source node. Afterwards, the non-source node calculates the hash value $h^{'}$ of the codewords that correspond to the obtained indexes, according to Equation 4.4, and compares it with the obtained hash value h. If they are equal, the received coded packet is considered as uncorrupted and is accepted, otherwise, the received coded packet is discarded.

4.4 Security Analysis

In this Section, we analyze the security of our proposed scheme in terms of the probability that a corrupted packet can not be detected by the next hop and how this probability can be affected by the number of codewords (u) as well as the number of MACs (l).

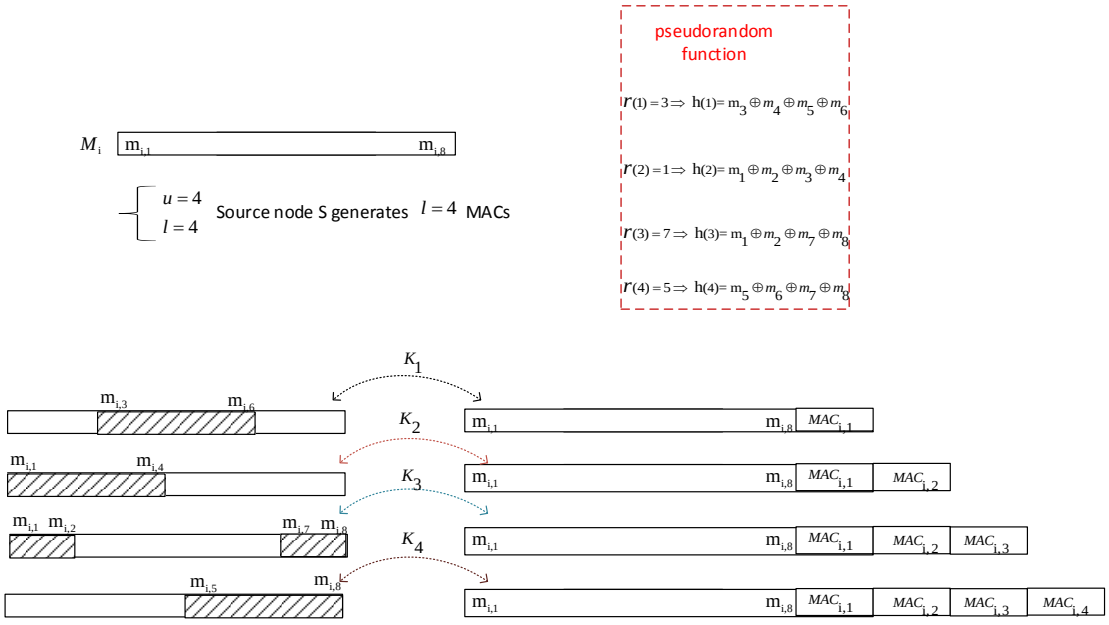

Figure 4.2: An example of tag generation. The source node has to calculate and append 4 MACs to the end of the native packet M_i, which consists of 8 codewords (i.e., $m_{i,1}, ..., m_{i,8}$). To generate the 4 MACs, a pseudo-random function is used firstly to generate 4 random values (i.e., $r_1, ..., r_4$), one for each MAC. Based on these random values, the source node defines the indexes of the codewords, which will be the input to the hash function h in order to calculate the corresponding hash values $h(1), ..., h(4)$.

We have considered that the adversary (i.e., the compromised node) knows the distributed key and thus, based on Equations 4.3 and 4.4, he is able to identify which codewords are verified by the corresponding MAC. Hence, he can corrupt these codewords and generate a false MAC for the corrupted packet. However, the corrupted codeword can be detected by the next node with a high probability.

Theorem 2. *The probability that a corrupted codeword in one MAC is not detected by the next node is not greater than* $(1 - \frac{u}{m})^{l-1}$.

Proof. In our proposed scheme, each MAC verifies the integrity of u codewords out of m codewords. Therefore, the probability that each codeword can occur in one MAC is $\frac{u}{m}$. However, the probability that this codeword does not occur in other MACs is calculated as follows:

$$p_{occur} = (1 - \frac{u}{m})^{l-1} \tag{4.6}$$

Consequently, if an adversary corrupts a codeword of a MAC, then the probability that the corrupted codeword will not be detected by the next node is given by Equation 4.6. □

From Theorem 2, it is clear that the probability of a corrupted packet not to be detected by the next node is given by Equation 4.6 as well. In Figure 4.3, based on Equation 4.6, we

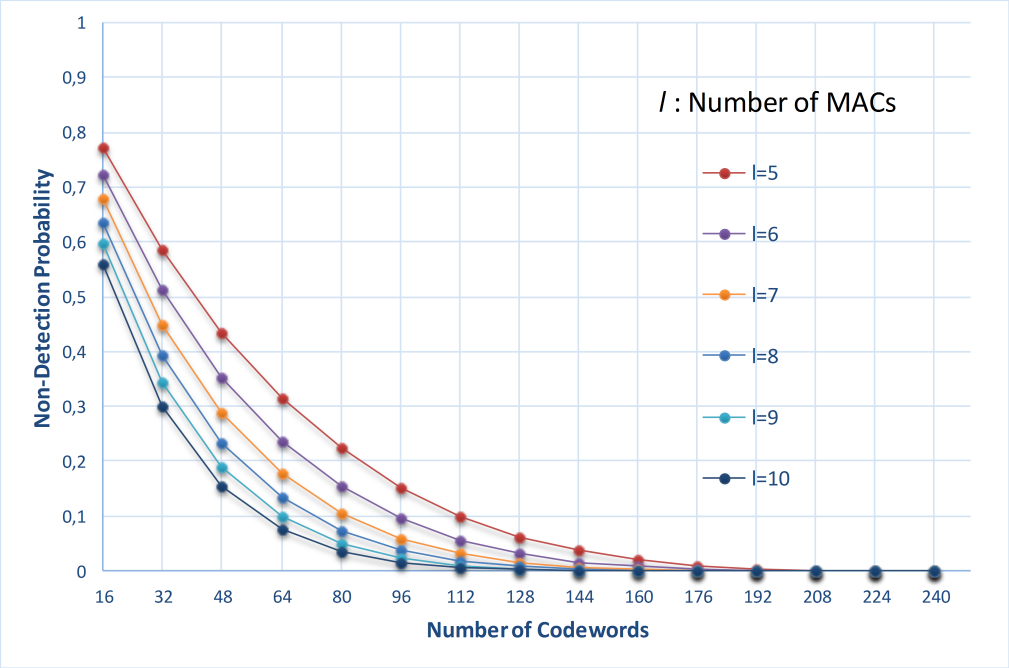

Figure 4.3: Corrupted Packet Non-Detection Probability in terms of the Number of Codewords.

illustrate this probability in terms of the number of codewords for different number of MACs. As it is shown in Figure 4.3, by increasing the number of codewords, the probability of a corrupted packet not to be detected by the next node decreases.

In addition, the probability of a corrupted packet not to be detected by the next node is also affected by the number of MACs.

Theorem 3. *By increasing the number of MACs, the probability that a corrupted packet not to be detected by the next hop decreases.*

Proof. Suppose that the total number of compromised nodes is c. Consequently, the adversary can totally obtain c keys. Thus, the probability that the adversary can decrypt one MAC of the coded packet is denoted as p_k and defined as follows:

$$p_k = \frac{c}{l} \tag{4.7}$$

Hence, adopting Equation (12) proposed in [2], the probability that the adversary, holding c keys, can decrypt c MACs out of the l MACs, is the following:

$$p_{dec}(c) = \binom{l}{c} p_k^c (1 - p_k)^{l-c} \tag{4.8}$$

where $c \ll l$.

As a result, the probability that a corrupted packet not to be detected by the next hop is given by Equation 4.8 as well. According to this Equation, it is clear that by increasing the number of MACs, the probability that the corrupted packet not to be detected by the next hop decreases. □

4.5 Performance Evaluation

In this section, we analyze the performance of our MAC-based scheme in terms of communication bandwidth and computational complexity.

4.5.1 Communication Bandwidth

To calculate the communication overhead of our proposed scheme, we take into consideration: a) the number of codewords of the native packet, b) the l MACs appended to the end of each native packet, and c) the l key indexes appended to the end of each native packet as well.

Each native packet has m codewords and each codeword requires $log_2\, p$ bits. Thus, the bit-length of each native packet is $m * log_2\, p$ bits.

In Figure 4.4, based on Equation 4.8, we illustrate this probability in terms of the number of MACs for different number of compromised nodes.

On the other hand, according to Equation 4.3, each MAC consists of a hash value, a random number, and the key index. Based on Equation 4.4, the hash value is the XOR result of u codewords and thus, it requires $log_2\, p$ bits.

Moreover, the random number, used to identify the indexes of the codewords, requires $log_2\, m$ bits. Finally, the index of the key requires $log_2\, |l|$ bits. Consequently, the bit-length of l MACs is $l * (\log_2\, p + \log_2\, m + \log_2\, |l|)$ bits. In addition, each key index appended to the

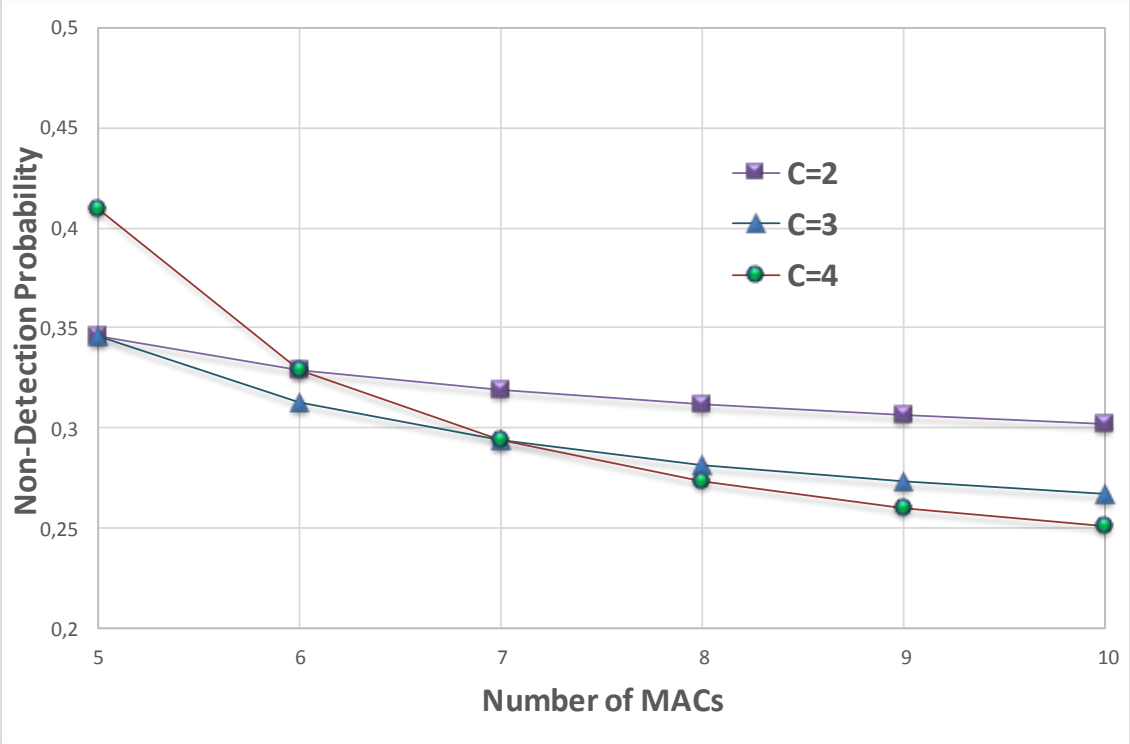

Figure 4.4: Corrupted Packet Non-Detection Probability in terms of the Number of MACs.

end of each native packet requires $log_2 |l|$ bits. Therefore, the bit-length of l key indexes is $l * (log_2 |l|)$ bits. However, similar to [2], in order to reduce the length of the coded packets in our scheme, only one key index can be appended to the end of each native packet.

Thus, our MAC-based scheme has the following communication overhead:

$$\frac{l * (\log_2 p + \log_2 m + \log_2 |l|) + \log_2 |l|}{m * \log_2 p} \tag{4.9}$$

Assuming that the number of MACs (i.e., l) is constant, m is equal to 256, and p is 256-bit long, the value of $log_2 |l|$ and the value of $log_2 m$ are negligible compared to the value of $log_2 p$. Hence, from the Equation 4.9, the communication overhead is calculated as $\frac{l}{m}$, which means that our scheme has the same communication overhead as the proposed scheme in [2]. However, our proposed scheme can detect the corrupted packets earlier (i.e., next node) and thus, communication bandwidth is saved.

4.5.2 Computational Complexity

According to Equations 4.3 and 4.4, $(u-1)$ XOR operations are needed to generate each MAC in our scheme. Thus, the generation of l MACs requires $l * (u-1)$ XOR operations. Nevertheless, the proposed scheme in [2] requires $l * u$ XOR operations to generate l MACs.

Moreover, to verify a coded packet in each non-source node, $(u-1)$ XOR operations are required in our scheme, since each non-source node calculates the hash value of the codewords that corresponds to the obtained indexes, according to Equation 4.4. On the other hand, the scheme proposed in [2] requires $l * u$ XOR operations for verification. In Figure 4.5, we illustrate the number of XOR operations required to verify the coded packet in our scheme and the scheme proposed in [2]. It is worthwhile to mention that, in contrast to the scheme proposed in [2], our scheme does not depend on the number of MACs (i.e., l).

Table 4.2 summarizes the number of XOR operations required for MAC generation and verification in our proposed scheme and the scheme proposed in [2]. According to Table 4.2, our scheme requires less XOR operations in total than the scheme proposed in [2] and thus, our scheme is more efficient in terms of computational complexity.

Table 4.2: Number of XOR Operations

	XOR NC Proposed Scheme	**[2]**
MAC	$l * (u-1)$	$l * u$
Verify	$u-1$	$l * u$

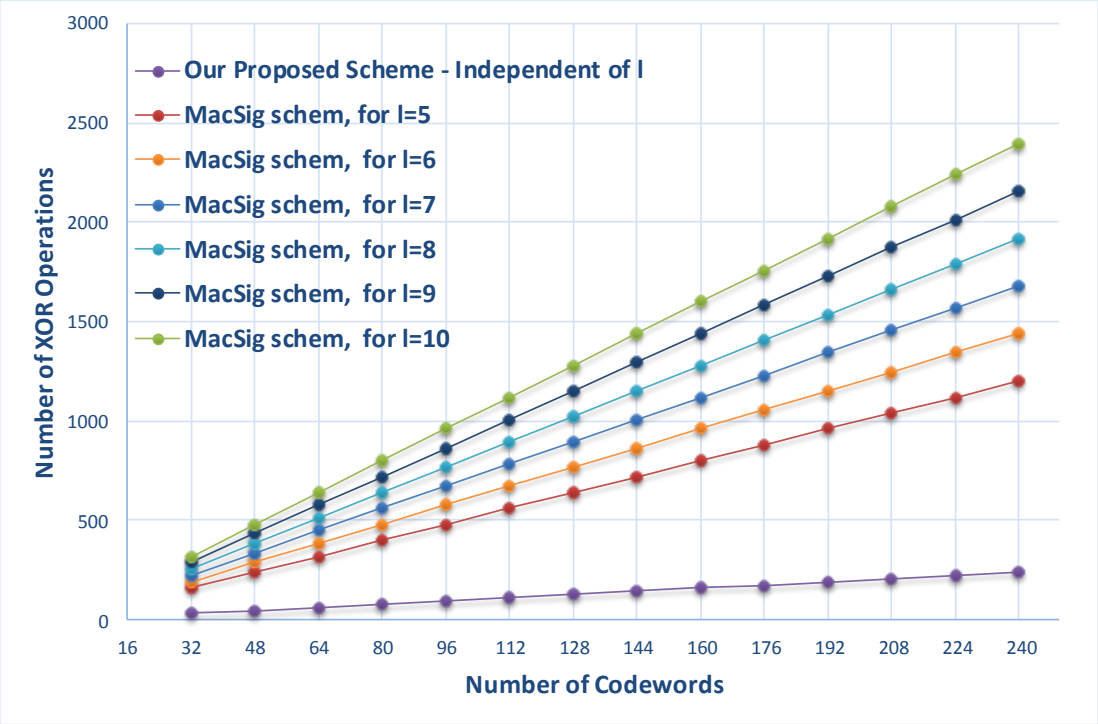

Figure 4.5: The number of XOR operations required to verify a coded packet in terms of the number of codewords in our proposed scheme and the scheme proposed in [2].

4.6 Conclusion

In this chapter, we have proposed an efficient Message Authentication Code (MAC)-based scheme providing resistance against pollution attacks in XOR NC-enabled WSNs. Our proposed scheme makes use of a number of MACs which are appended to the end of each native packet. Our results show that the proposed MAC-based scheme is more efficient compared to the scheme proposed in [2], in terms of communication bandwidth and computational complexity. The proposed scheme in [2] is the most competitive scheme in the literature for securing XOR NC against pollution attacks in resource-constrained wireless networks. Particularly, our scheme saves communication bandwidth, since it detects corrupted packets in the next hop with high probability, and its computational complexity is lower because it requires less XOR operations than the scheme proposed in [2]. Moreover, we observe that verification process in our proposed scheme is almost 5 to 10 times (l times) faster than [2].

Chapter 5

Advanced Homomorphic MAC for Secure RLNC

> "It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong."
>
> Richard Feynman

Network Coding (NC) has shown a considerable improvement in terms of capacity and robustness compared to traditional store-and-forward transmission paradigm. However, since the intermediate nodes in network coding-enabled networks have the ability to change the packets en-route, network coding-enabled networks are vulnerable to pollution attacks where a small number of polluted messages can corrupt bunches of legitimate messages. Recently, research efforts have focused towards schemes for protecting the transmitted messages against data pollution attacks. However, most of them are limited against tag pollution attacks. In this chapter, we aim to propose four schemes which enable the intermediate and sink nodes to detect and discard the polluted packets.

5.1 Introduction

Random Linear Network Coding (RLNC) was proposed by Ho et al. in [18] as a fully distributed approach for performing NC. In RLNC, each node selects randomly a set of coefficients and uses them to make linear combinations of the incoming packets. Furthermore, it is worthwhile to mention that RLNC achieves the same capacity as that achieved by the Max-flow Min-cut theorem [30]. However, RLNC is susceptible to data pollution attacks, where an adversary (e.g., compromised intermediate node) injects into the network corrupted packets that prevent the destination nodes from decoding correctly. This has as a result not only network resource waste but also increases energy waste at the nodes [128]. Hence, during the past few years, several schemes have been presented to provide resistance against data pollution attacks. Among them, those based on homomorphic Message Authentication Codes (MACs) are considered as a low-complexity solution for data pollution attacks [3, 25, 115]. However, most of the homomorphic MAC schemes are vulnerable to tag pollution attack, which was first defined by Li et al. in [26]. In tag pollution attacks, the adversaries modify

tags (i.e., MACs) appended to the end of the coded packet. A tag is a piece of information appended to the end of the coded packet to ensure integrity of the transmitted packet data. Each tag has a fixed position in the coded packet and thus, it is easy for an adversary to control its pollution.

Therefore, in this chapter, we present four schemes which can provide resistance against pollution attacks. These schemes, based on the tag creation, are discussed in two parts. In the first part, two homomorphic MAC-based schemes for network coding-enabled wireless networks are discussed. The first proposed scheme called Dual Homomorphic MAC Scheme, which makes use of two types of tags (i.e., MACs and D-MACs) to provide resistance against data pollution attacks and partially tag pollution attacks. The second proposed scheme is called efficient Dual Homomorphic MAC Scheme. This scheme makes use of the idea of Dual Homomorphic MAC Scheme and a signature to provide resistance against data pollution attacks and tag pollution attacks.

In the second part, the third scheme and fourth scheme are presented. The third scheme called Null Space-based Homomorphic MAC Scheme, makes use of two types of tags, where the first tags are based on the inner product of the native packet and keys, and the second tags are based on the null space property. The fourth scheme, called an efficient Null Space-based Homomorphic MAC Scheme, makes use of only one set of tags which can provide resistance against data pollution and tag pollution attacks.

The remainder of this chapter is organized as follows. In Section 5.2, we give an overview of the background and related work on RLNC and secure RLNC in wireless networks. In Section 5.3, the proposed Dual Homomorphic MAC Scheme is provided. Subsequently, the efficient Dual Homomorphic MAC Scheme is given in Section 5.4. In Section 5.5, the Null Space-based Homomorphic MAC Scheme is introduced. Furthermore, the efficient Null Space-based Homomorphic MAC Scheme is proposed in 5.6. Finally, Section 5.7 concludes the chapter.

5.2 Related Works

In this Section, we present related work on security schemes against pollution attacks in random linear network coding-enabled wireless networks. In addition, we present the key distribution mechanisms used by these security schemes.

This chapter focuses on the subject of security in linear network coding when passive attackers are present. As mentioned earlier, network coding by itself and without employing any security module has the ability to provide a certain level of inherent security due to the fact that the data traveling through the network is constantly evolving by the intermediate nodes. Therefore, at any given time, every network channel usually carries a symbol which is not an exact copy of any of the original data symbols that are generated at the source nodes. This innate security property which is inherited from the very fundamental characteristics of network coding, sometimes satisfies the security needs of a network; however, for some applications that require higher security assurance, additional security modules are needed.

5.2.1 Security schemes against pollution attacks in NC-enabled networks

Most of the research in the field of pollution attacks in network coding is focused on two types of security schemes: cryptographic schemes [25], [114] and information-theoretic schemes [21], [22].

- *Cryptographic schemes.* These schemes rely on additional verification information which is added by the sources through cryptographic techniques. This allows intermediate nodes to verify the original messages and filter out the polluted ones. This category includes the homomorphic hashing schemes, the homomorphic signature schemes and the homomorphic MACs.

    1. *homomorphic hashing schemes* are based on a homomorphic hash function which is applied by the source to the messages. These schemes rely also on additional secure communication channels in order to transmit the calculated hash values to the intermediate nodes. A well-known homomorphic hashing scheme is proposed by Krohn et al. in [19] which enables the intermediate nodes to verify on-the-fly. The extension of this work was presented in [20] which is focused on network coding-enabled networks and further reduces the expensive computation cost of hash functions in each intermediate nodes by enabling cooperation verification.
    2. *homomorphic signature schemes* were introduced for the first time by Charles, Jain and Lauter in [131] and have been based on Weil pairing over elliptic curves. The homomorphic property of the signatures in these schemes allows nodes to sign any linear combination of the incoming packets without contacting the signing entity. The calculation of signature covers the whole augmented message. In [24], Yu et al. proposed a homomorphic signature function which allows the relay nodes to verify the received messages by generating the signatures without contacting the signing entities. In Yu's scheme, the communication does not need any extra secure communication channel. Their experimental results show that its verification efficiency is improved up to 10 times compared to the other related work (i.e., CJLs scheme [131]). Moreover, in [21], an alternative lightweight scheme is proposed. It is much faster rather than the Yu's scheme, but it is not as secure as the first one.
    3. *homomorphic MACs* are based on appending some extra information to the original message. The basic definition of homomorphic MAC is defined by Agrawal et al. in [25] which allows the integrity checking of the network coded data. However, this idea is susceptible to tag pollution attacks. Kehdi et al. [115] presented a homomorphic MAC scheme to detect pollution attacks based on the subspace properties of random linear network coding, where null keys for verification are used. As a lot of null keys are used in each generation, a high bandwidth overhead can be incurred. RIPPLE [26] was proposed to counteract against the tag pollution problem. This work is based on a symmetric key based scheme for network coding authentication. RIPPLE allows a node to efficiently detect corrupted packets and encode only the authenticated (i.e., verified) ones. However, the global synchronization among all nodes is the problem of this scheme. In [3], the authors presented a hybrid-key cryptography approach which could achieve authentication in the presence of both data and tag pollution attacks. To achieve this, a number of tags and a signature are appended to each transmitted message. However, the signature which is necessary for preventing tag pollution is a time consuming process.

- *Information-theoretic schemes.* The information-theoretic schemes benefit from the reasonable computation performance in comparing to the cryptographic schemes, but they have two main drawbacks. First of all, they are not able to detect the polluted

messages at the intermediate nodes of the network. For example the work presented in [21], which extends the random network coding, relies on coding redundant information into the original messages allowing only the receivers to detect the polluted messages. Furthermore, the information-theoretic schemes are characterized by the limitation on the number of compromised nodes. These two drawbacks comprise the main motivation to interest to the cryptographic schemes.

5.2.2 Key Distribution Model

Nowadays, in emerging multicast communication systems, (e.g., pay TV and videoconferencing services), the key distribution is a challenging issue. There are various proposed schemes for key distribution in multicast systems. Key distribution models can be categorized in trusted-server schemes [134], public-key schemes and key pre-distribution schemes [135], [4]. The trusted-server and public-key schemes are infeasible to be used in wireless networks as the wireless nodes (i.e., sensors) have limitations in terms of energy and computing processes. On the other hand, key pre-distribution schemes are feasible to be used in wireless networks. In these schemes, the shared keys are distributed among the nodes of a group in such a way that every node in the group to be able to compute individually a common key associated with that group. Two examples of key pre-distribution schemes are given as follows.

In MacSig which is an efficient subspace authentication scheme for NC and presented in [3], the double-random key distribution has been proposed. This proposed key distribution model includes two random procedures. First of all, they assign each node with a random set of keys; next, the source node randomly selects a subset of keys from the key pools and uses these keys to generate the MACs. However, it is necessary to attach the indexes these keys to each packet. Hence, a considerable bandwidth overhead is incurred.

In KEPTE (Key Pre-distribution-based Tag Encoding) scheme [4], a key distribution model for practical NC has been proposed. According to this key distribution model, a Key Distribution Center (KDC) is responsible for the distribution of the keys. The KDC allocates N secret vectors at the source node to produce N tags. Moreover, it assigns two secret vectors to each node to check the correctness of its received packets.

Part I

Dual Homomorphic MAC

5.3 A Dual Homomorphic MAC Scheme

5.3.1 Introduction

Nowadays, NC has been used in various applications over different networks, such as wireless mesh networks [5], wireless sensor networks [136] and peer-to-peer systems [6]. In contrast to traditional and classical commodity flow, where information is only routed or replicated, information flow can also employ coding operations at intermediate nodes. In [18], Random linear Network Coding (RLNC) was studied as a fully distributed method for performing network coding.

However, RLNC is more susceptible to data pollution attack than the traditional store-and-forward approach. If a data pollution attack is not detected at the forwarders, then the sink nodes will not be able to recover the source messages correctly. It is worthwhile to mention that even a small number of polluted messages can infect a large number of downstream nodes because the pollution propagates via recoding. Furthermore, RLNC-enabled networks are also vulnerable to tag pollution attack, which is a more sophisticated type of pollution attack. In tag pollution attack, the adversary targets at modifying (i.e., polluting) the tags carried by the messages rather than modifying the content of the messages. It is possible for a message with polluted tags to travel multiple hops until it is detected and discarded. Yet, this results in a waste of network bandwidth. Specifically, a MAC or tag is a small piece of information appended to the end of the message packet. This piece of information is the output of a MAC function taking as inputs the message packet and a secret key. However, this solution is vulnerable to tag pollution attacks.

In this section, we propose a new homomorphic MAC-based scheme, called Dual-Homomorphic MAC (Dual-HMAC) for NC-enabled WNs in order to mitigate against both data pollution attacks, and partially tag pollution attacks. In our scheme, the source generates multiple MACs and Dual MACs (D-MACs) for each message. Each MAC ensures integrity of the transmitted message and each D-MAC ensures integrity of the MACs. In other words, by appending D-MACs, the proposed scheme mitigates partially tag pollution attacks.

5.3.2 Models and Assumptions

In this section, we provide the network model of NC-enabled wireless networks where our proposed scheme is applied. Also we provide the adversary model of this kind of networks.

5.3.2.1 Network Model

In this chapter, similarly to Chapter 4, we assume that our network model is based on a directed multigraph (S, I, E) which consists of the following components, as shown in Figure 5.1:

- **Source node** S: We have a source S which wants to multicast its messages. To achieve that, each message is divided into a sequence of packets and these packets are multicasted. Each packet consists of a number of symbols.

- **Non-source node set** I: This set includes the intermediate and destination nodes which recode and decode packets.

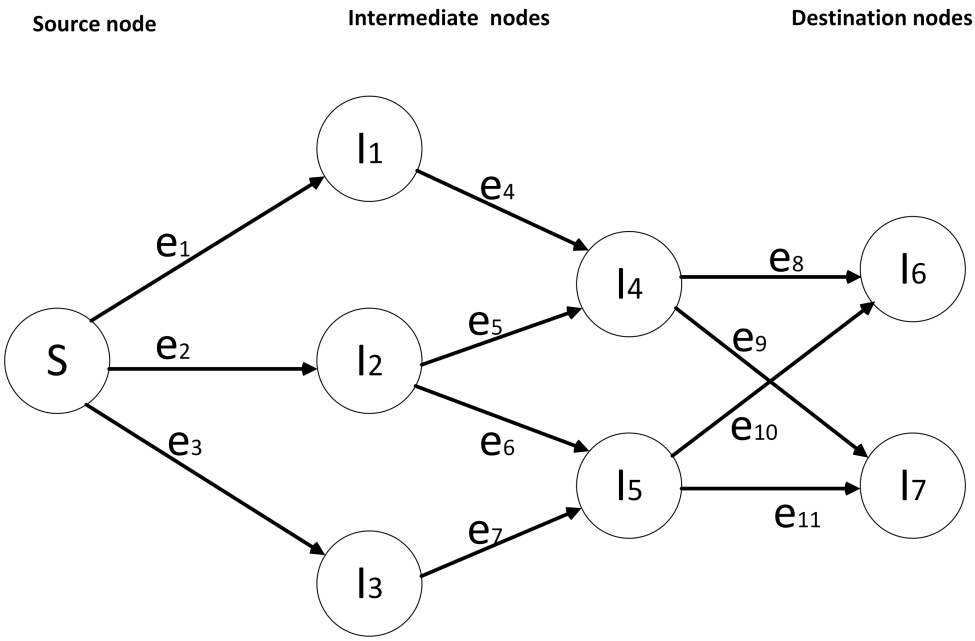

Figure 5.1: The network model of NC-enabled wireless networks where our proposed scheme is applied.

- **Link set** E: This set consists of all the links in the network. As a link is defined the connection between each pair of two nodes. In Figure 5.1, the set of links is represented as: $E = \{e_1, ..., e_{11}\}$.

We consider two types of packets: native packets and coded packets. Native packets are packets generated at the source node. On the other hand, coded packets are packets which are encoded and recoded at the source and intermediate nodes.

However, for simplicity, we refer to them as packets when it is not required to distinguish them between native and coded packets. At the setup phase and according to the assumption made by the most existing schemes ([19]- [126]), the source divides each message into a sequence of native packets and partitions them into generations.

Thus, we consider that each generation consists of m native packets denoted as $\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, ..., \underline{\mathbf{u}}_m$. Each packet $\underline{\mathbf{u}}_i$, for $1 \leq i \leq m$, is represented as a vector $\underline{u}_1, \underline{u}_2, ..., \underline{u}_n$ in the finite field $\mathbb{F}_p^n$, where $p = 2^8$ denotes the size of the finite field. In [137], it has been shown that 2^8 is usually sufficient for practical use and convenient for computation. Then, the source S generates a coded packet $\mathbf{u}_i$ for each native packet $\underline{\mathbf{u}}_i$ by prefixing $\underline{\mathbf{u}}_i$ with the i^{th} unit vector of dimension m.

The coded packet is represented as a row vector in the finite field $\mathbb{F}_p^{m+n}$ as follows:

$$\mathbf{u}_i = (\overbrace{\underbrace{0, \cdots, 0}_{i-1}, 1, 0, \cdots, 0}^{m}, \underline{u}_{i,1}, \cdots, \underline{u}_{i,n}) \in \mathbb{F}_p^{m+n} \tag{5.1}$$

Moreover, the source S transmits these coded packets to its neighbor nodes. An intermediate node buffers its received packets $\mathbf{u}_i$ temporarily and creates a coded packet y, which is a linear combination of a number of h coded packets $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_h$ belonging to the same generation. The coded packet is represented as follows:

$$y = \sum_{i=1}^{h} \alpha_i \mathbf{u}_i \tag{5.2}$$

where each $\alpha_i \in \mathbb{F}_p$ is a random coefficient chosen by each intermediate node. A coded packet y is considered to be valid if it is in the linear subspace spanned by the coded packets generated at the source. This is denoted as $y \in span\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$. In fact, when y is valid, these linear combination coefficients are the first m symbols of the packet y. Otherwise, y is invalid and it is denoted as $y \notin span\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$, which may be caused by transmission errors or pollution attacks. At the destination side, when a sink node has obtained m linearly independent coded packets, it can decode the coded packets, by using Gaussian eliminations [18], [138].

5.3.2.2 Adversary Model

Two types of pollution attacks are considered in our proposed schemes. The first one is the data pollution attack, in which the adversary exploits the knowledge of the pre-shared key to the compromised nodes in order to produce fake packet content. The main objective of the adversary is to create polluted packets that will pass the packet content verification of the next innocent intermediate nodes and cause incorrect decoding at the sink node.

The second type of pollution attacks includes the tag pollution attacks, in which the adversary randomly modifies tags so that packets with legitimate content will be discarded later from next intermediate nodes due to the corrupted tags. It is possible for a packet with legitimate content and corrupted tags to travel multiple hops until it is detected and discarded.

Figure 5.2 represents a tag pollution attack scenario over our network model topology. Particularly, two adversary nodes (i.e. I_1 and I_3) are considered in the scenario. The adversary node I_1 pollutes the tag t_5 of the coded packet P_1. However, the next nodes I_4, I_6, and I_7 have the keys K_1, K_3 and K_2 respectively, but not the key K_5 corresponding to the polluted tag t_5. Thus, the polluted packet P_1 will be accepted at the destination nodes I_6, and I_7. On the other hand, the adversary node I_3 pollutes the tag t_4 of the coded packet P_2. Since the next node I_5 has the key K_4 which is the corresponding key to the polluted tag t_4, the polluted coded packet P_2 is detected and discarded by I_5.

5.3.3 Proposed Scheme: Dual Homomorphic MAC (Dual-HMAC)

In this section, we present our Dual-HMAC scheme, which mitigates data pollution and partially tag pollution attacks for a network coding-enabled wireless networks. The Dual-HMAC scheme's outline, the construction and the correctness of the scheme as well as the key distribution model are provided in this section.

Our proposed solution is based on the homomorphic MAC scheme, defined by Agrawal et al. [25], and the MacSig scheme proposed in [3]. Additionally, our proposed scheme makes use of additional MACs, termed as D-MACs, in order to resist data pollution and partially tag pollution attacks. In contrast to the homomorphic MAC solution in [25], our proposed scheme is not susceptible to tag pollution attacks. Particularly, our proposed scheme can achieve resistance for half of the tags against tag pollution attack with significant low computational

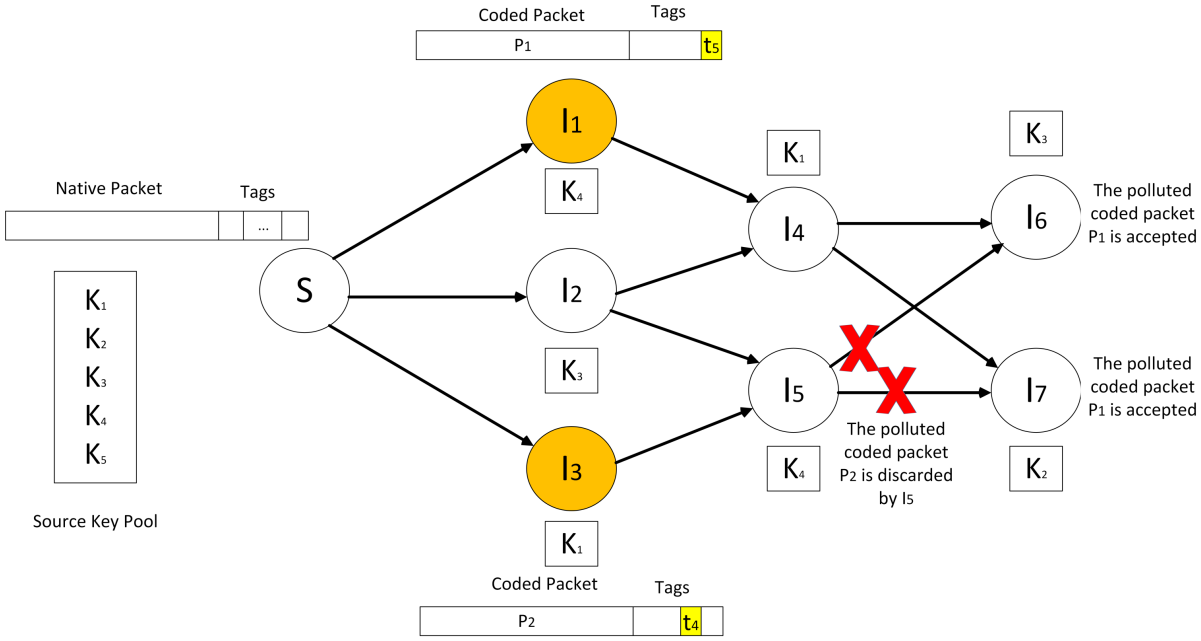

Figure 5.2: A tag pollution attack scenario over our network model topology.

complexity compared to the MacSig scheme, which uses a signature scheme characterized by a much considerable computational overhead.

Our scheme consists of three steps, as it is depicted in Fig 5.3, in order to generate the required MACs and D-MACs. In step 1, it computes a number of MACs (i.e., tags), for each message, according to the keys which are chosen randomly from the first pool of keys. Then, in step 2, it computes a number of D-MACs for the initial MACs, according to the keys which are chosen from the second pool of keys. Finally, in step 3, the computed MACs and D-MACs are appended to the message.

5.3.3.1 The Outline of the Dual-HMAC Scheme

In this section, we provide description of the proposed Dual-HMAC scheme with a formal method. This scheme includes four steps: *Setup*, *Tag generation*, *verification* and *encoding* detailed as follows:

1. *Setup*

   - Key Distribution Centre (KDC) distributes the following two sets of secrete key vectors to the source node S: $\mathcal{K}_S = \{\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_l\}, \mathbf{k}_h \in \mathbb{F}_p^{m+n+1}$ and $\mathcal{K}_S^{'} = \left\{\mathbf{k}_1^{'}, \mathbf{k}_2^{'}, \cdots, \mathbf{k}_{l'}^{'}\right\}, \mathbf{k}_{h'}^{'} \in \mathbb{F}_p^{l+1}$. KDC also distributes the following two subsets of secrete vectors to all intermediate nodes and sink nodes: $\mathcal{K}_{n_i} = \{\mathbf{k}_1, \cdots, \mathbf{k}_R\}, \mathbf{k}_r \in \mathbb{F}_p^{m+n+1}$, and $\mathcal{K}_{n_i}^{'} = \left\{\mathbf{k}_1^{'}, \cdots, \mathbf{k}_{R'}^{'}\right\}, \mathbf{k}_{r'}^{'} \in \mathbb{F}_p^{l+1}$.

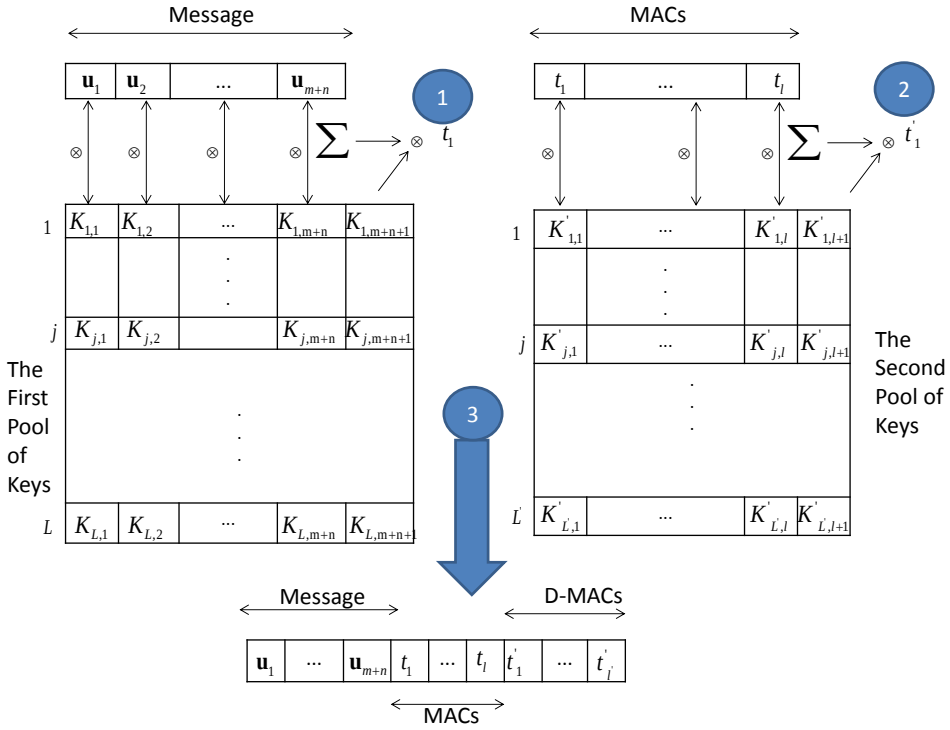

Figure 5.3: The three main steps of Dual-HMAC scheme.

- Note that $\mathcal{K}_S \bigcap \mathcal{K}'_S = \emptyset$, $\mathcal{K}_{n_i} \subset \mathcal{K}_S$, $\mathcal{K}'_{n_i} \subset \mathcal{K}'_S$, and if $i \neq j$, then $\mathcal{K}_{n_i} \neq \mathcal{K}_{n_j}$ so as $\mathcal{K}'_{n_i} \neq \mathcal{K}'_{n_j}$.
- Note that the subscriptions $h, h^{'}, r$ and $r^{'}$ represent the index of the key, which do not change in different subsets. In other words, if $\mathbf{k}_h = \mathbf{k}_r$, then $h = r$; if $\mathbf{k}_{h'} = \mathbf{k}_{r'}$, then $h^{'} = r^{'}$. The detail of probabilistic key distribution algorithm will be described in Section 5.3.3.4, as well as how to choose the key set size.

2. *Tag generation*

   For every data packet $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$, the source uses the $MAC\ (\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_l, \mathbf{u}_i)$ and $D\text{-}MAC\ \left(\mathbf{k}'_1, \mathbf{k}'_2, \cdots, \mathbf{k}'_{l'}, t_{\mathbf{u}_i,1}, \cdots, t_{\mathbf{u}_i,l}\right)$ algorithms to generate the $(l + l')$ tags.

3. *Verification*

   When a relay or sink node receives a coded packet $y \in \mathbb{F}_p^{m+n}$ with its tags, this node checks the correctness of packet using the algorithm *Verify* via its pre-distributed keys $\mathcal{K}_{n_i}$ and $\mathcal{K}^{'}_{n_i}$. If the output is 1, then the received coded packet y is considered to be a correct one. Otherwise, this packet is discarded.

4. *Encoding*

   When an intermediate node receives h encoded packets $u_i \in \mathbb{F}_p^{m+n}\ (1 \leq i \leq h)$, and they all are checked or considered to be correct, a forward coded packet along with new tags is

generated using the *Combine* $\left(\left(u_i, t_{u_i,1}, \cdots t_{u_i,l}, t^{'}_{u_i,1}, \cdots, t^{'}_{u_i,l'}\right)_{i=1}^{h}, (c_i)_{i=1}^{h}\right)$ algorithm with locally randomly generated coefficients c_i.

5.3.3.2 The Construction

In this subsection, we provide the construction of the scheme. Particularly, we present the above-mentioned four algorithms, namely *MAC, D-MAC, Combine* and *Verify* in detail. The algorithm *MAC* computes l tags for each data packet $\mathbf{u}_i$, and the algorithm *D-MAC* computes another $l^{'}$ tags for the tags. Moreover, the algorithm *Combine* generates new tags for a RLNC encoded packet. Furthermore, the algorithm *Verify* checks the correctness of a data packet with all tags. Details of these four algorithms are described as follows:

1. *MAC* $(\mathcal{K}_S, u_i)$:

   - **Input**: $\mathcal{K}_S$ consists of l secrete vectors $\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_l \in \mathbb{F}_p^{m+n+1}$, and the ith data packet $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$ as shown in Equation 5.1
   - **Output**: l tags $t_{\mathbf{u}_i,1}, t_{\mathbf{u}_i,2}, \cdots, t_{\mathbf{u}_i,l} \in \mathbb{F}_p$ for packet $\mathbf{u}_i$, where

$$t_{\mathbf{u}_i,l} = -\left(\sum_{j=1}^{m+n} u_{i,j} k_{l,j}\right) / k_{l,m+n+1} \tag{5.3}$$

2. *D-MAC* $\left(\mathcal{K}^{'}_S, t_{\mathbf{u}_i,1}, \cdots, t_{\mathbf{u}_i,l}\right)$

   - **Input**: $\mathcal{K}^{'}_S$ consists of $l^{'}$ secrete vectors $\mathbf{k}^{'}_1, \mathbf{k}^{'}_2, \cdots, \mathbf{k}^{'}_{l'} \in \mathbb{F}_p^{l+1}$, and the l' tags $t_{\mathbf{u}_i,l} \in \mathbb{F}_p$ generated in Equation 5.3 for the packet $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$
   - **Output**: l tags $t^{'}_{\mathbf{u}_i,1}, t^{'}_{\mathbf{u}_i,2}, \cdots, t^{'}_{\mathbf{u}_i,l'} \in \mathbb{F}_p$ for packet $\mathbf{u}_i$, where

$$t^{'}{}_{\mathbf{u}_i,l'} = -\left(\sum_{j=1}^{l} t_{\mathbf{u}_i,j} k^{'}_{l',j}\right) / k_{l',l+1} \tag{5.4}$$

3. *Combine* $\left(\left(\mathbf{u}_i, t_{\mathbf{u}_i,1}, \cdots t_{\mathbf{u}_i,l}, t^{'}_{\mathbf{u}_i,1}, \cdots, t^{'}_{\mathbf{u}_i,l'}\right)_{i=1}^{h}, (c_i)_{i=1}^{h}\right)$

   - **Input**: h vectors $\mathbf{u}_i \in \mathbb{F}_p^{m+n}\,(1 \le i \le h)$ as shown in Equation 5.2, each with $l + l^{'}$ tags $t_{u_i,1}, \cdots t_{u_i,l}, t^{'}_{u_i,1}, \cdots, t^{'}_{u_i,l'} \in \mathbb{F}_p\,(1 \le i \le h)$, and h constants $c_1, \cdots, c_h \in \mathbb{F}_p$
   - **Output**: a coded packet with $l + l^{'}$ tags generated by

$$\left(y, t_{y,1}, \cdots t_{y,l}, t^{'}_{y,1}, \cdots, t^{'}_{y,l'}\right) = \sum_{i=1}^{h} c_i \left(u_i, t_{u_i,1}, \cdots t_{u_i,l}, t^{'}_{u_i,1}, \cdots, t^{'}_{u_i,l'}\right) \tag{5.5}$$

4. *Verify* $\left(\mathcal{K}_{n_i}, \mathcal{K}^{'}_{n_i}, \left(y, t_{y,1}, \cdots t_{y,l}, t^{'}_{y,1}, \cdots, t^{'}_{y,l'}\right)\right)$

- **Input**: $\mathcal{K}_{n_i}$ consists of secrete vectors $\mathbf{k}_1, \cdots, \mathbf{k}_R \in \mathbb{F}_p^{m+n+1}, \{\mathbf{k}_1, \cdots, \mathbf{k}_R\} \subset \{\mathbf{k}_1, \cdots, \mathbf{k}_l\}$ and $\mathcal{K}'_{n_i}$ consists of secrete vectors $\mathbf{k}'_1, \cdots, \mathbf{k}'_{R'} \in \mathbb{F}_p^{l+1}, \left\{\mathbf{k}'_1, \cdots, \mathbf{k}'_{R'}\right\} \subset \left\{\mathbf{k}'_1, \mathbf{k}'_2, \cdots, \mathbf{k}'_{l'}\right\}$, as well as a RLNC encoded packet $y \in \mathbb{F}_p^{m+n}$ with $l + l'$ tags $t_{y,1}, \cdots t_{y,l}, t'_{y,1}, \cdots, t'_{y,l'} \in \mathbb{F}_p$
- **Output**: if

$$\forall \mathbf{k}_r \in \{\mathbf{k}_1, \cdots, \mathbf{k}_R\},\ \delta_r = \left(\sum_{j=1}^{m+n} y_j k_{r,j}\right) + t_{y,r} k_{r,m+n+1} = 0 \tag{5.6}$$

and

$$\forall \mathbf{k}'_{r'} \in \{\mathbf{k}'_1, \cdots, \mathbf{k}'_{R'}\},\ \delta'_{r'} = \left(\sum_{j=1}^{l} t_{y,j} k'_{r',j}\right) + t'_{y,r'} k'_{r',l+1} = 0 \tag{5.7}$$

then output 1; otherwise output 0.

5.3.3.3 The Correctness of Dual-HMAC

The proposed Dual-HMAC is going to be correct if the *Verify* algorithm could produce output 1 for the valid data packet and the valid RLNC encoded packet. Rigorously speaking, it should satisfy the following condition:

If $\textit{Verify}\left(\mathcal{K}_{n_i}, \mathcal{K}'_{n_i}, \left(\mathbf{u}_i, MAC(\mathcal{K}_S, \mathbf{u}_i), D - MAC(\mathcal{K}'_S, \mathbf{u}_i\right)\right) = 1$ and
If $\textit{Verify}\left(\mathcal{K}_{n_j}, \mathcal{K}'_{nj}, \left(\mathbf{u}_j, t_{\mathbf{u}_j,1}, \cdots t_{\mathbf{u}_j,l}, t'_{\mathbf{u}_j,1}, \cdots, t'_{\mathbf{u}_j,l'}\right)\right) = 1,\ \forall j = 1, \cdots, h$, then
$\textit{Verify}\left(\mathcal{K}_{n_j}, \mathcal{K}'_{n_j}, \textit{Combine}\left(\left(\mathbf{u}_j, t_{\mathbf{u}_j,1}, \cdots t_{\mathbf{u}_j,l}, t'_{\mathbf{u}_j,1}, \cdots, t'_{\mathbf{u}_j,l'}\right)_{j=1}^{h}, (c_j)_{j=1}^{h}\right)\right) = 1$

Theorem 4. *The first verification is correct.*

Proof. According to the description of "Setup", since $\mathcal{K}_{n_i} \subset \mathcal{K}_S$, $\forall \mathbf{k}_r \in \mathcal{K}_{n_i}$, there is a key $\mathbf{k}_l = \mathbf{k}_r$ and $l = r$. As a result, by letting $y = \mathbf{u}_i, \mathbf{k}_l = \mathbf{k}_r$, and substituting $t_{y,r} = t_{\mathbf{u}_i,l}$ using Equation 5.3, it is for sure that $\delta_r = 0, \forall \mathbf{k}_r \in \mathcal{K}_{n_i}$ in Equation 5.6.

Similarly, since $\mathcal{K}'_{n_i} \subset \mathcal{K}'_S$, $\forall \mathbf{k}'_{r'} \in \mathcal{K}'_{n_i}$, there is a key $\mathbf{k}_{l'} = \mathbf{k}_{r'}$ and $l' = r'$. As a result, by letting $y = \mathbf{u}_i, \mathbf{k}_{l'} = \mathbf{k}_{r'}$, and substituting $t'_{y,r'} = t'_{\mathbf{u}_i,l'}$ using Equation 5.4, it is for sure that $\delta'_{r'} = 0, \forall \mathbf{k}'_{r'} \in \mathcal{K}'_{n_i}$ in Equation 5.7.

As a result, $\textit{Verify}\left(\mathcal{K}_{n_i}, \mathcal{K}'_{n_i}, \left(\mathbf{u}_i, MAC(\mathcal{K}_S, \mathbf{u}_i), D - MAC(\mathcal{K}'_S, \mathbf{u}_i\right)\right) = 1$

□

Theorem 5. *The second verification is correct.*

Proof. According to Equation 5.3, the tags are created in a way that $\mathbf{k}_l$ is orthogonal to the concatenation of $\mathbf{u}_i$ and its l tags $\mathbf{k}_l \perp (\mathbf{u}_i \parallel t_{\mathbf{u}_i,l})$. This relationship is true for all packets in the same generation, which represents as $\mathbf{k}_l \perp Span\left\{(\mathbf{u}_i \parallel t_{\mathbf{u}_i,l})_{i=1}^{m}\right\}$. If $\textit{Verify}\left(\mathcal{K}_{n_j}, \mathcal{K}'_{nj}, \left(\mathbf{u}_j, t_{\mathbf{u}_j,1}, \cdots t_{\mathbf{u}_j,L}, t'_{\mathbf{u}_j,1}, \cdots, t'_{\mathbf{u}_j,L'}\right)\right) = 1,\ \forall j = 1, \cdots, h$, then $\mathbf{k}_l \perp Span\left\{(\mathbf{u}_j \parallel t_{\mathbf{u}_j,l})_{j=1}^{h}\right\}$.

Moreover, the algorithm *Combine* is a linear combination, which does not change the subspace. Hence, $(y \parallel t_{y,l}) \in Span\left\{(\mathbf{u}_j \parallel t_{\mathbf{u}_j,l})_{j=1}^{h}\right\}$, $\mathbf{k}_l \perp (y \parallel t_{y,l})$, and the Equation 5.6 $\delta_r = 0, \forall \mathbf{k}_r \in \mathcal{K}_{n_i}$. Similarly, we could derive $\delta_{r'}^{'} = 0, \forall \mathbf{k}_{r'}^{'} \in \mathcal{K}_{n_i}^{'}$. As a result, the output of *Verify* is 1, and the second condition is also correct. □

5.3.3.4 Our Key Distribution Model

In this thesis, we assume that a set of symmetric keys are distributed to all participant nodes in a secure and authenticated manner through a key distribution scheme. In this section, we provide the model of the key distribution scheme that we have adopted. A set system is a pair (X, F), where X is a finite set of elements and F is an ordered set of subsets of X.

Definition 1. A set system (X, F) is called a c cover-free family (c-CFF) if for any c sets $A_1, A_2, ...A_c \in F$ and any other set $B \in F$, we have

$$B \not\subset \bigcup_{J=1}^{c} A_i \tag{5.8}$$

Definition 2. A set system (X, F) is called a (c, d) cover-free family ((c, d)-CFF) if for any c sets $A_1, A_2, ...A_c \in F$ and any other set $B \in F$, we have

$$\left|B \backslash \bigcup_{J=1}^{c} A_j\right| > d \tag{5.9}$$

We believe that if key assignment is done properly, no coalition of c nodes can fool another node. Our key pre-distribution scheme considers two key pools which are available at the source node. We use all these two sets for checking the message's integrity. It is shown in [133] that in order to resist against less than c compromised nodes, the cardinality of the key size at the source must be $(c+1)$ times larger than those key size at the relay/sink nodes, which could be expressed as $l \geq (c+1)R$, and $l^{'} \geq (c+1)R^{'}$. More specifically, we claim that by this model, we can be sure each verifier has some shared keys which can verify the integrity of messages. Our scheme assigns two sets $\mathcal{K}_S = \{\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_l\}, \mathbf{k}_i \in \mathbb{F}_p^{m+n+1}$, $\mathcal{K}_S^{'} = \left\{\mathbf{k}_1^{'}, \mathbf{k}_2^{'}, \cdots, \mathbf{k}_{l'}^{'}\right\}, \mathbf{k}_{i'}^{'} \in \mathbb{F}_p^{l+1}$ to a source node which each has $l = l^{'} = |\mathcal{K}_S| = \left|\mathcal{K}_S^{'}\right| = \frac{1}{2(1-\delta)} e(c+1)ln(1/q)$ keys. Moreover, we distribute two sets $\mathcal{K}_{n_i} = \{\mathbf{k}_1, \cdots, \mathbf{k}_R\}, \mathbf{k}_r \in \mathbb{F}_p^{m+n+1}$, and $\mathcal{K}_{n_i}^{'} = \left\{\mathbf{k}_1^{'}, \cdots, \mathbf{k}_{R'}^{'}\right\}, \mathbf{k}_{r'}^{'} \in \mathbb{F}_p^{l+1}$ to each node which each of these sets has $R = R^{'} = |\mathcal{K}_{n_i}| = \left|\mathcal{K}_{n_i}^{'}\right| = \frac{1}{2(1-\delta)} eln(1/q)$ keys. In addition, q is the security parameter and it stands as $q = (N.(Nc))^{-1}$ [133], where N is the total number of recipients (usually $q = 10^{-3}$ would be sufficient) and c is a fixed number of corrupted users. Every key $\mathbf{k}_r$ or $\mathbf{k}_{r'}^{'}$ is included in a set of each verifier with probability of $1/2(c+1)$ (for simplicity, we consider the same length for both poll keys).

5.3.4 Security Analysis

In this section, we analyse the possibilities of an adversary launching pollution attack successfully. Hence, three possible pollution attack behaviours are analysed, and their security levels are quantified respectively. Finally, the possible knowledge that an adversary can access is analysed.

1. *Resistance Against Pollution Attack*
   We consider the following three possible pollution attack scenarios:

- 1st Pollution Behaviour: Data Pollution Attack
  An adversary node intends to make changes in a message. This polluted message can be detected immediately by the next hop; or in a few hops later (maximum $c-1$ hops later). It means the polluted message should travel some hops to be detected. The worst case happens whenever an adversary can do coalition with the other $c-1$ compromised nodes.

- 2nd Pollution Behaviour: Tag (i.e., MAC) Pollution Attack
  An adversary node intends to make changes in the MACs. This attack can be detected immediately by the next hop if he cannot compromise the next hop and get the next hop keys; or in a few hops later (maximum $c-1$ hops later). It is similar to the 1st pollution behaviour.

- 3rd Pollution Behaviour: Dual-Tag (i.e., D-MAC) Pollution Attack
  An adversary i, which is one of the N legitimate nodes ($N-i$ is the maximum number of hops between the adversary and the sink node) intends to make changes in D-MACs. This attack can be detected by the next hop by the probability of $\frac{1}{2(c+1)}$. The polluted D-MAC can travel some hops, before it is detected, by the probability of $\left[1-\frac{1}{2(c+1)}\right]^{N-i}$ (see Theorem 6).

  Theorem 6. *The probability that a downstream user receives the same key as key_x of user x is not less than $\left[1-\frac{1}{2(c+1)}\right]^{N-i}$.*

  Proof. The probability of recovering the same shared key between two nodes is $\frac{1}{2(c+1)}$, so the probability of the key_x from a user x is located in next hop $x+1$ is $\frac{1}{2(c+1)}$, and this probability would be $\left(\frac{1}{2(c+1)}\right)^2$ if this key finds in two hops later. In the other words, if the total number of hops between an intermediate node and a sink node, named as downstream node, is $N-i$, then the probability that a downstream user receives the same key as key_x of user x is not less than $\left[1-\frac{1}{2(c+1)}\right]^{N-i}$. □

2. *The Possible Knowledge of an Adversary*
   As mentioned earlier, we assume the source node is trustworthy and the process of key pre-distribution is secure. Hence, the secret key sets, $\mathcal{K}_S$ and $\mathcal{K}_S^{'}$, assigned to the source node are considered secure. However, an adversary can wiretap all the data communication in a network and may compromise several relay or sink nodes. Hence, the adversary can get access to the received packets from the previous hops, the key information distributed to him by the KDC and the key information stored at the compromised nodes.

5.3.5 Performance Evaluation

In this section, communication and computation overhead of the proposed Dual-HMAC scheme are presented.

5.3.5.1 Communication Overhead

In our scheme, each source message has $m+n$ codewords and each codeword has $\log_2 p$ bits. So, the bit-length of a source message is $(m+n)\log_2 p$. The L MACs and the $L^{'}$ D-MACs are appended to the source message. We consider $\log_2 p$ symbols for each MAC and D-MAC. By fixing $L+L^{'}=l$, our scheme has the following communication overhead:

$$\frac{(L+L^{'})\log_2 p}{(m+n)\log_2 p} = \frac{L+L^{'}}{m+n} = \frac{l}{m+n} \tag{5.10}$$

Hence, the Dual-HMAC scheme's overhead is equal to the scheme presented in [25]. However this scheme is vulnerable to tag pollution. Additionally, the bandwidth overhead between our scheme and the scheme proposed in [3] is given by the following equation:

$$\frac{O_{b_{Dual-HMAC}}}{O_{b_{MacSig}}} = \frac{\frac{l}{m+n}}{\frac{l+1}{m+n} + \frac{32l}{128(m+n)}} = \frac{1}{1.25 + \frac{1}{l}} \tag{5.11}$$

Base on equation 5.11, the bandwidth overhead comparison between our scheme and the scheme described in [3] , for different values of c, are depicted in the graphs in the following Figures 5.4, 5.5, and 5.6 respectively.

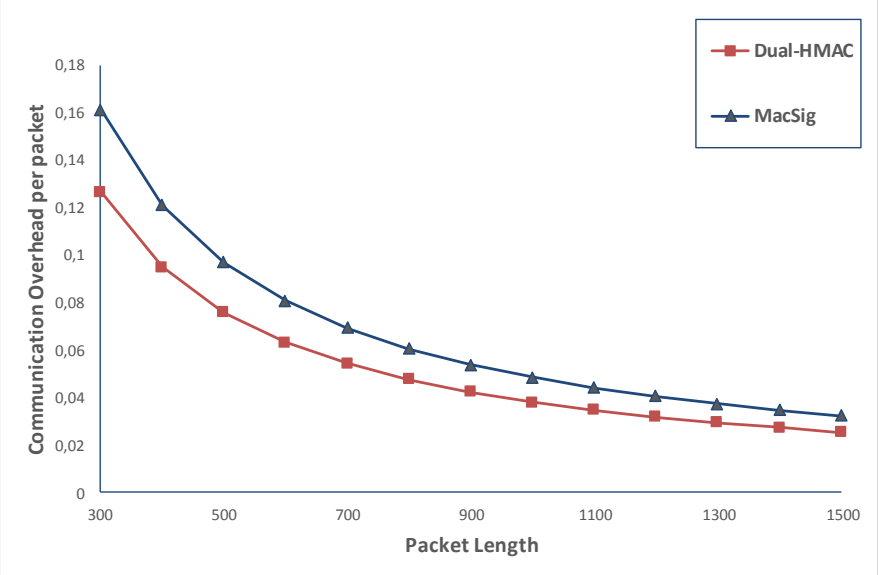

Figure 5.4: The bandwidth overhead comparison ($\delta = 0.1, n = 20m, \epsilon = 0.01$, and $c = 1$ in [3]).

5.3.5.2 Computation Overhead

Recall that to append L MACs and $L^{'}$ D-MACs, $L(m+n+1)$ and $L^{'}(L+1)$ multiplication operations for MACs and D-MACs are needed, respectively. In addition, for verification, we

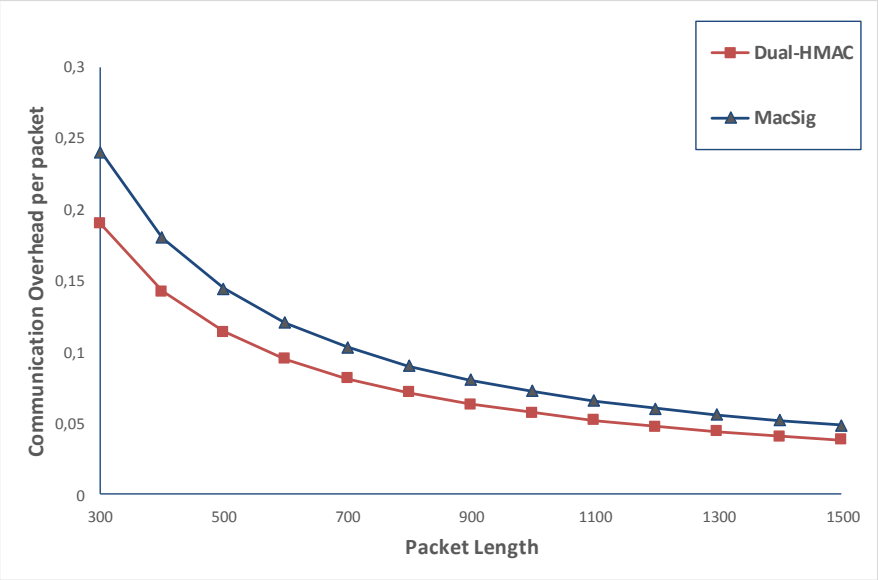

Figure 5.5: The bandwidth overhead comparison ($\delta = 0.1, n = 20m, \epsilon = 0.01$, and $c = 2$ in [3]).

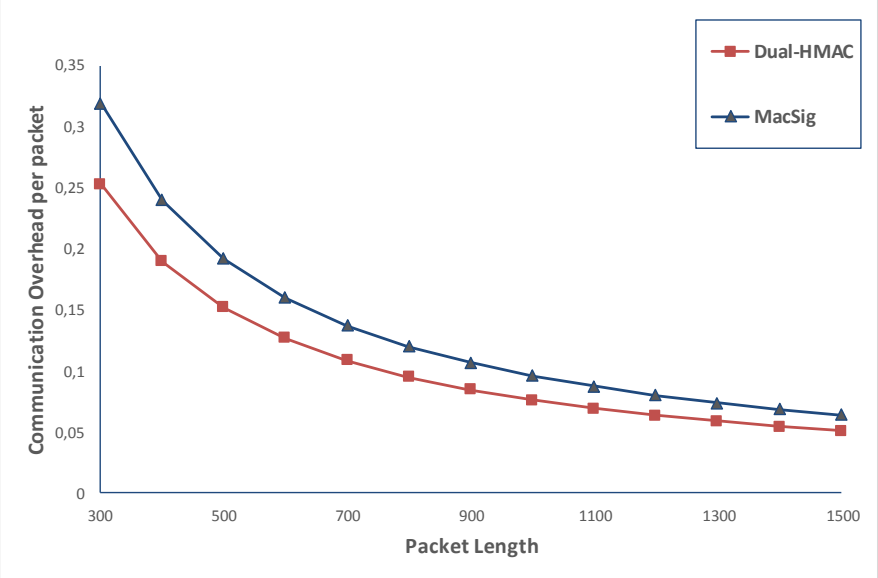

Figure 5.6: The bandwidth overhead comparison ($\delta = 0.1, n = 20m, \epsilon = 0.01$, and $c = 3$ in [3]).

consider that all the relay nodes have shared keys and each node needs to verify the message. So, as shown in equation 5.6 and 5.7, our scheme needs $L(m+n+1)$ multiplications for MACs and $L^{'}(L+1)$ multiplications for D-MACs.

5.3.6 Conclusion

This section has presented a homomorphic MAC-based scheme, called Dual-HMAC, for network coding-enabled WNs. The proposed scheme makes use of two types of tags to provide resistance against data pollution attacks and partially tag pollution attacks. Furthermore, our proposed scheme presents low communication overhead and low computational overhead compared to other existing schemes.

To conclude this section, consider the following.

- Dual-HMAC incurs a relatively low communication overhead which is almost equal to [25]; however, our scheme's overhead, according to 5.11, saves up to $(1.25 + 1/l)$ in comparing to MacSig approach [3]. This means that our scheme provides at least 25 % bandwidth saving.

- Dual-HMAC not only slightly increases the tag pollution resistance, but also decreases the computation overhead compared to the schemes in [3, 25].

- Dual-HMAC provides resistance for half of the tags against tag pollution attack. This results in reducing the possibility of tag pollution around 50% compared to the work presented in [25].

5.4 An Efficient Dual Homomorphic MAC Scheme

5.4.1 Proposed Homomorphic Message Authentication Code-based (HMAC) scheme

Our proposed scheme makes use of two types of homomorphic MACs, called *MACs* and *D-MACs*, and one homomorphic signature, called *Sign*, which are appended to the end of the coded packet. Particularly, each MAC ensures integrity of the content of the coded packet and each D-MAC ensures integrity of the MACs of the coded packet as it is also described in the previous section (5.3). In addition to MACs and D-MACs, our proposed scheme leverages a homomorphic signature to ensure integrity of D-MACs and achieve resistance against both data pollution and tag pollution attacks.

Our proposed HMAC scheme consists of four steps, as it is depicted in Figure 5.7. In step 1, the source generates the coded packets based on the native packets and by using the equation 5.1. Then, in step 2, the required MACs and D-MACs as well as the required Sign are created based on the equations 5.12, 5.13 and 5.14 respectively. Afterwards, in step 3, the intermediate nodes recode the received packets according to the equation 5.2. Finally, in step 4, the sink nodes decode the received packets and obtain the native packets.

The construction of our proposed scheme is based on the following four basic algorithms: *MAC*, *Sign*, *Combine*, and *Verify*. The definitions of these algorithms as well as the construction of HMAC are given below in sections 5.4.2 and 5.4.3, respectively. Finally, our proposed key distribution model for HMAC is presented in section 5.4.4.

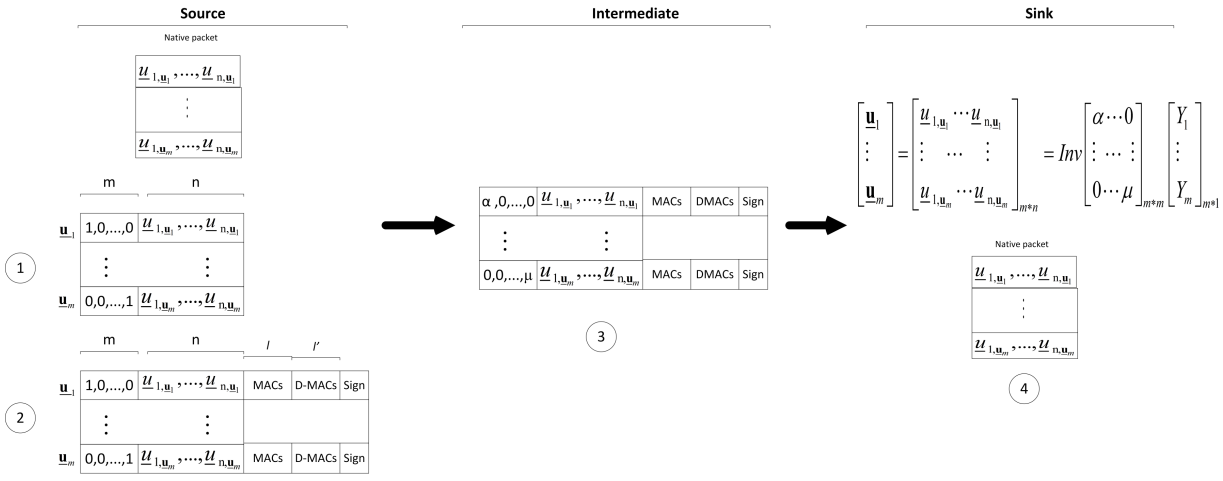

Figure 5.7: The four main steps of HMAC scheme.

5.4.2 The basic algorithms of HMAC

In HMAC, the source node makes use of the *MAC* and *Sign* algorithms. Moreover, the intermediate nodes perform the *Combine* and *Verify* algorithms, and the sink nodes perform the *Verify* algorithm.

- *MAC*
  Input:

  Two sets of key vectors: $\{\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l\}, \mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{\mathrm{m+n+1}}$, and $\{\mathcal{K}_1^{'}, \mathcal{K}_2^{'}, ..., \mathcal{K}_{l'}^{'}\}, \mathcal{K}_i^{'} \in \mathbb{F}_{\mathrm{p}}^{l+1}$. We consider l and $l^{'}$ as the number of MACs and D-MACs respectively.

  Output: Two sets of tags: l MACs $t_1, t_2, ..., t_l \in \mathbb{F}_{\mathrm{p}}$ and $l^{'}$ D-MACs $t_1^{'}, t_2^{'}, ..., t_{l'}^{'} \in \mathbb{F}_{\mathrm{p}}$.

- *Sign*
  Input: Select a generator g from a multiplicative cyclic group $\mathbb{G}$, and sample a private key vector $Pr_K = (\beta_1, \beta_2, ..., \beta_{l'+1})$, where $\beta_i \in \mathbb{F}_{\mathrm{p}}$, and compute a public key $PU_K = (g^{\beta_1}, g^{\beta_2}, ..., g^{\beta_{l'+1}})$.

  Output: A signature (i.e., $Sign \in \mathbb{F}_{\mathrm{p}}$)

- *Combine*
  Input: h random numbers from $\mathbb{F}_{\mathrm{p}}$ and h vectors $y_i \in F_{\mathrm{p}}^{\mathrm{m+n}} (1 \leq i \leq h)$. Each of these vectors y_i includes: a) l MACs $t_{y_i,1}, t_{y_i,2}, ..., t_{y_i,l} \in \mathbb{F}_{\mathrm{p}} (1 \leq i \leq h)$, b) $l^{'}$ D-MACs $t_{y_i,1}^{'}, t_{y_i,2}^{'}, ..., t_{y_i,l'}^{'} \in \mathbb{F}_{\mathrm{p}} (1 \leq i \leq h)$, and c) one signature $Sign_{y_i} \in \mathbb{F}_{\mathrm{p}} (1 \leq i \leq h)$.

  Output: A vector $Y \in \mathbb{F}_{\mathrm{p}}^{m+n+l+l^{'}+1}$.

- *Verify*
  Input: Two subsets of key vectors of l and $l^{'}$, the public key PU_K, and $Y \in \mathbb{F}_{\mathrm{p}}^{m+n+l+l^{'}+1}$.

  Output: Either 1 (accept), or 0 (reject).

5.4.3 The Construction of HMAC

- ***Setup***.

  The KDC distributes two sets of keys $\{\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l\}$, where $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{\mathrm{m+n+1}}$, and $\{\mathcal{K}_1^{'}, \mathcal{K}_2^{'}, ..., \mathcal{K}_{l'}^{'}\}$, where $\mathcal{K}_i^{'} \in \mathbb{F}_{\mathrm{p}}^{l+1}$, as well as a private key Pr_K to the source node. Moreover, the KDC shares the public key and two subsets of $\mathcal{K}_i$ and $\mathcal{K}_j^{'}$ to each intermediate and sink node where $\mathcal{K}_i \in \mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l$, and $\mathcal{K}_j^{'} \in \mathcal{K}_1^{'}, \mathcal{K}_2^{'}, ..., \mathcal{K}_{l'}^{'}$.

- ***Tag Generation***. Two sets of tags: l MACs $t_1, t_2, ..., t_l \in \mathbb{F}_{\mathrm{p}}$ and $l^{'}$ D-MACs $t_1^{'}, t_2^{'}, ..., t_{l'}^{'} \in \mathbb{F}_{\mathrm{p}}$ are calculated according to the following equations:

$$t_i = -\frac{\sum_{j=1}^{m+n} \mathcal{K}_{i,j}\mathbf{u}_j}{\mathcal{K}_{i,m+n+1}}, (i = 1, ..., l) \tag{5.12}$$

$$t_{i'}^{'} = -\frac{\sum_{r=1}^{l} \mathcal{K}_{i',r}^{'} t_r}{\mathcal{K}_{i',l+1}^{'}}, (i^{'} = 1, ..., l^{'}) \tag{5.13}$$

- ***Sign***. Based on the private key and D-MACs, a signature is calculated as:

$$Sign = -\frac{\sum_{i=1}^{l^{'}} \beta_i t_i^{'}}{\beta_{l'+1}} \tag{5.14}$$

- ***Combine.*** h random numbers from $\mathbb{F}_{\mathrm{p}}$ and h vectors $y_i \in F_{\mathrm{p}}^{\mathrm{m+n}} (1 \leq i \leq h)$. Each of these vectors y_i includes l MACs, $l^{'}$ D-MACs, and the signature $Sign_{y_i}$. Output is a linear combination and presented as a vector $Y \in \mathbb{F}_{\mathrm{p}}^{m+n+l+l^{'}+1}$.

- ***Verify.*** Based on the two vectors of keys $\mathcal{K}_i$ and $\mathcal{K}_j^{'}$, as well as the public key PU_K, an intermediate or destination node checks the correctness of the received coded packets according to the following equations:

$$\delta_i = (\sum_{j=1}^{m+n} \mathcal{K}_{i,j} Y_{i,j}) + \mathcal{K}_{i,m+n+1} t_i \qquad i = 1, ..., l \tag{5.15}$$

$$\delta_{i'}^{'} = (\sum_{j=1}^{l} \mathcal{K}_{i',j}^{'} t_{i',j}) + \mathcal{K}_{i',l+1}^{'} t_{i'}^{'} \qquad i^{'} = 1, ..., l^{'} \tag{5.16}$$

$$\eta = (\prod_{i=1}^{l^{'}} g^{\beta_i t_i^{'}}) . g^{\beta_{l'+1}{}^{Sign}} \tag{5.17}$$

  If $\delta_i = 0$, for at least one specific i, and $\delta_i^{'} = 0$, for at least one specific $i^{'}$, and $\eta = 1$, then the output of the verification procedure is equal to 1 (i.e. the packet has passed the verification).

  On the other hand, the packet is discarded if any of the above equations are not satisfied. We describe the verification procedure in Algorithm 2.

Algorithm 2: Verification Procedure

Input:

- A packet $Y \in \mathbb{F}_{\mathrm{p}}^{m+n}$,
- A set of key vectors $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{m+n+1}$, where $\mathcal{K}_i \in \mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l$,
- A set of key vectors $\mathcal{K}_j^{'} \in \mathbb{F}_{\mathrm{p}}^{l+1}$, where $\mathcal{K}_j^{'} \in \mathcal{K}_1^{'}, \mathcal{K}_2^{'}, ..., \mathcal{K}_{l'}^{'}$,
- public key PU_K,
- l MACs where $t_i \in \mathbb{F}_{\mathrm{p}}^{l}$,
- $l^{'}$ D-MACs where $t_i^{'} \in \mathbb{F}_{\mathrm{p}}^{l^{'}}$,
- a signature $Sign \in \mathbb{F}_{\mathrm{p}}$

;
Output: 0 or 1 ; /* Either 1 (accept), or 0 (reject) */
;
main()
foreach *node* **do**
  foreach $MAC(i.e., t_i)$ **do**
    ; /* Checking the correctness of the MACs */
    $count_1 = 0$;
    foreach $\mathcal{K}_i$ **do**
      if $\delta_i = (\sum_{j=1}^{m+n} \mathcal{K}_{i,j} Y_{i,j}) + \mathcal{K}_{i,m+n+1} t_r = 0$ **then**
        $count_1 + +$,;
        break; ;
    if $count_1 \neq 0$ **then**
      break ;;
  if $count_1 = 0$ **then**
    $output = 0$;;
    exit; ;
    else
      foreach D-MAC (i.e., $t_i^{'}$) **do**
        ; /* Checking the correctness of the D-MACs */
        $count_2 = 0$;
        foreach $\mathcal{K}_{i}^{'}$ **do**
          if $\delta_i^{'} = (\sum_{j=1}^{l} \mathcal{K}_{i,j}^{'} t_{i,j}) + \mathcal{K}_{i,l+1}^{'} t_r^{'} = 0$ **then**
            $count_2 + +$,;
            break; ;
      if $count_2 = 0$ **then**
        $output = 0$;;
        exit; ;
      else
        ; /* Checking the correctness of the signature */
        if $\eta = (\prod_{i=1}^{l^{'}} g^{\beta_i t_i^{'}}) . g^{\beta_{l'+1}{}^{Sign}} = 1$ **then**
          $output = 1$; ; /* accept and pass the packet to the next hop */
        else
          $output = 0$; ; /* (reject the packet) */

5.4.4 The Key Distribution Model of HMAC

According to the key distribution model of HMAC, the KDC assigns to the source l key vectors of the set $\{\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l\}, \mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{\mathrm{m+n+1}}$, and $l^{'}$ key vectors of the set $\{\mathcal{K}_1^{'}, \mathcal{K}_2^{'}, ..., \mathcal{K}_{l'}^{'}\}, \mathcal{K}_i^{'} \in \mathbb{F}_{\mathrm{p}}^{l+1}$. We calculate the l and $l^{'}$ based on the following formula:

$$l = l^{'} = L/2 = \frac{1}{2(1-\delta)} e(c+1)ln(1/q) \tag{5.18}$$

Where L is given by the following equation which is provided in [3]:

$$L = l + l^{'} = \frac{1}{1-\delta} e(c+1)\ln(1/q) \tag{5.19}$$

δ and q are security parameters and c is the number of compromised nodes as they are defined in [3].

Furthermore, according to our key distribution model, which is based on the cover free set systems proposed in [25] and [133], two key vectors are distributed to each intermediate node. The one key vector $\{\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l\}$, $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{\mathrm{m+n+1}}$ is used for MAC verification, based on Equation 5.15, and the other key vector $\{\mathcal{K}_1^{'}, \mathcal{K}_2^{'}, ..., \mathcal{K}_{l'}^{'}\}$, $\mathcal{K}_i^{'} \in \mathbb{F}_{\mathrm{p}}^{l+1}$ is used for D-MAC verification, based on Equation 5.16. According to our key distribution model, when the key assignment has been completed, resistance against c compromised nodes can be achieved. It is worthwhile to mention that our key distribution model takes into consideration that the number of keys at the source should be c times more than the number of the keys assigned to each intermediate or sink node in order to provide resistance against c compromised nodes.

In addition, we have considered that each node, in our key distribution model, has only one assigned key, since one key is enough to check the correctness of each coded packet and also the number of keys that the compromised nodes can access is minimized.

5.4.5 Security Analysis of HMAC

In this section, the possibilities of an adversary lunching a data pollution attack or a tag pollution attack successfully are discussed. We consider the source S sends m coded packets in each generation and each coded packet includes l MACs, $l^{'}$ D-MACs, and one signature.

- **Data Pollution** The following Definition 1 gives the essence of data pollution attacks in HMAC.

  Definition 1. *A corrupted packet (i.e.,$X^{'} \in \mathbb{F}_{\mathrm{p}}^{\mathrm{m+n}}$) is called as a polluted packet if the following conditions are satisfied:*

  1. $X^{'} \notin Span\,\{\underline{\mathbf{U}}_{\mathbf{1}}, ..., \underline{\mathbf{U}}_{\mathbf{m}}\}$
  2. *If an adversary is able to satisfy the following equation for at least one key vector of the set* $\{\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l\}$, $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{\mathrm{m+n+1}}$*:*

$$(\sum_{j=1}^{m+n} \mathcal{K}_{i,j} X_j^{'}) + \mathcal{K}_{i,m+n+1} t_i = 0 \qquad i = 1, ..., l \tag{5.20}$$

5.4.5.1 The adversary injects a polluted packet $X^{'}$

The adversary aims to modify a legitimate packet in order to create a polluted packet $X^{'}$ with the same MACs, the D-MACs and signature which satisfy the following equations:

$$(\sum_{j=1}^{m+n} \mathcal{K}_{i,j} X_j^{'}) + \mathcal{K}_{i,m+n+1} t_i = 0 \qquad i = 1, ..., l \tag{5.21}$$

$$(\sum_{j=1}^{l} \mathcal{K}_{i^{'},j}^{'} t_j) + \mathcal{K}_{i^{'},l+1}^{'} t_{i^{'}}^{'} = 0 \qquad i^{'} = 1, ..., l^{'} \tag{5.22}$$

$$(\prod_{i=1}^{l^{'}} \beta_i^{t_i^{'}}).\, {\beta_{l^{'}+1}}^{Sign} = 1 \tag{5.23}$$

However, without making any changes in the MACs, the D-MACs and the signature, the adversary only needs to satisfy the equation 5.21 in order to lunch a data pollution attack successfully.

Theorem 7. *Without making any changes in the MACs, the D-MACs, and the signature, the probability that the adversary can pass the verification of the polluted packet (i.e., $X^{'}$) through the equation 5.21, is not greater than $1/p$.*

Proof. We assume that the adversary tries to create $X^{'}$ which can pass the equation 5.21. For this reason, the adversary needs to change at least two symbols of the legitimate packets' content data to create a polluted one. By considering that the adversary does not change the MACs, the D-MACs or the signature, he needs to change only one symbol of the packet's content data and find another one in order to satisfy the equation 5.21. Thus, the probability that the polluted packet passes the verification is equal to find a symbol from the finite field $\mathbb{F}_p$. This probability is equal to $1/p$.

□

Theorem 8. *Without making any changes in the MACs, the D-MACs, and the signature, and by considering that there are d neighbor nodes for the adversary, the probability that the polluted packet (i.e., $X^{'}$) can pass the verification of the neighbor nodes depends on which keys they have and is not greater than $1/l^d$.*

Proof. According to the Theorem 7, the polluted packet has passed the verification through the equation 5.21, so that the probability that the next hops will consider the polluted packet as a legitimate one depends on the keys that they have. Particularly, the next hops will consider the polluted packet as a legitimate one only if they have the same key with the compromised node. However, for d nodes, it happens with the probability of $1/l^d$[1].

□

[1]For instance, if we use the settings in [3], and consider $\delta = 0.1, c = 1, 2$, and 3, where c is the number of compromised nodes, this probability in case of only two nodes (the adversary and its neighbor) is not greater than $0.005, 0.003$ and 0.001, respectively.

- **Tag Pollution**

  The following Definitions 2 and 3 give the essence of tag (i.e. MAC and D-MAC) pollution attacks in HMAC.

  Definition 2. *A corrupted MAC (i.e. $T \in \mathbb{F}_{\mathrm{p}}^{l}$) is called as a polluted MAC if the following condition is satisfied:*

  1. *If an adversary is able to satisfy the following equation for at least one key vector of the set $\{\mathcal{K}_1^{'}, \mathcal{K}_2^{'}, ..., \mathcal{K}_{l'}^{'}\}$, $\mathcal{K}_i^{'} \in \mathbb{F}_{\mathrm{p}}^{l+1}$:*

$$(\sum_{j=1}^{l} \mathcal{K}_{j,i'}^{'} T_{j,i'}) + \mathcal{K}_{l+1,i'}^{'} t_{i'}^{'} = 0 \qquad i^{'} = 1, ..., l^{'} \tag{5.24}$$

  Definition 3. *A corrupted D-MAC (i.e. $T^{'} \in \mathbb{F}_{\mathrm{p}}^{l^{'}}$) is called as a polluted D-MAC if the following condition is satisfied:*

  1. *If an adversary is able to satisfy the following equation:*

$$(\prod_{i=1}^{l^{'}} \beta_i^{T_i^{'}}).\, \beta_{l'+1}{}^{Sign} = 1 \tag{5.25}$$

  First of all, the adversary has to create a new signature (i.e. $Sign^{'}$) for the packet in which he intends to modify any of its MACs or D-MACs in order to create a polluted packet that will pass the verification of equations 5.24 and 5.25.

  Theorem 9. *If the adversary modifies any of the MACs or the D-MACs of a packet that he intends to pollute, the probability that the polluted packet will pass the verification through the equation 5.25 is similar to the probability of solving a Discrete Logarithm Problem, which is negligible.*

  Proof. We assume that the adversary has to calculate a new signature $Sign^{'}$, which can satisfy the equation 5.25, for the polluted MACs and D-MACs. For this reason, the adversary derives $Sign^{'}$ from the equation 5.25. It is similar to solve a Discrete Logarithm Problem, and the probability of success is equal to $\frac{1}{p} = \frac{1}{2^8}$, which is negligible, because HMAC is defined over the finite field $\mathbb{F}_{\mathrm{p}}$ with $\mathrm{p} = 2^8$.

  □

  Theorem 10. *If the adversary modifies the signature of a packet that he intends to pollute, the probability of the polluted packet to pass the verification through the Equation 5.14, on the next node, is 0.*

  Proof. According to our key distribution model, described in Section 5.4.4, KDC distributes the public key $PU_K = (g^{\beta_1}, g^{\beta_2}, ..., g^{\beta_{l'+1}})$ to all nodes. Thus, if the adversary modifies the signature (i.e., sign'), the probability that the next node will not verify, based on Equation 5.14, the modified signature is 0, since it holds the public key $PU_K = (g^{\beta_1}, g^{\beta_2}, ..., g^{\beta_{l'+1}})$.

  □

Then, the adversary, who has already verified the polluted packet, intends to pass the verification of the neighbor nodes. However, it can be achieved when the adversary and the neighbor nodes have the same key vectors (i.e. one key vector for verifying the MACs and one key vector for verifying the D-MACs).

Theorem 11. *The probability that the adversary has the same key vectors with d neighbor nodes is not greater than* $\frac{1}{(l * l')^d}$.

Proof. According to our key distribution model, KDC assigns one set of l key vectors for MAC verification and one set of $l^{'}$ key vectors for D-MAC verification to each node. Thus, the probability that d nodes have the same set of keys with the adversary is almost $\frac{1}{(l * l')^d}$, where $l = l^{'} = \frac{L}{2}$. □

5.4.6 Performance Evaluation

In this section, we analyze the performance of our proposed HMAC scheme in terms of computational complexity, communication overhead, and key storage overhead. We follow the settings defined in [3] and thus we set $p = 2^8, n = 1024$, and $m = 32$. We compare HMAC with the MacSig scheme, presented in [3], which is the most well-known tag pollution immune scheme taking advantage of homomorphic MACs and one signature.

5.4.6.1 Computational Complexity

The computational complexity of HMAC is considered for the source and non-source nodes (i.e. intermediate and sink nodes). For the source node, we calculate the complexity of tags' generation (i.e., MACs and D-MACs) and signatures' generation. For the other nodes, we calculate the time complexity of the verification algorithm of HMAC.

- ***At source node:***
  The number of MACs and D-MACs which are created at the source node is equal to $L = l + l^{'}$, where l is the number of MACs and $l^{'}$ is the number of D-MACs. However, the source node has to calculate $m + n + 1$ and $l + 1$ multiplications for generating each MAC and D-MAC, respectively. Moreover, the source node has to calculate a signature which requires $l^{'} + 1$ multiplications to be generated.

- ***At Non-source nodes:***
  At each node, the following multiplications and exponentiations are required for verification of a packet: a) $l * (m + n + 1)$ multiplications for MAC verification, b) $l^{'} * (l + 1)$ multiplications for D-MAC verification, and c) $l^{'} + 1$ exponentiations for signature verification.

In terms of exponentiations, we have considered that each exponentiation is equivalent to $\frac{3}{2}|p|$ multiplications as it is shown in [3]. Figure 5.8 represents the number of multiplications which are required for verification of a packet at each non-source node for three different numbers of compromised nodes (i.e. $c = 1, 2$ and 3). Our demonstration shows the efficiency of HMAC compared to MacSig scheme which is proposed in [3]. HMAC is almost 10 times faster than it.

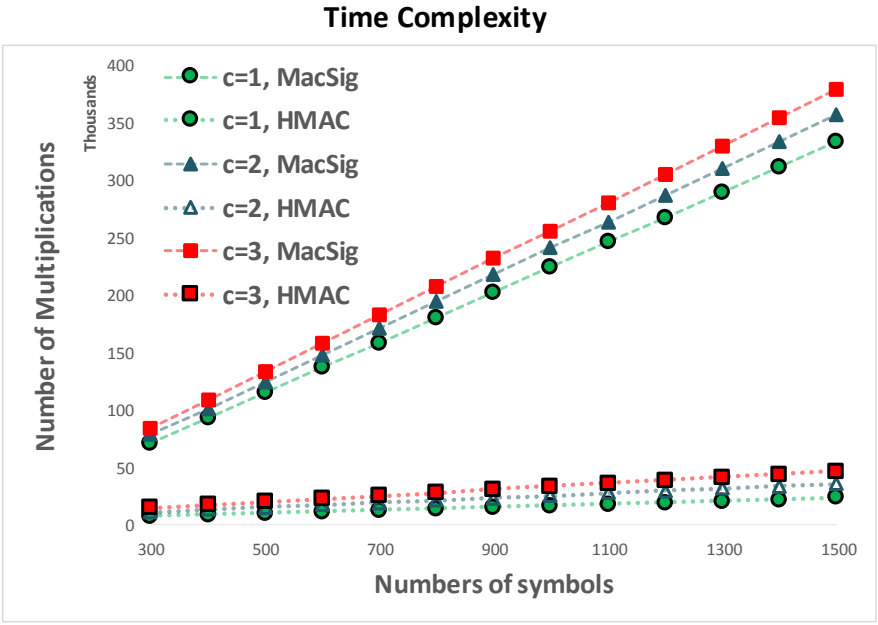

Figure 5.8: The number of multiplications required for verification of a coded packet at each non-source node for three different numbers of compromised nodes.

5.4.7 Communication Overhead

To calculate the communication overhead of HMAC, we take into consideration the number of symbols of the MACs, the D-MACs and the signature appended to the end of each coded packet. The total number of symbols of the MACs and the D-MACs is given by the equation 5.19. However, the signature adds one more symbol and thus the total number of symbols of each coded packet is given by the following equation:

$$T_{symbols} = \frac{1}{1-\delta} e(c+1)\ln(q) + 1 \tag{5.26}$$

In other words, the communication overhead of HMAC is $T_{symbols} * \lceil \log_2 p \rceil$ bits.

Compared to MacSig scheme in [3], HMAC achieves 25% less communication overhead. Our key distribution model allows HMAC to achieve this considerable gain in terms of communication overhead. Finally, we depict the HMAC communication overhead in Figure 5.9.

5.4.8 Key storage Overhead

The source node needs to store the l key vectors for generating the MACs and the $l^{'}$ key vectors for generating the D-MACs of each coded packet. The size of each key vector for generating the MACs is $m+n+1$ and the size of each key vector for generating the D-MACs is $l+1$. Also the source node has to store the private key vector, with size $l^{'}+1$, for generating

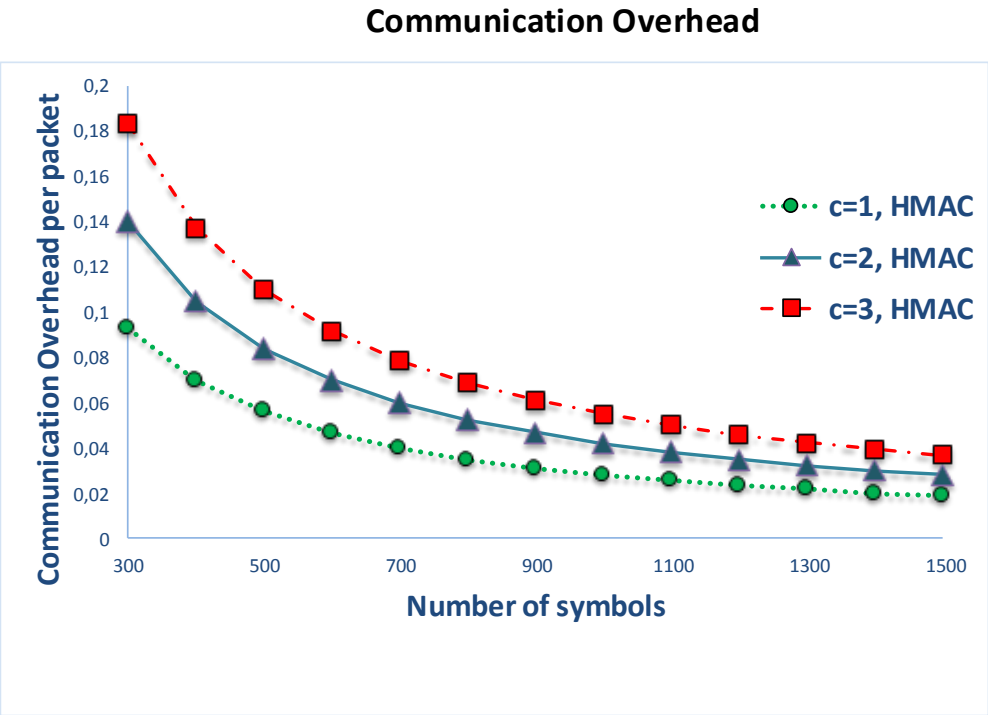

Figure 5.9: HMAC communication overhead per coded packet.

the signature of each coded packet. Thus, the total key storage size (in bits) required at the source node is given by the following equation:

$$T_{Source} = \{l * (m + n + 1) + l^{'} * (l + 1) + l^{'} + 1)\} * \lceil \log_2 p \rceil \tag{5.27}$$

Furthermore, each of the intermediate and sink node has to store one key vector for verifying the corresponding MAC, one key vector for verifying the corresponding D-MAC, and the public key for verifying the signature of each coded packet. Hence, the total key storage size (in bits) required at each intermediate and sink node is given by the following equation:

$$T_{Non-Source} = \{(m + n + 1) + (l + 1) + l^{'} + 1)\} * \lceil \log_2 p \rceil \tag{5.28}$$

Compared to MacSig scheme in [3], HMAC achieves almost 50% less key storage overhead at the source node. In addition, HMAC achieves a significant gain (200 to 400 times), compared to MacSig scheme, in terms of the total key storage size required at each non-source node. Figure 5.10 shows the total key storage size required by HMAC at the source node compared to MacSig. Moreover, Figure 5.11 shows the total key storage size required by HMAC, at each non-source node, compared to MacSig.

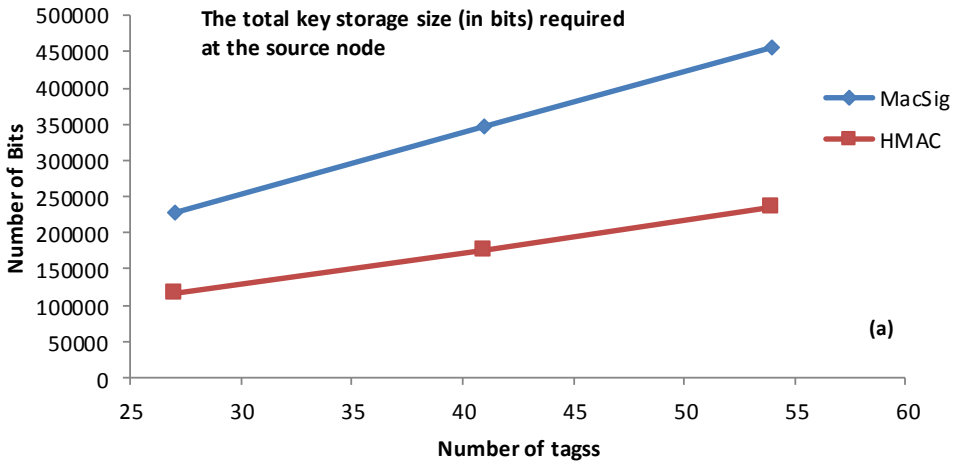

Figure 5.10: The total key storage size at the source for HMAC and MacSig in terms of the number of tags (i.e. MACs, D-MACs, and signature).

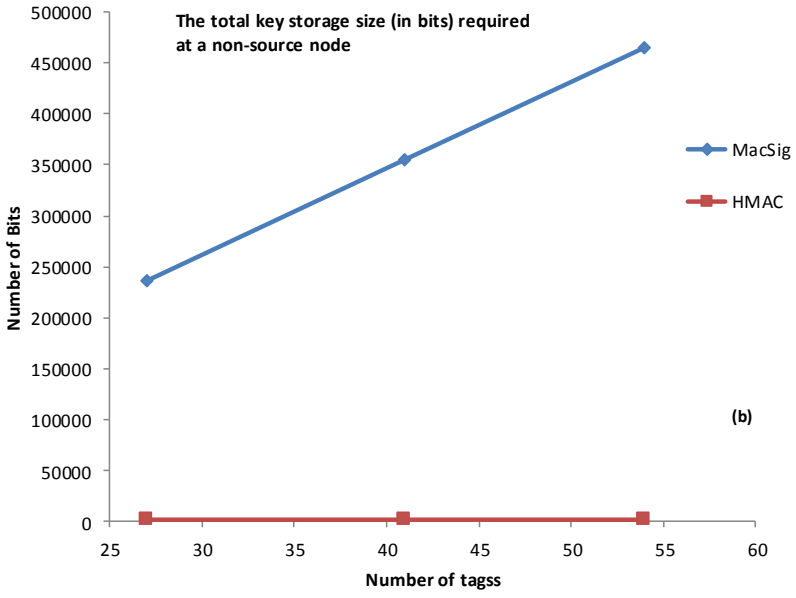

Figure 5.11: The total key storage size at a non-source for HMAC and MacSig in terms of the number of tags (i.e. MACs, D-MACs, and signature).

5.4.9 Conclusion

In this section, an efficient homomorphic MAC-based scheme, called HMAC, has been proposed to provide resistance against data pollution attacks and tag pollution attacks by using three types of homomorphic tags (i.e. MACs, D-MACs, and one signature) which are appended to the end of the coded packet.

The performance evaluation of HMAC shows that it is more efficient in terms of complexity, communication overhead and key storage overhead, compared to the MacSig scheme, which is the most well-known tag pollution immune scheme taking advantage of homomorphic MACs and signature. Particularly, HMAC has a verification process almost 10 times faster than MacSig, and achieves 25% less communication overhead than MacSig. Finally, HMAC achieves almost 50% less key storage overhead at the source node and a significant gain, in terms of the key storage overhead at each non-source node, compared to the MacSig scheme.

Part II

Null Space-based Homomorphic MAC

5.5 A Null Space-based Homomorphic MAC Scheme

5.5.1 Introduction

To detect both data and tag pollution attacks in an efficient way, in this Section, we propose an efficient null space-based homomorphic MAC scheme for RLNC-enabled wireless networks. According to our scheme, the source generates multiple MACs (tags) and D-MACs for each message packet. The former one ensures the integrity of the packet and the latter one ensures the integrity of the MACs. We show that our scheme can resist against data and tag pollution attacks.

The rest of the Section is outlined as follows. In Section 5.5.2, the outline of the proposed scheme is discussed. Then in Section 5.5.3 and Section 5.5.4, we describe the construction and correctness of the proposed scheme, respectively. Furthermore, in Section 5.5.5 the security analysis of the proposed scheme is given. Our performance evaluation takes place in Section 5.5.6. Finally, Section 5.5.7 concludes the Section.

5.5.2 Outline

To generate the required MACs and D-MACs, our scheme consists of three steps. In step 1, it computes the l MACs (i.e., tags) for each message according to the keys which are chosen randomly from a set of secret keys $\mathcal{K}_S$. Then, in step 2, our scheme computes a number of D-MACs (i.e., tags of tags), which should satisfy an orthogonality property of the initial MACs calculated in step 1, and it is calculated by using the properties of Null keys which was defined in the previous section. Finally, in step 3, the computed MACs and D-MACs are appended to the message. These three steps are depicted in Figure 5.12.

5.5.3 Construction

This scheme includes four steps: *Setup, Tag Generation, Verification* and *Encoding* detailed as follows:

1. *Setup:*

   - Key Distribution Centre (KDC) distributes the following set of secret keys (i.e., key vectors) to the source node S:

   $$\mathcal{K}_S = \{\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_l\}, \mathbf{k}_i \in \mathbb{F}_p^{m+n+1} \tag{5.29}$$

   - KDC also distributes the following subsets of secret vectors to all intermediate nodes and sink nodes:

   $$\mathcal{K}_{n_i} = \{\mathbf{k}_1, \cdots, \mathbf{k}_R\}, \ \ \mathbf{k}_r \in \mathbb{F}_p^{m+n+1} \tag{5.30}$$

2. *Tag generation*

   For every data packet $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$ of each generation consisting of m data packets, the source uses the equations 5.31 and 5.32 to generate the l MACs and $l^{'}$ D-MACs,

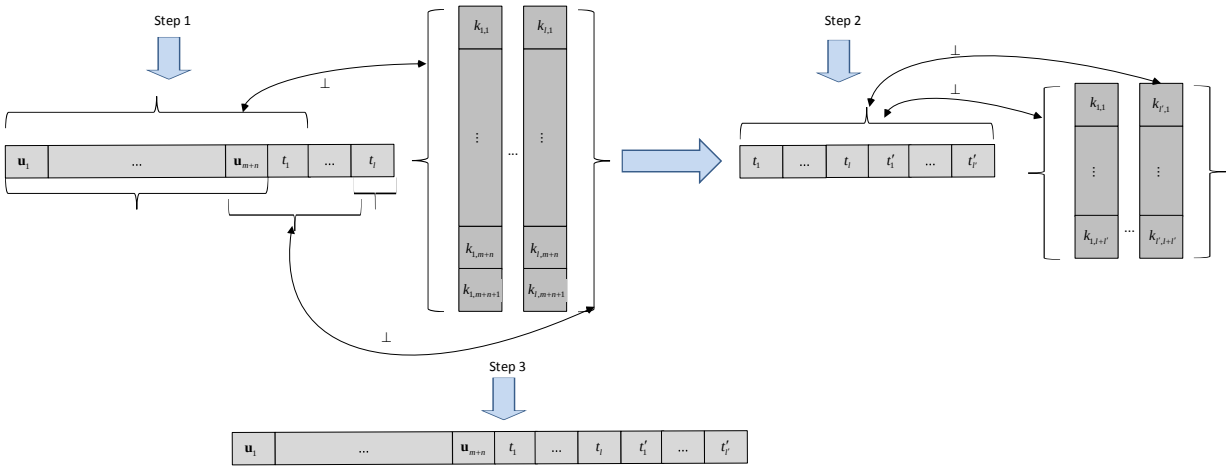

Figure 5.12: Our proposed scheme description. In the first step, we make l MACs (i.e., tags) by choosing l sets of keys from pool keys. Next, in step 2, we use the same keys to generate $l^{'}$ D-MACs (we use $l^{'}$ of l keys which is used for the first step). Finally, in step 3, we append these l MACS and $l^{'}$ D-MACs at the end of message packet.

respectively:

$$t_{\mathbf{u}_{i,l}} = -\frac{\sum_{j=1}^{m+n} \mathbf{u}_{i,j} \times k_{l,j}}{k_{l,m+n+1}} \tag{5.31}$$

$$\begin{pmatrix} k_{1,1} & \cdots & k_{1,l} \\ \vdots & \ddots & \vdots \\ k_{l',1} & \cdots & k_{l',l} \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_l \end{pmatrix} + \begin{pmatrix} k_{1,(l+1)} & \cdots & k_{1,(l+l')} \\ \vdots & \ddots & \vdots \\ k_{l',(l+1)} & \cdots & k_{l,(l+l')} \end{pmatrix} \begin{pmatrix} {t'}_1 \\ \vdots \\ {t'}_{l'} \end{pmatrix} = \mathbf{o}_{(l' \times 1)} \tag{5.32}$$

3. *Verification:*

When a relay or sink node receives a coded packet$y \in \mathbb{F}_p^{m+n}$ with its tags, this node checks the correctness of packet y using the algorithm *Verify* via its pre-distributed keys $\mathcal{K}_{n_i}$. If the results of following equations are 0, the received coded packet y is correct and the output is 1; otherwise the output is 0 and then this packet is discarded.

$$\delta_r = \left(\sum_{j=1}^{m+n} y_j \; k_{r,j} \right) + t_{y,r} \; k_{\; r,\, m+n+1} = 0, \quad \forall \; \mathbf{k}_r \in \{\mathbf{k}_1, \cdots, \mathbf{k}_R\} \tag{5.33}$$

$$\delta_r^{'} = \left(\sum_{j=1}^{l} t_{y,j} \; k_{r,j} \right) + \left(\sum_{j=L+1}^{l'} {t'}_{y,j} \; k_{r,j} \right) = 0, \quad \forall \; \mathbf{k}_r \in \{\mathbf{k}_1, \cdots, \mathbf{k}_R\} \tag{5.34}$$

4. *Encoding:* When an intermediate node receives h encoded packets $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$ $(1 \leq i \leq h)$, and they all are checked or considered to be correct, a forward coded packet along with new tags is generated using the following algorithm with locally randomly generated coefficients c_i:

$$combine\left(\left(\mathbf{u}_i, t_{\mathbf{u}_i,1}, \cdots t_{\mathbf{u}_i,L}, t'_{\mathbf{u}_i,1}, \cdots, t'_{\mathbf{u}_i,L'}\right)_{i=1}^{h}, (c_i)_{i=1}^{h}\right) \tag{5.35}$$

5.5.4 The Correctness of proposed scheme

Our proposed scheme is correct if the *Verify* algorithm passed the verification by getting 1 for the output of the two following algorithms. Our first algorithm (i.e., Algorithm 3) provides resistant against pollution attack; and the second one (i.e., Algorithm 4) takes care of tag pollution attacks. The Algorithm 4 is run whenever Algorithm 3 is given 1 as the result; otherwise the native packet is discarded.

Theorem 12. *Algorithm 3 is correct.*

Proof. According to the description of "*Setup*", there is a key $\mathbf{k}_l = \mathbf{k}_r$ for $l = r$.

As a result, by letting $y = \mathbf{u}_i, \mathbf{k}_l = \mathbf{k}_r$, and substituting $t_{y,r} = t_{\mathbf{u}_i,l}$ using Equation 5.31, then for sure, according to Equation 5.33, we have that: $\delta_r = 0, \forall \mathbf{k}_r \in \mathcal{K}_{n_i}$.

As a result, the Algorithm 3 is correct:

$$Verify\left(\mathcal{K}_{n_i}, \quad (\mathbf{u}_i,\ MAC(\mathcal{K}_S, \mathbf{u}_i))\right) = 1 \tag{5.36}$$

□

Theorem 13. *Algorithm 4 is correct.*

Proof. According to Equation 5.32, the D-MACs are created in a way that $\mathbf{k}_l$ is orthogonal to the concatenation of $t_{\mathbf{u}_i,j}$ and $t'_{\mathbf{u}_i,r}$ where $(j{=}1, ..., l)$, $(r{=}1, ..., l')$. It can be represented as: $k_{\mathbf{u}_i,i} \perp (t_{\mathbf{u}_i,j} \parallel t'_{\mathbf{u}_i,r})$. This relationship is true for all packets in the same generation. We assume that an adversary modifies d tags (i.e., t_i or t'_j), so regarding the Equation 5.32, if we assigned only one key vector to each node, the possibility of getting the result 1 is equal to $\frac{1}{p^d}$ and it is very small (i.e., if $\mathbb{F}_p = \mathbb{F}_{2^8}$ and $d = 2$, this probability should be $\frac{1}{2^{16}} \approx 0.001\%$.). However, according to the key distribution model, we assign more than one vector keys, and it means that modifying any tags would be impossible and it will be detected immediately. Thus, the probability of a packet with modified tags passing the verification at two nodes is small and the second Algorithm is also correct. □

5.5.5 Security analysis

We assume the source node is trustworthy and the process of key pre-distribution is secure. Hence, the secret key sets k_S, assigned to the source node are considered secure. However, an adversary can wiretap all the data communication in a network and may compromise several relay or sink nodes. Hence, the adversary can get access to the received packets from the previous hops, the key information distributed to him by the KDC and the key information stored at the compromised nodes. We consider three types of attacks as the following:

Algorithm 3: Verification Procedure for data integrity

Input:

- A packet $Y \in \mathbb{F}_{\mathrm{p}}^{m+n}$,
- A set of key vectors $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{m+n+1}$, where $\mathcal{K}_i \in \mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l$,
- l MACs where $t_i \in \mathbb{F}_{\mathrm{p}}^{l}$,
- $count = 0$

;
Output: 0 or 1 ; /* Either 1 (accept), or 0 (reject) */
;
main()
foreach $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{m+n+1}$ **do**
  if $Verify\,(\mathcal{K}_{n_i},\ \ (\mathbf{u}_i,\ MAC(\mathcal{K}_S, \mathbf{u}_i))) = 1$ **then**
    $count++$;
    else
      continue;
if $count > 0$ **then**
  $output = 1$;
  ; /* (call algorithm 4) */
  exit; ;
  else
    $output = 0$; ; /* (reject the packet) */

Algorithm 4: Verification Procedure for tag integrity

Input:

- A packet $Y \in \mathbb{F}_{\mathrm{p}}^{m+n}$,
- A set of key vectors $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{m+n+1}$, where $\mathcal{K}_i \in \mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_l$,
- l MACs where $t_i \in \mathbb{F}_{\mathrm{p}}^{l}$,
- $count = 0$

;
Output: 0 or 1 ; /* Either 1 (accept), or 0 (reject) */
;
main()
foreach $\mathcal{K}_i \in \mathbb{F}_{\mathrm{p}}^{m+n+1}$ **do**
  if $Verify\,(\mathcal{K}_{n_i},\ \ (MAC(\mathcal{K}_S, \mathbf{u}_i), D - MAC(\mathcal{K}_S, \mathbf{u}_i))) = 1$ **then**
    $count++$;
    else
      continue;
if $count > 0$ **then**
  $output = 1$;
  ; /* (Accept the packet */
  exit; ;
  else
    $output = 0$; ; /* (reject the packet) */

1. *Data Pollution attack:*

   If an adversary makes any change in the native packet, it is detected immediately; however, there is a possibility to travel more hops if these hops don't have the key. This probability is negligible.

2. *Tag Pollution attack (i.e., MACs):*

   By checking the result of Algorithm 4, if an adversary makes any change in any tags, it is detected immediately.

3. *Tag Pollution attack (i.e., D-MACs):*

   Regarding the tags (i.e., D-MACs) which are based on Null space properties and calculated by source node, there is no possibility to alter these tags.

5.5.6 Performance evaluation

In this section we provide a communication and computational complexity comparison of our proposed scheme and the two related works which can resist against tag pollution attacks [3], [4].

5.5.6.1 Communication Overhead

In [3], the MacSig scheme generates L MACs and a signature. This scheme uses a variable number of MACs which is calculated by $L = \frac{1}{\delta-1}e(c+1)\ln\frac{1}{q}$, where δ and q are security parameters, and c is the number of compromised nodes. Their idea relies on calculating L MACs and a signature by a source node. Totally, $L+1$ tags are appended at the end of each native packet. An intermediate node or sink node should verify each received packet by using the verification algorithm. We set the number MACs equal to l where this value is less than the number of MACs which is used in [3] (i.e., L). Moreover, we consider the value of $l^{'}$ as the number of D-MACs, where this value can be less or equal to l. In other word, the total number of tags which be used in our proposed scheme is $l + l' < L$.

However, the idea of [4] is based on generating tags and appending these tags to each native packet. According to their consideration, this value is somehow equal to L. To end this section, our proposed scheme provides a less communication overhead in comparing to the related works.

5.5.6.2 Computational Complexity

For providing L MACs and a signature at a source node, MacSig [3] needs to use $L(m+n+1)$ and $(m+L+1)$ multiplications, respectively. However, for verification phase, it needs to do multiplications and exponentiations, where according to their setup parameters; it should be equal to $\frac{3}{2}|p|(m+L+1)+(m+n+1)L$ multiplications totally. However, KEPTE [4] needs to calculate $N(m+n)$ multiplications.
In our proposed scheme, we need to generate $l+l'$ tags. So, a source node needs to do $l(m+n+1)+l'(l'+l)$ multiplications. However, each node needs to verify a received packet by calculating $l(m+n+1)+(l'+l)$ multiplications. For more detail, see Table 5.1.

Table 5.1: Computational Complexity. L and N are the total number of tags which is used in [3] and [4], respectively.

	At the source node	**At each intermediate or sink node**
MacSig [3]	$(m+n+1)L+(m+L+1)$	$\frac{3}{2}\|p\|(m+L+1)+(m+n+1)L$
KEPTE [4]	$N(m+n)$	$N(m+n)$
Our proposed Scheme	$l(m+n+1)+l'(l'+l)$	$l(m+n+1)+(l'+l)$

5.5.7 Conclusion

In this section, a lightweight encryption scheme on top of network coding, to further to provide resistance against data pollution and tag pollution attacks by using two types of tags (i.e., MACs and D-MACs) was presented. We showed that our proposed scheme is efficient in computation, and incurs less communication overhead and keys storage for encryptions/decryptions.

5.6 An Efficient Null Space-based Homomorphic MAC Scheme

5.6.1 Introduction

In this Section, we propose an efficient homomorphic MAC scheme, where each tag has not a fixed location in the coded packet, providing resistance against tag pollution attacks. In the proposed scheme, the tags are calculated based on null space properties and then, they are secretly swapped with the packet data [17]. Hence, they are not appended to the end of the coded packet and thus, an adversary is not able to distinguish them from the packet data. Our results show that the proposed scheme is more efficient compared to the schemes proposed in [3] and [4], in terms of computational complexity without incurring additional communication overhead. To the best of our knowledge, the proposed schemes in [3] and [4] are the most competitive schemes in the literature for providing resistance against tag pollution attacks in RLNC.
The rest of the Section is outlined as follows. In Section 5.6.2, the RLNC model is presented. Then in Section 5.6.4, the proposed efficient null space-based homomorphic MAC scheme is described. Furthermore, in Section 5.6.5 the security analysis of the proposed scheme is given. Our performance evaluation takes place in Section 5.6.6. Finally, Section 5.6.7 concludes the Section.

5.6.2 RLNC Model

Similarly to the Section 5.3, at the setup phase, the source divides each message into a sequence of native packets and partitions them into generations. Thus, we consider that each generation consists of m native packets denoted as $\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, ..., \underline{\mathbf{u}}_m$. Each native packet $\underline{\mathbf{u}}_i$, for $1 \leq i \leq m$, is represented as a vector $\underline{u}_1, \underline{u}_2, ..., \underline{u}_n$ in the finite field $\mathbb{F}_p^n$, where $p = 2^8$ and it denotes the size of the finite field. Then, the source S generates a coded packet $\mathbf{u}_i$ for each native packet $\underline{\mathbf{u}}_i$ by prefixing $\underline{\mathbf{u}}_i$ with the i^{th} unit vector of dimension m. The coded packet is

reintroduced and represented as a row vector in the finite field $\mathbb{F}_p^{m+n}$ as follows:

$$\mathbf{u}_i = (\overbrace{\underbrace{0, \cdots, 0}_{i-1}, 1, 0, \cdots, 0}^{m}, \underline{u}_{i,1}, \cdots, \underline{u}_{i,n}) \in \mathbb{F}_p^{m+n} \tag{5.37}$$

For simplicity, Equation 5.37 can also be written as follows:

$$\mathbf{u}_i = (u_{i,1}, \cdots, u_{i,m+n}) \in \mathbb{F}_p^{m+n} \tag{5.38}$$

After that, the source S transmits the coded packets to its neighbor nodes. Each intermediate node buffers its received packets $\mathbf{u}_i$ temporarily and creates a coded packet y, which is a linear combination of a number of h coded packets $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_h$ belonging to the same generation. A coded packet is represented as follows:

$$y = \sum_{i=1}^{h} \alpha_i \mathbf{u}_i \tag{5.39}$$

where α_i is randomly selected from $\mathbb{F}_p$, and all the arithmetic operations are done over the finite field $\mathbb{F}_p$. A coded packet y is considered to be valid if it is in the linear subspace spanned by the coded packets generated at the source. This is denoted as $y \in span\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$. In fact, when y is valid, the linear combination coefficients are the first m symbols of the packet y. Otherwise, y is invalid and it is denoted as $y \notin span\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$, which may be caused due to transmission errors or pollution attacks. At the destination, when a sink node obtains m linearly independent coded packets, it can decode them by using Gaussian eliminations [18].

5.6.3 Adversary Model

In our adversary model, the adversary aims to corrupt (i.e., pollute) tags of coded packets belonging to the same generation, so that coded packets with legitimate content will be discarded later from next intermediate nodes due to the corrupted tags. It is possible for a coded packet with legitimate content and corrupted tags to travel multiple hops if the intermediate nodes hold the same keys with the adversary or they do not hold those keys that can verify the coded packet and detect the corrupted tags. Otherwise, if the next intermediate nodes hold the keys that verify the coded packet, then the corrupted tags will be detected immediately and the coded packet will be discarded. Moreover, we assume the source node and the destination nodes are trustworthy and secure. Finally, the process of key distribution is considered secure.

5.6.4 An Efficient Null Space-based homomorphic MAC Scheme

5.6.4.1 Construction

The proposed homomorphic MAC scheme is based on null space properties [115] for tag generation and verification. The proposed scheme consists of the following five steps:

1. Key Distribution to Source Node
   A Key Distribution Center (KDC) distributes L key vectors $\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_L$ to the source node. Each of them is represented in the finite field $\mathbb{F}_{\mathrm{p}}^{m+n+L}$.

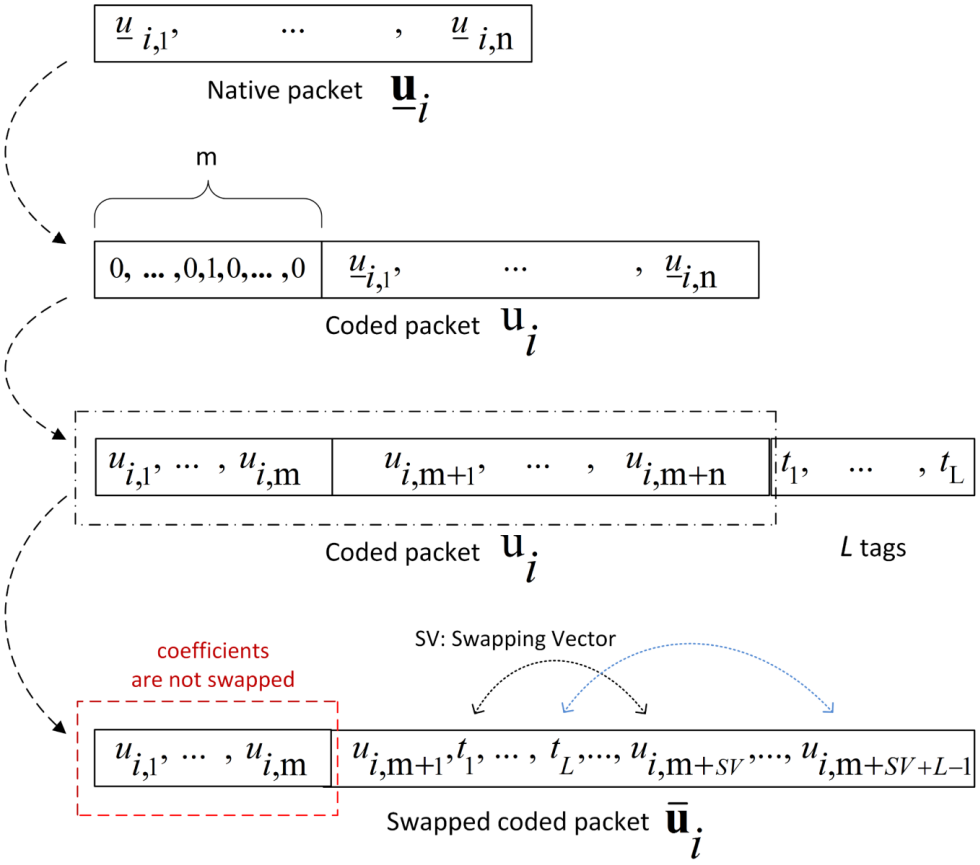

Figure 5.13: Swapping process in the proposed null space-based homomorphic MAC scheme.

2. Tag generation
The source node S uses the L key vectors $\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_L$ to produce L tags for each coded packet consisting of $m+n$ symbols. These L tags $(t_1, t_2, ..., t_L$, where $t_i \in \mathbb{F}_p)$ are calculated according to the following formula:

$$\begin{bmatrix} \mathcal{K}_{1,1} & \dots & \mathcal{K}_{1,m+n} \\ \vdots & \vdots & \vdots \\ \mathcal{K}_{L,1} & \dots & \mathcal{K}_{L,m+n} \end{bmatrix}_{L*(m+n)} * \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ \vdots \\ u_{i,m+n} \end{bmatrix}_{(m+n)*1} + \begin{bmatrix} \mathcal{K}_{1,m+n+1} & \dots & \mathcal{K}_{1,m+n+L} \\ \vdots & \vdots & \vdots \\ \mathcal{K}_{L,m+n+1} & \dots & \mathcal{K}_{L,m+n+L} \end{bmatrix}_{L*L} * \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_L \end{bmatrix}_{L*1} = 0 \tag{5.40}$$

Afterwards, the source node appends the L calculated tags to the end of the coded packet $\mathbf{u}_i$, which is created by prefixing the native packet $\underline{\mathbf{u}}_i$ with m coefficients, as it is shown in Fig 5.13.

3. Swapping
   To avoid tag pollution attacks, the L tag symbols of the coded packet $\mathbf{u}_i$ are swapped with only L out of the n symbols of the coded packet $\mathbf{u}_i$. It is worthwhile to mention that the coefficients of the coded packet do not participate in the swapping process so that the destination nodes can decode correctly. Particularly, the swapping process is based on a secret value SV (i.e., positive integer) that plays the role of the swapping vector. This secret value is generated randomly by the KDC through a pseudorandom function, and it is known to the source node and all the destination nodes. However, it is unknown to the intermediate nodes. The result of this swapping process is a swapped coded packet $\overline{\mathbf{u}}_i$, where the L tags symbols are mixed with the n symbols of the coded packet $\mathbf{u}_i$, as it is shown in Fig 5.13, where $SV = 2$. Each swapped coded packet is represented as follows:

$$\overline{\mathbf{u}}_i = Swap(\mathbf{u}_i)_{SV} \tag{5.41}$$

   On the other hand, at the destination nodes, an inverse swapping is required, before RLNC-decoding, to obtain the native packet.

4. Key Distribution to Intermediate and Destination Nodes
   Based on the swapping vector SV, the KDC creates new key vectors $\mathcal{K}'_1, \mathcal{K}'_2, ..., \mathcal{K}'_L$ by swapping accordingly the symbols of each key vectors $\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_L$, which have been used by the source node for tag generation. Each new key vector is represented as follows:

$$\mathcal{K}'_i = Swap(\mathcal{K}_i)_{SV} \tag{5.42}$$

   The KDC of our scheme adopts a key distribution model, based on the cover free set systems [25, 133], in order to provide resistance against c compromised nodes. In this model, the maximum number of key vectors that should be assigned to each intermediate and destination node cannot be more than $R = e * (c+1) * ln(1/q)$, where q is a security parameter (usually $q = 10^{-3}$). In our proposed scheme, this assumption is satisfied since only one key vector is required to be assigned by the KDC to each intermediate and destination node. This is why each key vector is orthogonal to the swapped coded packet, and thus the intermediate and destination nodes require only one key vector to verify the swapped coded packet.

5. Verification
   Given that each intermediate and destination node holds a key vector, $\mathcal{K}'_i$, they verify a swapped coded packet $\overline{\mathbf{u}}_i$ based on the following formula:

$$\delta = Swap(\mathcal{K}_i)_{SV} * Swap(\mathbf{u}_i)_{SV} = \sum_{j=1}^{m+n+L} \mathcal{K}'_{i,j} * \overline{\mathbf{u}}_{i,j} \tag{5.43}$$

   If $\delta = 0$, then the swapped coded packet $\overline{\mathbf{u}}_i$ is accepted and transmitted to the next nodes. Otherwise, it is discarded.

5.6.4.2 Correctness

Theorem 14. *(Correctness) The proposed scheme is correct.*

Proof. (Poof by Contradiction) We assume to the contrary that the proposed scheme is not correct. If this is the case, from Equation 5.43 we have that $\delta \neq 0$. We consider a coded packet $x^i = (x_1^i, ..., x_{m+n}^i)$ and L key vectors $K = (\mathcal{K}_1, ..., \mathcal{K}_L)$, to generate L tags $t^i = (t_1^i, ..., t_L^i)$ based on Equation 5.40. According to Equation 5.43 and by considering $SV = 1$, we have the following:

$$Swap(\mathcal{K})_{SV} * Swap(\overline{x}^{i^T})_{SV} =$$

$$\begin{bmatrix} \mathcal{K}_{1,1}...\mathcal{K}_{1,m+n+1}...\mathcal{K}_{1,m+n+L} & \mathcal{K}_{1,m+1}...\mathcal{K}_{1,m+L} \\ \vdots & \\ \mathcal{K}_{L,1}...\mathcal{K}_{L,m+n+1}...\mathcal{K}_{L,m+n+L} & \mathcal{K}_{L,m+1}...\mathcal{K}_{L,m+L} \end{bmatrix} * \\ \begin{bmatrix} \mathbf{x}_1^i & ... & t_1^i & ... & t_L^i & ... & \mathbf{x}_{m+n}^i & \mathbf{x}_{m+1}^i & ... & \mathbf{x}_{m+L}^i \end{bmatrix}^T \tag{5.44}$$

$$= \begin{bmatrix} \mathcal{K}_{1,1} * \mathbf{x}_1^i + ...\mathcal{K}_{1,m+n+L} * t_L^i \\ \vdots \\ \mathcal{K}_{L,1} * \mathbf{x}_1^i + ...\mathcal{K}_{L,m+n+L} * t_L^i \end{bmatrix}_{L*1} \tag{5.45}$$

However, the vector $\overline{x}^i = [x_1^i, ..., x_{m+n}^i, t_1^i, ..., t_L^i]$ is orthogonal to each of the L key vectors according to Equation 5.40. Thus, we have the following:

$$K * \overline{x}^{i^T} = \begin{bmatrix} \mathcal{K}_{1,1} & ... & \mathcal{K}_{1,m+n+L} \\ \vdots & \vdots & \vdots \\ \mathcal{K}_{L,1} & ... & \mathcal{K}_{L,m+n+L} \end{bmatrix} * \begin{bmatrix} \mathbf{x}_1^i \\ \mathbf{x}_2^i \\ \vdots \\ t_L^i \end{bmatrix}$$
$$= \begin{bmatrix} \mathcal{K}_{1,1} * \mathbf{x}_1^i + ...\mathcal{K}_{1,m+n+L} * t_L^i \\ \vdots \\ \mathcal{K}_{L,1} * \mathbf{x}_1^i + ...\mathcal{K}_{L,m+n+L} * t_L^i \end{bmatrix}_{L*1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{L*1} \tag{5.46}$$

From Equations 5.45 and 5.46, it is obvious that $\delta = 0$. However, this contradicts the original assumption that $\delta \neq 0$. Thus, the proposed construction is correct. According to the inductive reasoning, it can also be proved that the proposed scheme is correct for each SV $(1 \leq SV \leq n)$.

□

5.6.5 Security Analysis

We consider that an adversary can wiretap all the swapped coded packets (i.e., $\overline{\mathbf{u}}_i$) of the network and obtain the key vector $\mathcal{K}_i^{'}$ of each compromised node. In this section, we firstly calculate the probability of the adversary to guess the swapping vector SV of the source and then, we calculate the probability of the adversary to launch a tag pollution attack.

- Probability of Guessing the Swapping Vector SV
  SV cannot be obtained from compromised intermediate nodes, since it is known only to

the source node and the destination nodes which are considered trustworthy and secure. Also, SV cannot be derived from the transmitted swapped coded packets $\overline{\mathbf{u}}_i$ because they are already swapped and RLNC-encoded. Thus, the adversary can only take a random guess, which has a probability of $\frac{1}{n}$ to be correct. Moreover, the adversary is not able to confirm whether the random guess is correct or not.

- Probability of Launching a Tag Pollution Attack
  To launch a tag pollution attack, an adversary has to corrupt a swapped coded packet and send it to a neighbor node, where it should pass the verification step. According to Equation 5.43, the corrupted packet will pass the verification in a neighbor node, if this node keeps the same key vector with the adversary. Otherwise, the corrupted packet will be detected and discarded by the neighbor node. According to our key distribution model, each intermediate and destination node is assigned only one key vector out of the L source key vectors. Thus, in the case that the adversary has only one neighbor node, the probability that the adversary has the same key with its neighbor node is not greater than $1/L$. In other words, in this case, the probability that the adversary can launch a tag pollution attack is not greater than $1/L$. However, according to the RLNC model that we have considered for the proposed scheme, the case that the adversary has only one neighbor node is not possible at all. Indeed, it is only a theoretical case that we take into consideration for the sake of completeness. On the other hand, in the case that the adversary has d neighbor nodes, the probability that the adversary has the same key vector with one of its d neighbor nodes is not greater than $1/L^d$. Consequently, the probability of launching a tag pollution attack in this case is not greater than $1/L^d$, which is negligible. As a result the corrupted packet will be detected immediately. This case is realistic according to the RLNC model considered for the proposed scheme.

5.6.6 Performance Evaluation

We analyze the performance of our proposed null space-based homomorphic MAC scheme in terms of computational complexity and communication overhead. We follow the settings defined in [3] and thus, we set $p = 2^8, n = 1024$, and $m = 32$. We compare our proposed scheme with MacSig [3] and KEPTE [4] which are the most competitive schemes in the literature for providing resistance against tag pollution attacks in RLNC.

5.6.6.1 Computational Complexity

The computational complexity of our proposed null space-based homomorphic MAC scheme is considered for the source and non-source nodes. For the source node, we calculate the complexity of tag generation. For non-source nodes, we calculate the complexity of the verification step.

At source node: The number of tags which is generated at the source node is equal to L. According to Equation 5.40, $L * (m + n + L)$ finite field multiplications are required for generating the L tags. From Figure 5.14, we observe that our proposed scheme requires almost the same number of multiplications for tag generation as both the MacSig and KEPTE schemes.

At Non-source nodes: According to Equation 5.43, the verification step requires $m + n + L$ finite field multiplications. From Figure 5.15, we observe that our proposed scheme

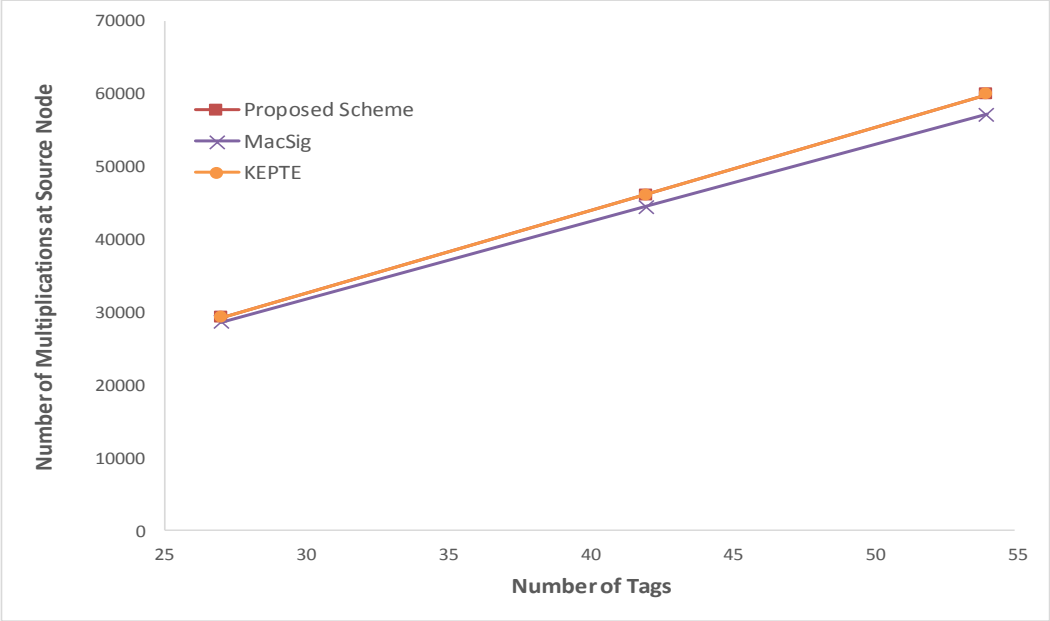

Figure 5.14: The number of multiplications required for tag generation at the source node.

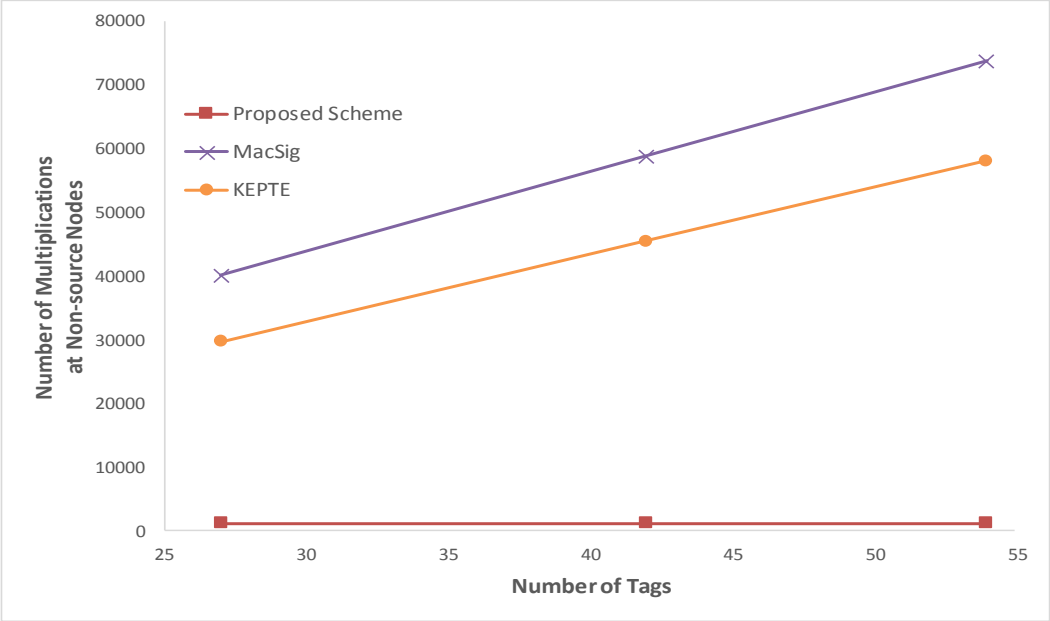

Figure 5.15: The number of multiplications required for tag verification at each non-source node.

requires much less number of multiplications for tag verification compared to both the MacSig and KEPTE schemes.

5.6.6.2 Communication Overhead

To calculate the communication overhead of our proposed scheme, we take into consideration the number of the tags appended to the end of each coded packet. However, each tag is one symbol of the finite field $\mathbb{F}_p^n$, where $p = 2^8$, and the total number of tags is $L = \frac{1}{1-\delta} e(c+1) \ln(\frac{1}{q})$, as it is shown in [3]. Consequently, the communication overhead of our proposed scheme is $L * \lceil \log_2 p \rceil$ bits per packet. Moreover, for the three different values of L given in [3] (i.e. $L = 27, 42$ and 54), our proposed scheme incurs 2% to 5% communication overhead, which is similar to the one incurred by the KEPTE scheme [4], but half of the communication overhead incurred by the MacSig scheme (*i.e.*, 5% to 10%) [3].

5.6.7 Conclusion

In this Section, we have proposed a null space-based homomorphic MAC scheme providing resistance against tag pollution attacks in RLNC. The performance evaluation of our scheme shows that it is more efficient compared to the MacSig and KEPTE schemes, which are the most competitive tag pollution immune schemes, in terms of computational complexity without incurring additional communication overhead.

5.7 Conclusion

NC-enabled wireless networks are vulnerable to pollution attacks where a malicious node injects into the network corrupted packets that prevent the destination nodes from decoding correctly. In this chapter, four schemes have been proposed to provide resistance against data and pollution attacks. The first two scheme, called Dual-HMAC and HMAC respectively, are based on inner multiplication and orthogonality of a native packet and key vector. Both schemes make use of two types of MACs (tags). However, in HMAC scheme, we append a signature, as well as the two tags, to provide resistance against tag pollution attacks. In the third and fourth schemes, the tags are created according to the null space properties. Our last scheme, called an efficient null space-based homomorphic MAC scheme also swaps the tags into the data packet. Thus, lunching a tag pollution attack in this scheme is meaningless. Moreover, our results have shown that our proposed schemes can benefit from a low computational complexity, key storage and bandwidth overheads.

Chapter 6

Main Contributions and Future Work

"To achieve great things, two things are needed; a plan, and not quite enough time."

Leonard Bernstein

In this thesis, motivated by the recent surge in network coding research and its applications, we have considered wireless communication networks which allow for network coding operations, i.e., the relay nodes perform XOR or linear NC. However, the advent of NC paradigm opens new challenges in terms of security aspects. During the thesis, we have mainly focused on pollution attacks that arise in this context. Moreover, our goal in this thesis was to propose some lightweight secure schemes that are resistance against pollution attacks. We showed (via qualitative comparisons) that the proposed schemes are not only very simple and lightweight when it comes to the computational complexity, but also they are highly bandwidth communication and storage efficient. We now present an overview of the original main contributions of this dissertation.

6.1 Summary of the Thesis

We have studied the general problem of how to secure a typical wireless network that uses network coding as its routing policy. In a very broad sense, we categorized the security threats and attacks in wireless networks into two main classes such as: active attacks and passive attacks. The main difference between these two attack types is that in active attacks (which are also known as pollution attacks or Byzantine attacks) the attacker tries to breach the integrity of the data, while in passive attacks (also known as eavesdropping attacks or wiretapping attacks) the confidentiality of the data is targeted and threatened.

- The emerging Wireless Networks (WNs) are a promising technology that can contribute significantly towards the improvement of safety, convenience and comfort in the daily life of modern societies. However, WNs are characterized by limitations that can affect the advancement of using WNs. These limitations are related to low communication bandwidth, packet loss and power consumption. Therefore, it is obvious that the success

and prosperity of WNs are dependent on advances of WNs in these limitations. Towards this direction, the application of NC technology in WNs, leading to NC-enabled WNs, can be a promising solution. Recent research efforts show that NC technology can provide significant benefits to WNs in terms of improvement in network capacity and robustness among others. However, due to the fact that NC technology enables the intermediate nodes to manipulate and mix incoming data packets from neighboring nodes as well as to transmit data packets between them, security issues are raised in NC-enabled WNs. Additionally, these security issues can be exploited by attackers to compromise the security of the WNs, which is a critical factor for their success. Therefore, in Chapter 3, we have focused on the current existing NC-enabled WNs attacks that can have a negative impact on security of the whole wireless networks. Furthermore, we have presented the current mitigation techniques, derived from the literature, against the most common attacks identified here. In addition, our objective was to propose a framework for positioning our research proposals towards the design and deployment of more reliable and efficient defense mechanisms against known attacks for NC-enabled WNs.

- In Chapter 4, we have proposed an efficient Message Authentication Code (MAC)-based scheme providing resistance against pollution attacks in XOR NC-enabled WNs. Our proposed scheme makes use of a number of MACs which are appended to the end of each native packet. Our results show that the proposed MAC-based scheme is more efficient compared to the scheme proposed in [2], in terms of communication bandwidth and computational complexity. The proposed scheme in [2] is the most competitive scheme in the literature for securing XOR NC against pollution attacks in resource-constrained wireless networks. Particularly, our scheme saves communication bandwidth, since it detects corrupted packets in the next hop with high probability, and its computational complexity is lower because it requires less XOR operations than the scheme proposed in [2]. More precisely, we observe that verification process in our proposed scheme is almost 5 to 10 times (l times) faster than [2].

- In Chapter 5, we presented our work into two parts.

  – In the first part, we proposed two schemes, called a dual homomorphic Message Authentication Code scheme and an efficient homomorphic Message Authentication Code scheme respectively. The first proposed scheme makes use of two types of tags to provide resistance against data pollution attacks and partially tag pollution attacks. Furthermore, our proposed scheme presents low communication overhead and low computational overhead compared to other existing schemes. In the second proposed scheme (i.e., an efficient homomorphic Message Authentication Code scheme), we have provided resistance against data pollution attacks and tag pollution attacks by using three types of homomorphic tags (i.e. MACs, D-MACs, and one signature) which are appended to the end of the coded packet. The performance evaluation of our proposed scheme shows that it is more efficient in terms of complexity, communication overhead and key storage overhead, compared to the MacSig [3] scheme, which is the most well-known tag pollution immune scheme taking advantage of homomorphic MACs and signature. Particularly, our proposed scheme has a verification process almost 10 times faster than MacSig, and achieves 25% less communication overhead than MacSig. Moreover, our scheme

achieves almost 50% less key storage overhead at the source node and a significant gain, in terms of the key storage overhead at each non-source node, compared to the MacSig scheme.

– In the second part, we presented two schemes, called a null space homomorphic Message Authentication Code scheme and an efficient null space homomorphic Message Authentication Code scheme respectively. In the null space homomorphic Message Authentication Code scheme, the main idea was based on the randomization and the subspace properties of random network coding. In our proposed scheme, a lightweight encryption overlay scheme was integrated to provide resistance against data pollution and tag pollution attacks by using two types of tags (i.e., MACs and D-MACs). In this scheme, the D-MACs are calculated based on null space properties. We showed that our proposed scheme is efficient in computation, and incurs less communication overhead and keys storage for encryptions/decryptions. Finally, in the last scheme, we have proposed an efficient null space homomorphic Message Authentication Code scheme. In the proposed scheme, to avoid tag pollution attacks, each tag has not a fixed location in the coded packet. More precisely, the tags are calculated based on null space properties and then, they are secretly swapped with the packet data. Hence, they are not appended to the end of the coded packet and thus, an adversary is not able to distinguish them from the packet data. Our results show that the proposed scheme is more efficient compared to the schemes proposed in [3] and [4], in terms of computational complexity without incurring additional communication overhead. To the best of our knowledge, the proposed schemes in [3] and [4] are the most competitive schemes in the literature for providing resistance against tag pollution attacks in Random Linear Network Coding (RLNC).

6.2 Future Research Directions

The heterogeneous and dynamic nature of wireless communications can raise many security challenges. More precisely, NC-enabled wireless networks can be the target of a wide spectrum of security attacks including eavesdropping attacks, Byzantine attacks, pollution attacks, Denial of Service (DoS) attacks, and entropy attacks among others. In this sense, a lot of effort has been placed on secure NC schemes against these types of attacks. However, in this thesis, we only considered pollution attacks (i.e., data and tag pollution attacks). We now give an overview of future lines of research based on the work presented in this thesis.

6.2.1 Other Threat Models and Techniques

In Chapters 4 and 5, we proposed several secure network coding-enabled wireless networks schemes with two different pollution model scenarios: the first considers that the intermediate nodes only make changes to a packet data that is transmitted in the network, and the second assumes the attacker makes changes to the tags appended at the end of data packets. Therefore, one direction for future work is to adapt our schemes to be resistance against entropy attacks. As mentioned in Chapter 3, entropy attackers create and distribute coded packets that are valid linear combinations of the native data. As a result, a pollution defense is not helpful

against an entropy attack because all of the defenses against pollution attacks rely on the fact that an invalid coded packet is not a valid linear combination.

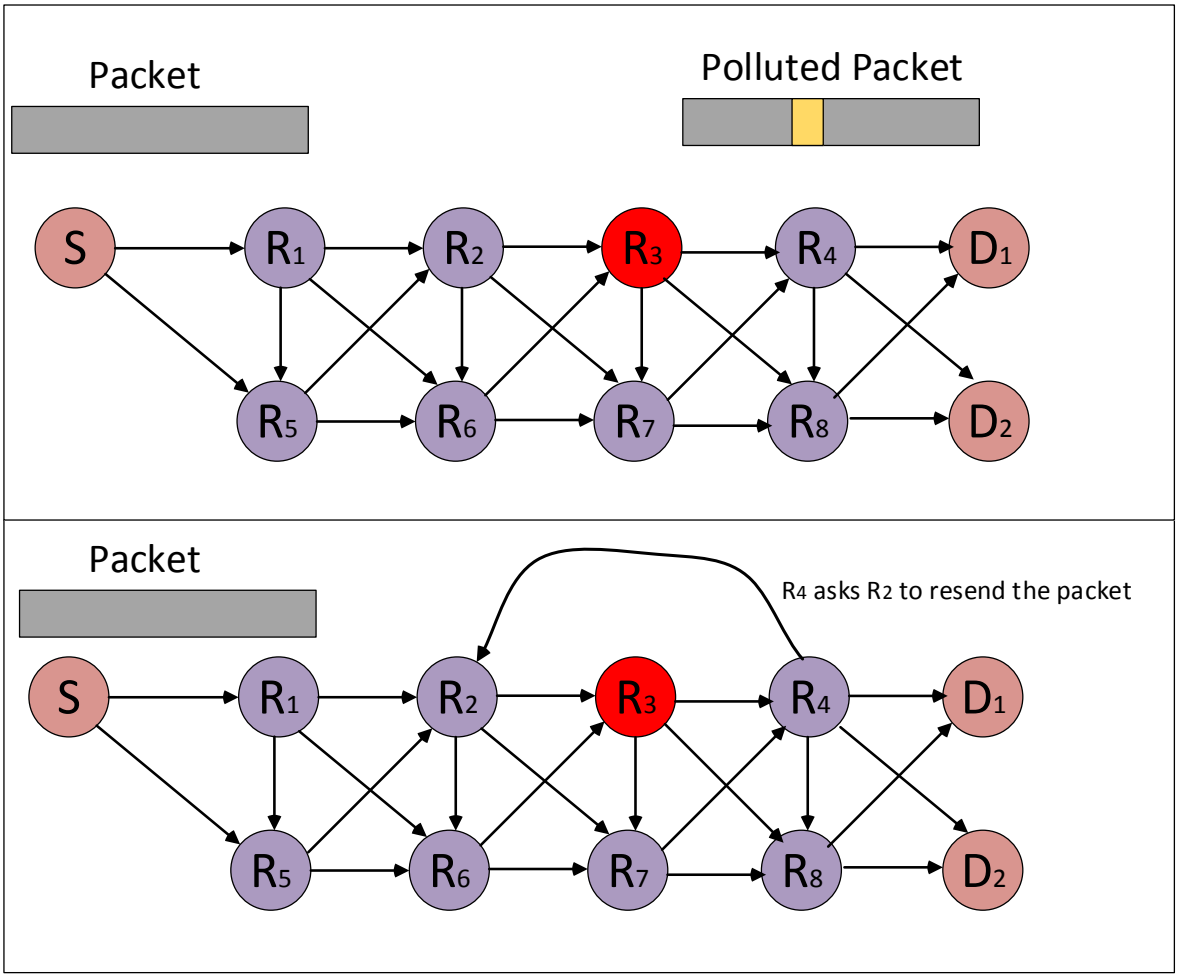

Figure 6.1: A scenario of locating scheme. In top, the polluted packet is discarded by node R_4. However, in the lower diagram, instead of discarding the polluted packet who has been polluted by R_3, there is a possibility to detect the attacker and ask from at least one node before the attacker (i.e., R_2) to resend the unpolluted packet through R_7. This leads to avoid wasting the energy.

6.2.2 Key Distribution Model

Another security challenge lies in the fact that the key distribution model, which is adopted in this thesis, is based on cover-free families [25,133] and is limited to a number of compromised nodes considered in our proposed schemes. However, the techniques used to detect pollution attacks in such scenarios must be independent to the number of adversaries. Thus, one promising approach may lie to investigate the use of different key distribution models which are not dependent on the number of compromised nodes.

6.2.3 Energy Efficiency

Energy efficiency is of vital importance for wireless communications in future networks, especially battery-constrained wireless and wireless sensor devices. In our proposed schemes in this thesis, we focused on finding the polluted packets and discarded them so as to avoid resource waste. However, in future, we could plan to not only detect the polluted packets, but to also find the exact location of the attacker. This may avoid discarding the packets, and to save energy. In this regard, we would like to consider the following schemes:

6.2.3.1 Locating Scheme

Since the proposed defense mechanisms against pollution attacks can be classified into three categories: 1) error correction [16, 118, 139], 2) pollution detection [3, 19, 20, 25, 26, 114, 115], and 3) polluters locations [104, 123, 140], we have studied only the pollution detection in this thesis. In this regards, future work should be focused on locating the attackers. More precisely, parents and children of any intermediate node cooperate to detect polluted packets sent by the adversary. This locating scheme forces nodes in the network to truthfully cooperate to exactly locate the pollution attackers. For instance, this scenario is shown in Figure 6.1. A packet which is transmitted by the node S is polluted by node R_3. In this case, a locating scheme not only can detect the polluted packet, but also detect R_3 as the adversary. Thus, instead of discarding the packet by R_4, it can ask from R_2 to resend the packet through different routes (i.e., R_7).

6.2.3.2 Isolating Scheme

After detecting the polluters with locating schemes, it would be interesting to design a scheme to isolate the polluting nodes from the rest of the network. This may also lead to energy saving in the network. However, what would be the tradeoff between locating the attackers and isolating the attackers?

References

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

[2] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient scheme for securing xor network coding against pollution attacks," in INFOCOM 2009, IEEE, pp. 406–414, IEEE, 2009.

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

[4] X. Wu, Y. Xu, C. Yuen, and L. Xiang, "A tag encoding scheme against pollution attack to linear network coding," Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 1, pp. 33–42, 2014.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, Trading structure for randomness in wireless opportunistic routing, vol. 37. ACM, 2007.

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

[7] 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, no. 6, pp. 1956–1970, 2011.

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

[9] J. Tan and M. Médard, "Secure network coding with a cost criterion," in 2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, pp. 1–6, IEEE, 2006.

[10] D. Wang, D. Silva, and F. R. Kschischang, "Constricting the adversary: A broadcast transformation for network coding," in Proc. of Allerton, 2007.

[11] R. W. Yeung and N. Cai, "On the optimality of a construction of secure network codes," in 2008 IEEE International Symposium on Information Theory, 2008.

[12] S. Yang and R. W. Yeung, "Coding for a network coded fountain," in Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on, pp. 2647–2651, IEEE, 2011.

[13] S. Yang and R. W. Yeung, "Large file transmission in network-coded networks with packet loss: a performance perspective," in Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, p. 117, ACM, 2011.

[14] H. Yao, D. Silva, S. Jaggi, and M. Langberg, "Network codes resilient to jamming and eavesdropping," IEEE/ACM Transactions on Networking (TON), vol. 22, no. 6, pp. 1978–1987, 2014.

[15] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, "Mis: Malicious nodes identification scheme in network-coding-based peer-to-peer streaming," in INFOCOM, 2010 Proceedings IEEE, pp. 1–5, IEEE, 2010.

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

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

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

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

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

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

[22] 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. 26th IEEE International Conference on Computer Communications. IEEE, pp. 616–624, IEEE, 2007.

[23] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," in Information Theory, 2007. ISIT 2007. IEEE International Symposium on, pp. 556–560, IEEE, 2007.

[24] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, IEEE, 2008.

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

[26] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "Ripple authentication for network coding," in INFOCOM, 2010 Proceedings IEEE, pp. 1–9, IEEE, 2010.

[27] C. Kaufman, R. Perlman, and M. Speciner, Network security: private communication in a public world. Prentice Hall Press, 2002.

[28] J. F. Kurose and K. W. Ross, "Computer networking: A top-down approach, 6/e," 2013.

[29] W. Stallings, Cryptography and Network Security, 4/E. Pearson Education India, 2006.

[30] B. Bollobás, Graph theory. Elsevier, 1982.

[31] K. Rosen, J. Michaels, J. Gross, J. Grossman, and D. Shier, "Handbook of discrete and combinatorial mathematics, crc," Press, Washington DC, vol. 490, 2000.

[32] N. Cai and R. W. Yeung, "Network coding and error correction," in Proc. 2002 IEEE Inform. Theory Workshop, pp. 119–122, 2002.

[33] R. Koetter and M. Médard, "Beyond routing: An algebraic approach to network coding," in INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 1, pp. 122–130, IEEE, 2002.

[34] T. Ho, D. R. Karger, M. Médard, and R. Koetter, "Network coding from a network flow perspective," in IEEE International Symposium on Information Theory, pp. 441–441, Citeseer, 2003.

[35] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," IEEE Transactions on Information Theory, vol. 51, no. 6, pp. 1973–1982, 2005.

[36] L. Ma, "Design and reliability performance evaluation of network coding schemes for lossy wireless networks," 2013.

[37] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," IEEE Transactions on Information Theory, vol. 51, no. 8, pp. 2745–2759, 2005.

[38] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," IEEE Transactions on Information Theory, vol. 56, no. 9, pp. 4539–4551, 2010.

[39] 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 (ToN), vol. 16, no. 3, pp. 497–510, 2008.

[40] J. Widmer and J.-Y. Le Boudec, "Network coding for efficient communication in extreme networks," in Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, pp. 284–291, ACM, 2005.

[41] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "Further results on coding for reliable communication over packet networks," in Proceedings. International Symposium on Information Theory, 2005. ISIT 2005., pp. 1848–1852, IEEE, 2005.

[42] S. Acedanski, S. Deb, M. Médard, and R. Koetter, "How good is random linear coding based distributed networked storage," in Workshop on Network Coding, Theory and Applications, pp. 1–6, 2005.

[43] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Médard, and N. Ratnakar, "Network coding for wireless applications: A brief tutorial," IWWAN, 2005.

[44] A. El Fawal, K. Salamatian, D. Cavin, Y. Sasson, and J.-Y. Le Boudec, "A framework for network coding in challenged wireless network," in MobiSys 2006, no. LCA-POSTER-2006-004, 2006.

[45] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," in Packet Video 2007, pp. 191–200, IEEE, 2007.

[46] N. Sundaram, P. Ramanathan, and S. Banerjee, "Multirate media streaming using network coding," in Proc. 43rd Allerton Conference on Communication, Control, and Computing, Monticello, IL, Sep, 2005.

[47] P. Frossard, J. C. De Martin, and M. R. Civanlar, "Media streaming with network diversity," Proceedings of the IEEE, vol. 96, no. 1, pp. 39–53, 2008.

[48] T.-Y. Chung, C.-C. Wang, Y.-M. Chen, and Y.-H. Chang, "Pnecos: a peer-to-peer network coding streaming system," in Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08. IEEE International Conference on, pp. 379–384, IEEE, 2008.

[49] M. Wang and B. Li, "Network coding in live peer-to-peer streaming," IEEE Transactions on Multimedia, vol. 9, no. 8, pp. 1554–1567, 2007.

[50] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," IEEE Journal on Selected Areas in Communications, vol. 25, no. 9, pp. 1655–1666, 2007.

[51] Y. Liu, Y. Peng, W. Dou, and B. Guo, "Network coding for peer-to-peer live media streaming," in 2006 Fifth International Conference on Grid and Cooperative Computing (GCC'06), pp. 149–155, IEEE, 2006.

[52] C. Feng and B. Li, "On large-scale peer-to-peer streaming systems with network coding," in Proceedings of the 16th ACM international conference on Multimedia, pp. 269–278, ACM, 2008.

[53] K. Nguyen, T. Nguyen, and S.-c. Cheung, "Peer-to-peer streaming with hierarchical network coding," in 2007 IEEE International Conference on Multimedia and Expo, pp. 396–399, IEEE, 2007.

[54] D. G. Padmavathi, M. Shanmugapriya, et al., "A survey of attacks, security mechanisms and challenges in wireless sensor networks," arXiv preprint arXiv:0909.0576, 2009.

[55] L. Lima, J. P. Vilela, P. F. Oliveira, and J. Barros, "Network coding security: Attacks and countermeasures," arXiv preprint arXiv:0809.1366, 2008.

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

[57] M. Bloch and J. Barros, Physical-layer security: from information theory to security engineering. Cambridge University Press, 2011.

[58] D. Katabi, S. Katti, W. Hu, H. Rahul, and M. Medard, "On practical network coding for wireless environments," in Communications, 2006 International Zurich Seminar on, pp. 84–85, IEEE, 2006.

[59] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," 2003.

[60] J. Liu, D. Goeckel, and D. Towsley, "Bounds on the gain of network coding and broadcasting in wireless networks," in IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, pp. 724–732, IEEE, 2007.

[61] J. Liu, D. Goeckel, and D. Towsley, "The throughput order of ad hoc networks employing network coding and broadcasting," in MILCOM 2006-2006 IEEE Military Communications conference, pp. 1–7, IEEE, 2006.

[62] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network coding-based broadcast in mobile ad-hoc networks," in IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, pp. 1739–1747, IEEE, 2007.

[63] X. Li, C.-C. Wang, and X. Lin, "Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints," in INFOCOM, 2010 Proceedings IEEE, pp. 1–5, IEEE, 2010.

[64] Y. Wu, P. A. Chou, S.-Y. Kung, et al., "Information exchange in wireless networks with network coding and physical-layer broadcast," tech. rep., MSR-TR-2004, 2005.

[65] R. Prasad, H. Wu, D. Perkins, and N.-F. Tzeng, "Local topology assisted xor coding in wireless mesh networks," in 2008 The 28th International Conference on Distributed Computing Systems Workshops, pp. 156–161, IEEE, 2008.

[66] C. Chekuri, C. Fragouli, and E. Soljanin, "On average throughput and alphabet size in network coding," IEEE/ACM Transactions on Networking (TON), vol. 14, no. SI, pp. 2410–2424, 2006.

[67] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, pp. 1028–1036, IEEE, 2007.

[68] S. Maheshwar, Z. Li, and B. Li, "Bounding the coding advantage of combination network coding in undirected networks," IEEE Transactions on Information Theory, vol. 58, no. 2, pp. 570–584, 2012.

[69] D. Platz, D. H. Woldegebreal, and H. Karl, "Random network coding in wireless sensor networks: Energy efficiency via cross-layer approach," in 2008 IEEE 10th International Symposium on Spread Spectrum Techniques and Applications, pp. 654–660, IEEE, 2008.

[70] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," ACM SIGCOMM Computer Communication Review, vol. 36, no. 1, pp. 63–68, 2006.

[71] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 2, pp. 585–594, IEEE, 2000.

[72] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., vol. 3, pp. 1607–1617, IEEE, 2005.

[73] Y. Xi and E. M. Yeh, "Distributed algorithms for minimum cost multicast with network coding," IEEE/ACM Transactions on Networking, vol. 18, no. 2, pp. 379–392, 2010.

[74] W.-L. Yeow, A. T. Hoang, and C.-K. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in INFOCOM 2009, IEEE, pp. 190–198, IEEE, 2009.

[75] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," IEEE Transactions on communications, vol. 53, no. 11, pp. 1906–1918, 2005.

[76] J. Choi and D. To, "Energy efficiency of harq-ir for two-way relay systems with network coding," in European Wireless, 2012. EW. 18th European Wireless Conference, pp. 1–5, VDE, 2012.

[77] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient broadcasting using network coding," IEEE/ACM Transactions on Networking (TON), vol. 16, no. 2, pp. 450–463, 2008.

[78] O. Trullols-Cruces, J. M. Barcelo-Ordinas, and M. Fiore, "Exact decoding probability under random linear network coding," IEEE communications letters, vol. 15, no. 1, pp. 67–69, 2011.

[79] R. A. Costa, D. Munaretto, J. Widmer, and J. Barros, "Informed network coding for minimum decoding delay," in 2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, pp. 80–91, IEEE, 2008.

[80] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," IEEE Transactions on Information Theory, vol. 54, no. 12, pp. 5511–5524, 2008.

[81] W.-L. Yeow, A. T. Hoang, and C.-K. Tham, "On average packet delay bounds and loss rates of network-coded multicasts over wireless downlinks," in 2009 IEEE International Conference on Communications, pp. 1–6, IEEE, 2009.

[82] M. Nistor, D. E. Lucani, T. T. Vinhoza, R. A. Costa, and J. Barros, "On the delay distribution of random linear network coding," IEEE Journal on Selected Areas in Communications, vol. 29, no. 5, pp. 1084–1093, 2011.

[83] B. Han and S. Lee, "Efficient packet error rate estimation in wireless networks," in Testbeds and Research Infrastructure for the Development of Networks and

Communities, 2007. TridentCom 2007. 3rd International Conference on, pp. 1–9, IEEE, 2007.

[84] Z. Yang, M. Li, and W. Lou, "R-code: Network coding-based reliable broadcast in wireless mesh networks," Ad Hoc Networks, vol. 9, no. 5, pp. 788–798, 2011.

[85] Z. Wang and M. Hassan, "Blind xor: Low-overhead loss recovery for vehicular safety communications," IEEE Transactions on Vehicular Technology, vol. 61, no. 1, pp. 35–45, 2012.

[86] W. Fang, F. Liu, Z. Liu, L. Shu, and S. Nishio, "Reliable broadcast transmission in wireless networks based on network coding," in Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, pp. 555–559, IEEE, 2011.

[87] M. Ghaderi, D. Towsley, and J. Kurose, "Network coding performance for reliable multicast," in MILCOM 2007-IEEE Military Communications Conference, pp. 1–7, IEEE, 2007.

[88] G. Breed, "Bit error rate: Fundamental concepts and measurement issues," High Frequency Electronics, vol. 2, no. 1, pp. 46–47, 2003.

[89] O. Awoniyi and F. A. Tobagi, "Packet error rate in ofdm-based wireless lans operating in frequency selective channels," in Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, pp. 1–13, IEEE, 2006.

[90] F.-C. Kuo, K. Tan, X. Li, J. Zhang, and X. Fu, "Xor rescue: exploiting network coding in lossy wireless networks," in 2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 1–9, IEEE, 2009.

[91] Z. Wang, M. Hassan, and T. Moors, "Efficient loss recovery using network coding in vehicular safety communication," in 2010 IEEE Wireless Communication and Networking Conference, pp. 1–6, IEEE, 2010.

[92] Z. Guo, B. Wang, P. Xie, W. Zeng, and J.-H. Cui, "Efficient error recovery with network coding in underwater sensor networks," Ad Hoc Networks, vol. 7, no. 4, pp. 791–802, 2009.

[93] M. Médard and A. Sprintson, "Network coding: Fundamentals and applications, ser," 2012.

[94] P. Li, S. Guo, S. Yu, and A. V. Vasilakos, "Reliable multicast with pipelined network coding using opportunistic feeding and routing," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 12, pp. 3264–3273, 2014.

[95] E. Ancillotti, R. Bruno, and M. Conti, "Reliable data delivery with the ietf routing protocol for low-power and lossy networks," IEEE Transactions on Industrial Informatics, vol. 10, no. 3, pp. 1864–1877, 2014.

[96] 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.

[97] P. Larsson and N. Johansson, "Multi-user arq," in 2006 IEEE 63rd Vehicular Technology Conference, vol. 4, pp. 2052–2057, IEEE, 2006.

[98] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, IEEE, 2008.

[99] X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," IEEE Journal on Selected Areas in Communications, vol. 27, no. 5, pp. 622–634, 2009.

[100] L. Lima, "Network coding security algebraic properties and lightweight solutions," 2010.

[101] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network coding theory: A survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 1950–1978, 2013.

[102] V. Nazaritalooki, "Reliable and energy efficient routing for ad hoc networks," 2014.

[103] J. Dong, R. Curtmola, C. Nita-Rotaru, and D. K. Yau, "Pollution attacks and defenses in wireless interflow network coding systems," IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 5, pp. 741–755, 2012.

[104] A. Le and A. Markopoulou, "Locating byzantine attackers in intra-session network coding using spacemac," in 2010 IEEE International Symposium on Network Coding (NetCod), pp. 1–6, IEEE, 2010.

[105] E. K. Wang, Y. Ye, X. Xu, S.-M. Yiu, L. C. K. Hui, and K.-P. Chow, "Security issues and challenges for cyber physical system," in Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, pp. 733–738, IEEE Computer Society, 2010.

[106] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems," Pervasive and Mobile Computing, vol. 7, no. 4, pp. 397–413, 2011.

[107] J. Sen, "A survey on wireless sensor network security," arXiv preprint arXiv:1011.1529, 2010.

[108] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, Handbook of applied cryptography. CRC press, 2010.

[109] M. Anand, Z. Ives, and I. Lee, "Quantifying eavesdropping vulnerability in sensor networks," in Proceedings of the 2nd international workshop on Data management for sensor networks, pp. 3–9, ACM, 2005.

[110] A. Mishra and K. M. Nadkarni, "Security in wireless ad hoc networks," in The handbook of ad hoc wireless networks, pp. 499–549, CRC Press, Inc., 2003.

[111] A. J. Newell, R. Curtmola, and C. Nita-Rotaru, "Entropy attacks and countermeasures in wireless network coding," in Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, pp. 185–196, ACM, 2012.

[112] H. S. Chiu and K.-S. Lui, "Delphi: wormhole detection mechanism for ad hoc wireless networks," in Wireless pervasive computing, 2006 1st international symposium on, pp. 6–pp, IEEE, 2006.

[113] M. Adeli and H. Liu, "Secure network coding with minimum overhead based on hash functions," Communications Letters, IEEE, vol. 13, no. 12, pp. 956–958, 2009.

[114] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in Public Key Cryptography–PKC 2009, pp. 68–87, Springer, 2009.

[115] E. Kehdi and B. Li, "Null keys: Limiting malicious attacks via null space properties of network coding," in INFOCOM 2009, IEEE, pp. 1224–1232, IEEE, 2009.

[116] N. Cai, R. W. Yeung, et al., "Network error correction, ii: Lower bounds," Communications in Information & Systems, vol. 6, no. 1, pp. 37–54, 2006.

[117] R. W. Yeung, N. Cai, et al., "Network error correction, i: Basic concepts and upper bounds," Communications in Information & Systems, vol. 6, no. 1, pp. 19–35, 2006.

[118] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," Information Theory, IEEE Transactions on, vol. 54, no. 8, pp. 3579–3591, 2008.

[119] W. Qiao, J. Li, and J. Ren, "An efficient error-detection and error-correction (edec) scheme for network coding," in Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, pp. 1–5, IEEE, 2011.

[120] Y. Li and J. Lui, "Identifying pollution attackers in network-coding enabled wireless mesh networks," in Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on, pp. 1–6, IEEE, 2011.

[121] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 255–265, ACM, 2000.

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

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

[124] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," Wireless networks, vol. 8, no. 5, pp. 521–534, 2002.

[125] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of the 9th ACM conference on Computer and communications security, pp. 41–47, ACM, 2002.

[126] C. Cheng, T. Jiang, and Q. Zhang, "Tesla-based homomorphic mac for authentication in p2p system for live streaming with network coding," Selected Areas in Communications, IEEE Journal on, vol. 31, no. 9, pp. 291–298, 2013.

[127] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," in Proceedings of the second ACM conference on Wireless network security, pp. 111–122, ACM, 2009.

[128] M. Kim, L. Lima, F. Zhao, J. Barros, M. Médard, 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, no. 5, pp. 692–702, 2010.

[129] A. Le and A. Markopoulou, "On detecting pollution attacks in inter-session network coding," in INFOCOM, 2012 Proceedings IEEE, pp. 343–351, IEEE, 2012.

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

[131] 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.

[132] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in wireless network coding," ACM Transactions on Information and System Security (TISSEC), vol. 14, no. 1, p. 7, 2011.

[133] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," in INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 2, pp. 708–716, IEEE, 1999.

[134] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," IEEE Communications magazine, vol. 32, no. 9, pp. 33–38, 1994.

[135] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," ACM Transactions on Information and System Security (TISSEC), vol. 8, no. 2, pp. 228–258, 2005.

[136] D. Petrović, K. Ramchandran, and J. Rabaey, "Overcoming untuned radios in wireless networks with network coding," IEEE/ACM Transactions on Networking (TON), vol. 14, no. SI, pp. 2649–2657, 2006.

[137] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," 2003.

[138] Y. Yang, C. Zhong, Y. Sun, and J. Yang, "Network coding based reliable disjoint and braided multipath routing for sensor networks," Journal of Network and Computer Applications, vol. 33, no. 4, pp. 422–432, 2010.

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

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