Network Coding based Information Security in Multi-hop Wireless Networks

by

Yanfei Fan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Multi-hop Wireless Networks (MWNs) represent a class of networks where messages are forwarded through multiple hops of wireless transmission. Applications of this newly emerging communication paradigm include asset monitoring wireless sensor networks (WSNs), command communication mobile ad hoc networks (MANETs), community- or campus-wide wireless mesh networks (WMNs), etc.

Information security is one of the major barriers to the wide-scale deployment of MWNs but has received little attention so far. On the one hand, due to the open wireless channels and multi-hop wireless transmissions, MWNs are vulnerable to various information security threats such as eavesdropping, data injection/modification, node compromising, traffic analysis, and flow tracing. On the other hand, the characteristics of MWNs including the vulnerability of intermediate network nodes, multi-path packet forwarding, and limited computing capability and storage capacity make the existing information security schemes designed for the conventional wired networks or single-hop wireless networks unsuitable for MWNs. Therefore, newly designed schemes are highly desired to meet the stringent security and performance requirements for the information security of MWNs.

In this research, we focus on three fundamental information security issues in MWNs: efficient privacy preservation for source anonymity, which is critical to the information security of MWNs; the traffic explosion issue, which targets at preventing denial of service (DoS) and enhancing system availability; and the cooperative peer-to-peer information exchange issue, which is critical to quickly achieve maximum data availability if the base station is temporarily unavailable or the service of the base station is intermittent. We have made the following three major contributions.

Firstly, we identify the severe threats of traffic analysis/flow tracing attacks to the information security in network coding enabled MWNs. To prevent these attacks and achieve source anonymity in MWNs, we propose a network coding based privacy-preserving scheme. The unique “mixing” feature of network coding is exploited in the proposed scheme to

confuse adversaries from conducting advanced privacy attacks, such as time correlation, size correlation, and message content correlation. With homomorphic encryption functions, the proposed scheme can achieve both privacy preservation and data confidentiality, which are two critical information security requirements.

Secondly, to prevent traffic explosion and at the same time achieve source unobservability in MWNs, we propose a network coding based privacy-preserving scheme, called SUNC (Source Unobservability using Network Coding). Network coding is utilized in the scheme to automatically absorb dummy messages at intermediate network nodes, and thus, traffic explosion induced denial of service (DoS) can be naturally prevented to ensure the system availability. In addition to ensuring system availability and achieving source unobservability, SUNC can also thwart internal adversaries.

Thirdly, to enhance the data availability when a base station is temporarily unavailable or the service of the base station is intermittent, we propose a cooperative peer-to-peer information exchange scheme based on network coding. The proposed scheme can quickly accomplish optimal information exchange in terms of throughput and transmission delay.

For each research issue, detailed simulation results in terms of computational overhead, transmission efficiency, and communication overhead, are given to demonstrate the efficacy and efficiency of the proposed solutions.

Acknowledgements

I would like to express my deepest gratitude to Professor Xuemin (Sherman) Shen, my advisor. I thank you for your continuing guidance and support during my four years of research. Your sharp sense of research direction, great enthusiasm, and strong belief in the potential of this research has been a tremendous force for the completion of this work. I have learned so many things from you, including doing research, writing papers, giving seminars, and many more. Most importantly, I thank you for encouraging me in each step of my growing path. Your strong belief in me and continuous encouragement have made this research such an exciting experience that our collaboration finally produces something that we are both proud of.

This thesis would not have been possible without the assistance of many people. I would also like to express my extreme appreciation to my thesis committee members: Professor Baochun Li, Professor Liping (Lee) Fu, Professor Sagar Naik and Professor Liang-Liang Xie. They have contributed their precious time to read my thesis, and provided valuable suggestions and comments that helped to improve the quality of this thesis.

I would also like to thank my colleagues and friends at Security Discussion Group of BBCR Lab. My discussions with Xiaodong Lin, Yixin Jiang, Minghui Shi, Jiming Chen, Rongxing Lu, Haojin Zhu, Chenxi Zhang, Xiaoting Sun, Yipin Sun, Xiaohui Liang, Sanaa Taha, Mohamed Elsalih Mahmoud, Albert Wasef, Hao (Tom) Luan, Mohammad Towhidul Islam, Mahdi Asefi, and Fangqin Liu have given me many inspirations. I feel so fortunate to work with many wonderful people in BBCR Lab, such as Stanley Liu, Bin Lin, Bong Choi, Ho Ting (Anderson) Cheng, Khadige Abboud, and more. I thank them all.

There are many other people whose names are not mentioned here. It does not mean that I have forgotten you or your help. It is a privilege for me to work and share life with so many bright and energetic people. Your talent and friendship have made Waterloo such a great place to live.

I would never get this far without the support of my parents. Thank you for always believing in me and supporting me. Your love and encouragement have been and will always be a great source of inspiration in my life.

Table of Contents

AUTHOR'S DECLARATION ii
Abstract iii
Acknowledgements v
Table of Contents vii
List of Figures x
List of Tables xi
List of Abbreviations xii
Chapter 1 Introduction 1
1.1 MWN Motivating Application Scenarios 3
1.1.1 Asset Monitoring Wireless Sensor Networks 3
1.1.2 Command Communication Mobile Ad hoc Networks 4
1.1.3 Wireless Mesh Networks 5
1.2 Research Issues in MWNs: Non-Security Aspects 5
1.3 Information Security in MWNs: Research Motivations and Contributions 6
1.3.1 Motivations 6
1.3.2 Contributions 7
1.4 Outline of This Thesis 8
Chapter 2 Information Security of MWNs: Threats, Requirements, Characteristics, and Challenges. 11
2.1 Threats to MWNs 11
2.2 Information Security Requirements 12
2.2.1 Confidentiality 13
2.2.2 Integrity 13
2.2.3 Availability 14
2.2.4 Privacy 14
2.3 Characteristics of MWNs 16
2.3.1 Open Wireless Channels 16
2.3.2 Vulnerability of Intermediate Nodes 17
2.3.3 Multi-path Forwarding 18
2.3.4 Resource Constraints 18
2.4 Identified Research Challenges 19

2.4.1 Thwarting Traffic Analysis Attacks 19
2.4.2 Thwarting Internal Adversaries 20
2.4.3 Preventing Traffic Explosion 20
2.4.4 Trade-off between Security and Performance 21
2.5 Preliminaries 21
2.5.1 Network Coding 21
2.5.2 Homomorphic Encryption Functions 24
2.6 Summary 24
Chapter 3 Network Coding Based Privacy Preservation against Traffic Analysis in Multi-hop Wireless Networks 25
3.1 Threat Models 29
3.2 Network Coding Based Privacy-Preserving Scheme for MWNs 30
3.2.1 The Proposed Privacy-Preserving Scheme 30
3.2.2 Invertibility of a GEM 33
3.3 Security Analysis 36
3.4 Performance Evaluation and Optimization 38
3.4.1 Invertible Probability 38
3.4.2 Computational Overhead 41
3.4.3 Performance Optimization 42
3.5 Related Work 45
3.6 Summary 46
Chapter 4 Preventing Traffic Explosion and Achieving Source Unobservability in Multi-hop Wireless Networks using Network Coding 47
4.1 Threat Models 51
4.2 The Proposed Privacy-Preserving Scheme 51
4.2.1 Parameter Setting 52
4.2.2 Generating Dummy Traffic 53
4.2.3 Embedding Real Traffic 54
4.2.4 Recovering Real Messages 56
4.3 Security Analysis 57
4.3.1 Information Security and Privacy 57
4.3.2 Adversaries and Attacks 58

4.4 Performance Evaluation 60
4.4.1 Invertible Probability 60
4.4.2 Communication Efficiency 61
4.4.3 Computational Overhead 62
4.5 Summary 68
Chapter 5 Cooperative Peer-to-Peer Information Exchange in Network Coding Enabled Wireless Networks 71
5.1 Network Model 73
5.2 Investigations on Information Exchange Principles 74
5.3 The Proposed PIE Scheme 76
5.3.1 The PIE Scheme 76
5.3.2 Discussions 80
5.4 Performance Evaluation 81
5.4.1 Transmission Efficiency 82
5.4.2 Computational Overhead 84
5.5 Summary 85
Chapter 6 Conclusions and Future Work 87
6.1 Conclusions 87
6.2 Future Work 88
6.2.1 Privacy Preservation for DTNs 88
6.2.2 Enhancing Data Availability for DTNs 89
Bibliography 91

List of Figures

Figure 2.1: Random Coding (Mixing) at Intermediate Nodes 22
Figure 3.1: Privacy Threats in MWNs 26
Figure 3.2: Attack Model: (a) Outside Attacker, (b) Inside Attacker 29
Figure 3.3: Homomorphic Encryption on Packet Tags 31
Figure 3.4: Packet Tagging before Source Encoding 31
Figure 3.5: Privacy Enhancement in terms of Computational Complexity 37
Figure 3.6: Invertible Probability vs. Field Size 40
Figure 3.7: Invertible Probability vs. Total Times of Random Coding 41
Figure 3.8: Computational Overhead vs. Size of Algebraic Structure 44
Figure 4.1: Attack Model: External Attacker and Internal Attacker 51
Figure 4.2: Degrading Factor of Invertible Probability vs. λ / μ 61
Figure 4.3: Linear Dependence Analysis 64
Figure 4.4: The Average Complexity of Linear Dependence Analysis 66
Figure 4.5: Average Computational Overhead 68
Figure 5.1: Flow Chart of PIE 77
Figure 5.2: Efficiency vs. the Number of Peers 81
Figure 5.3: Efficiency vs. the Number of Blocks 82
Figure 5.4: PIE vs. Rarest First 83
Figure 5.5: Efficiency vs. Sparsity 84
Figure 5.6: Computational Overhead vs. Sparsity (PIE and Rarest First) 85

List of Tables

Table 5.1: List of Notations 74

List of Abbreviations

AP	Access Point
BS	Base Station
CPR	Cooperative Peer-to-peer Repair
DoS	Denial of Service
DF	Degrading Factor
GEM	Global Encoding Matrix
GEV	Global Encoding Vector
HEF	Homomorphic Encryption Function
LEV	Local Encoding Vector
MANET	Mobile Ad hoc Network
MWN	Multi-hop Wireless Network
P2P	Peer-to-Peer
PDA	Personal Digital Assistant
PFS	Proxy-based Filtering Scheme
PII	Personal Identifiable Information
TFS	Tree-based Filtering Scheme
VoD	Video on Demand
VoIP	Voice over IP
WBAN	Wireless Body Area Network
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network

Chapter 1
Introduction

With the development of the Internet and the boom of various applications, wireless networks have experienced an explosive expansion. Wireless Local Area Network (WLAN) technologies, represented by Wi-Fi [1], are so successful as to be accepted quickly by all over the world. Wireless Metropolitan Area Networks (WMANs) such as WiMAX [2] are also developing quickly on the way of being widely deployed. Wireless Personal Area Networks (WPANs) and Wireless Body Area Networks (WBANs) are very promising in the applications of personal entertainment and E-health [3]. Moreover, other emerging wireless technologies [4] have also attracted a lot of attentions from both the academia and industry.

Wireless networks have a lot of advantages for them to be adopted so quickly by users all around the world. The main advantages include the following features.

- ✧ ***Convenience:*** The open nature of wireless channels allows users to access network resources conveniently from any location within radio coverage. With the increasing popularity of portable devices such as laptops, PDAs, netbook, and smart phones, this feature is particularly important.
- ✧ ***Mobility:*** With the emergence of public wireless networks, such as a campus wireless network [5], users can maintain a steady connection with their desired networks to access the Internet without intermission even when they are in a moving car.
- ✧ ***Low Cost:*** Wireless networking hardware is relatively cheap especially when compared with the potentially increasing cost of physical cables in wired networks.
- ✧ ***Expandability:*** Wireless networks can serve a suddenly-increased number of clients with the existing equipment. On the contrary, additional clients would require additional wiring in a wired network.

- ✧ ***Easy Deployment:*** Compared with wired networks, which have the cost and complexity of actual physical cables winding through floors, the initial setup of even an infrastructure-based wireless network requires only a single access point.

For a given networking situation, current wireless solutions may not be applicable for a number of reasons. Most of them are related to the limitations of current technologies.

- ➢ ***Range:*** The current effective range of an 802.11b/g network with standard equipment is on the order of tens of meters. Although it is sufficient for an average home or a small office, it is insufficient for a larger structure such as a community. New technologies are being developed, and multi-hop wireless networking is very promising in extending the radio coverage range.
- ➢ ***Security:*** The open nature of wireless channels not only brings convenience, but also makes the channels vulnerable to various attacks such as eavesdropping and jamming [6]. In a wired network, an adversary has to overcome the physical obstacle to tap into the actual wires, but this is not an obstacle any more in wireless networks.
- ➢ ***Reliability:*** Wireless signals are broadcast in nature, causing a wide variety of interference. In addition, the complex propagation effect, such as fading and multipath, makes wireless networks more unreliable.
- ➢ ***Speed:*** The speed of most wireless networks (typically 1-54 Mbps) is relatively slow compared with wired networks (100 Mbps to several Gbps). New wireless standards such as 802.11n are to address this limitation and will support the peak throughput up to 200 Mbps.

In the future, wireless communication will keep developing, and wireless networks will get more extensive deployment. A variety of new technologies will be exploited to overcome the limitations. New evolutions on modulation and multiplexing technologies will enhance the reliability and speed of wireless channels. New security and privacy schemes will diminish the vulnerability of wireless networks. Multi-hop Wireless Networks (MWNs) [7]

are regarded as a promising solution for extending the limit radio coverage range of wireless networks. In addition, through multi-path packet forwarding, they are also promising in enhancing reliability, speed, and security.

1.1 MWN Motivating Application Scenarios

In addition to the improvement for existing wireless networks, MWNs motivate some new application scenarios. In this section, we will briefly introduce the following three MWN motivating application scenarios.

1.1.1 Asset Monitoring Wireless Sensor Networks

Wireless Sensor Networks (WSNs) [8] are promising in many application scenarios, and thus attract much attention from both the academia and the industry. Most current research on WSNs is focused on the system architecture, node deployment, and communication protocols of multi-hop WSNs. Multi-hop wireless networking is critical to WSNs since it is not only the requirements of applications such as data collection but also the requirements of network connectivity and node battery life.

An important application driven by WSN and MWN technologies in the future will be the applications that monitor a valuable asset. For example, sensors can be deployed in natural habitats to monitor endangered animals [9]. When an endangered animal such as an elephant or a panda passes by a sensor, the sensor perceives and records the activity of this animal. The data storage capacity of each sensor is limited, and the sensed data may be lost if it cannot be collected in time. Data collection can be done in two kinds of methods. The first method is that the administrator of the natural habitat can drive a car to collect the data manually. The second method is to utilize MWN technologies to form the deployed sensors into a WSN; then, some sink nodes can be deployed to collect the sensed data through multi-hop wireless transmission automatically. In this thesis, we consider only the second data collection method as well as the corresponding multi-hop WSN used for data collection. In

these asset monitoring applications, security and privacy issues are very critical, and the location privacy of source node is especially important. The details of security and privacy threats to asset monitoring WSNs will be explored in Chapter 2.1.

1.1.2 Command Communication Mobile Ad hoc Networks

Mobile Ad hoc Networks (MANET) have many attractive applications, such as emergency communication, disaster rescue, battlefield communication, and some commercial applications [10]. Multi-hop wireless networking technologies can assist MANET to reach farther communication range more reliably, thus to achieve a better mission completion in the above applications. A command communication system can be transplanted to MANETs, forming a command communication MANET.

Command communication MANET is very suitable for battlefield situations since no infrastructure can be warranted in those scenarios [11]. A command communication MANET may be comprised of several or tens of mobile nodes, which can be the carry-on wireless devices of equipped soldiers. These nodes can communicate with each other through direct transmission or multi-hop forwarding. In this network, one or two nodes may be the critical nodes, which are probably the commanders of the military group. Most command communication activities take place between the commanders and soldiers. Soldiers may report a real-time situation of the battlefield, and commanders may issue an order to soldiers to indicate what to do for the next step. This unique communication pattern may leak the location privacy of critical nodes. In addition, due to the hostile battlefield environment, command communication MANETs are very prone to malicious attacks. A simple vulnerability may be utilized by adversaries, thus probably leading to soldier casualties or mission impossible. Thus, security and privacy protection is very critical to command communication MANETs, and various threats will be investigated in details in Chapter 2.1.

1.1.3 Wireless Mesh Networks

Wireless Mesh Networks (WMNs) have been extensively regarded as a key technology for next-generation wireless networking [12]. Assisted with multi-hop wireless networking technologies, WMNs have inspired many novel application scenarios such as community networking, enterprise networking, building automation, broadband home networking, and high-speed metropolitan area network.

WMNs are dynamically self-organized and self-configured, and nodes in WMNs automatically establish ad hoc networks and maintain mesh connectivity. Through maintaining the wireless mesh backbone, WMNs distinguish their capacities from ad hoc networks. Through multi-hop forwarding, the same coverage can be achieved by WMNs with much lower transmission power. However, the available MAC and routing protocols are not scalable for multi-hop forwarding, and throughput drops significantly as the number of hops increases. In this sense, multi-hop wireless networking technologies are the kernel of WMNs.

1.2 Research Issues in MWNs: Non-Security Aspects

Although MWNs are promising in expanding the radio coverage range for the future wireless networks, many other requirements should be satisfied before they can be widely adopted.

With the spread of radio coverage and the increase of wireless users, wireless networks need to accommodate more and more portable devices. The computing power and transmission rate of a MWN should satisfy the requirement of emerging various applications [13]. New applications, especially the live multimedia applications such as IPTV and Video on Demand, will certainly raise the requirements of MWN performance.

Compared with traditional single-hop wireless networks such as cellular networks, MWNs have higher requirements in mobility management [14]. Due to not only the increase of wireless users but also the decrease of single AP coverage, handoff will be happening more frequently in MWNs. For the sake of live applications which are sensitive to handoff delay, mobility management in MWNs needs to achieve fast and seamless handoff.

Due to the broadcast propagation fashion and the interference effect of wireless signal, the robustness and reliability [15] is an important performance metric for WMNs, especially when they encounters malicious attacks. The open nature of wireless channels makes malicious attacks much easier, and the propagation fashion makes wireless signal more vulnerable to interference. To enhance the robustness and reliability of MWNs, not only is the progress in signal modulation technologies essential, but the upper layer resource allocation and routing technologies are indispensable.

1.3 Information Security in MWNs: Research Motivations and Contributions

The term "information security" means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, availability, and privacy [16]. Confidentiality means preserving authorized restrictions on access and disclosure for proprietary information; integrity means guarding against improper information modification or destruction and includes ensuring information non-repudiation and authenticity; availability means ensuring timely and reliable access to and use of information [17]; and privacy means preventing unauthorized access, use, or disclosure on private information, including personal identity information, personal location information, and personal activity information. We will discuss our motivations and contributions on information security research in the following parts.

1.3.1 Motivations

With the evolvement of the Internet, a common user has already been able to obtain a great computing power and massive data access capability which was unimaginable only a few years ago. These progresses will comprise a security threat if they are in the hands of a malicious user. Significant computing power can break a key or password space in a brute force way easily, only if the encrypted text or the hashed password is readily available;

substantial data access may assist to reveal the confidential or private information; the combination of significant computing power and substantial data access, however, will lead to a severe threat to security and privacy preservation of MWNs.

Security and privacy is a major concern of MWNs. For example, attacks such as traffic analysis and flow tracking may leak the location information of users and thus expose the location privacy. Security and privacy issues not only threaten the current users of wireless technologies, but also become one of the main obstacles against the wide adoption of new wireless technologies such as multi-hop wireless networking.

On the one hand, the open and shared nature of wireless channels makes MWNs vulnerable to various malicious attacks, such as eavesdropping, data injection/modification, node compromising, and traffic analysis. These malicious attacks may pose a great threat to the security and privacy of a networking system. On the other hand, the research on the information security in MWNs is very challenging due to their characteristics such as open wireless channels, vulnerability of intermediate network nodes, multi-path packet forwarding, and various resource constraints. These characteristics make existing information security schemes unsuitable for MWNs.

To provide security and privacy protection for wireless users, also to sweep away the obstacles on the way of the wide adoption of emerging wireless technologies, we put our research emphasis on security and privacy issues in MWNs. We also take performance related issues into consideration since performance issues are also obstacles for MWNs to be extensively adopted.

1.3.2 Contributions

This thesis aims at developing novel solutions to a number of challenging security issues in MWNs. The major contribution of this thesis is summarized as follows.

- ✧ We identified and summarized the unique information security characteristics of MWNs, which distinguish the information security of MWNs from that of other networks.
- ✧ We identified the unique features of network coding for the information security of MWNs, since network coding is one of the major techniques based on which we develop our solutions.
- ✧ We proposed an efficient network coding based privacy preservation scheme against traffic analysis and flow tracing in MWNs.
- ✧ We proposed a lightweight scheme to prevent traffic explosion and achieve source unobservability in MWNs using network coding.
- ✧ We discussed the information availability in hybrid wireless networks and proposed a cooperative peer-to-peer information exchange scheme in network coding enabled hybrid wireless networks.

1.4 Outline of This Thesis

The remainder of this thesis is organized as follows.

Chapter 2 gives the general threats to MWNs and the basic requirements of information security. We also summarize the characteristics of MWNs and identify challenges to achieve information security in MWNs. Then, the network coding model is reviewed, followed by a summary.

In Chapter 3, we propose a novel efficient network coding based privacy-preserving scheme against traffic analysis and flow tracing in MWNs. With homomorphic encryption on Global Encoding Vectors (GEVs), the proposed scheme offers two significant privacy-preserving features, packet flow untraceability and message content confidentiality, for efficiently thwarting the traffic analysis attacks. Moreover, the proposed scheme keeps the random coding feature, and each sink can recover the source packets by inverting the GEVs

with a very high probability. Theoretical analysis and simulative evaluation demonstrate the validity and efficiency of the proposed scheme.

Chapter 4 proposes a novel scheme, called SUNC (Source Unobservability using Network Coding), to prevent traffic explosion while achieving source unobservability. With network coding, specially designed dummy messages can be absorbed at intermediate forwarders, and, thus, traffic explosion can be naturally prevented. In addition, SUNC can achieve privacy in a stronger threat model. Assisted by homomorphic encryption, SUNC can offer forwarder blindness, which is an important privacy property for thwarting internal attackers. Security analysis and performance evaluation demonstrate the efficacy and efficiency of the proposed SUNC.

In Chapter 5, we study the issue of scheduling transmission opportunities among nodes (peers) to achieve higher network throughput and lower transmission delay for network coding enabled wireless networks. By conducting an in-depth investigation on the scheduling principles, we propose a cooperative Peer-to-peer Information Exchange (PIE) scheme with an efficient and light-weight scheduling algorithm. PIE can not only fully exploit the broadcast nature of wireless channels, but also take advantage of cooperative peer-to-peer information exchange. Qualitative analysis and extensive simulations demonstrate the effectiveness and efficiency of PIE.

Chapter 6 concludes this thesis and gives some future work.

Chapter 2
Information Security of MWNs: Threats, Requirements, Characteristics, and Challenges

With the expanding of the Internet and the thriving of various Internet applications, wireless networks such as Wi-Fi are widely adopted due to its convenience and low cost. However, wireless networks still suffer from their current technical shortcomings such as limited radio coverage, low system reliability, as well as lack of security and privacy protection. Multi-hop wireless technologies are promising to overcome these shortcomings. The technologies can extend the radio coverage by deploying new Access Points (APs) around the existing networking infrastructure, which thus can provide Internet access to larger areas. Multi-hop wireless technologies can also enhance the robustness and reliability of a wireless network by choosing multiple forwarding paths. Due to these benefits, multi-hop wireless technologies attract a lot of attention from both academia and industry.

In this chapter, we first discuss the possible security threats to MWNs, and then give the basic requirements of information security. Although similar security and privacy objectives have been proposed in existing networking systems, we discuss the difference between the information security of MWNs and that of existing networking systems by identifying and summarizing the characteristics of MWNs. In addition, we identify the challenges to achieve information security in MWNs and propose four research issues related to the information security of MWNs. Since our solutions to the proposed issues are based on network coding, we give a brief review of the network coding model, followed by a summary.

2.1 Threats to MWNs

According to [18], [19], and [20], the possible threats to the information security of MWNs can be summarized as follows.

- *Eavesdropping:* Due to the open nature of wireless channels, eavesdropping can be easily launched by adversaries without any physical wire tapping. Global Eavesdropping is possible since the scale of a MWN may be not large enough or many attackers may collude together to launch it.
- *Data Injection/Modification:* Open wireless channels have no any obstacles to prevent adversaries from injecting data packets. Data modification can also be easily done by intercepting data packets first and then injecting them back to the wireless network.
- *Node Compromising:* Wireless network nodes or mobile stations may be compromised more easily than network routers in wired networks. Once a network node is compromised, secret information such as various keys may be obtained by adversaries. The adversary may stay inside the network node and become an internal attacker, which is much more troublesome than outside attackers.
- *Traffic Analysis/Flow Tracing:* An advanced attack to compromise user privacy is traffic analysis, which can be launched by analyzing the traffic patterns or traffic flows between communication parties. Flow tracing attacks can be launched through advanced techniques such as time correlation, size correlation, or message content correlation. A particular flow tracing attack is the content correlation based back-tracing attack, which can be utilized to compromise the location privacy of source nodes.

These threats to MWNs directly lead to the definitions of information security requirements, which are discussed in the following section.

2.2 Information Security Requirements

Generally, four fundamental requirements are identified for the information security of MWNs: confidentiality, integrity, availability, and privacy.

2.2.1 Confidentiality

Confidentiality is the term used to describe the property of preserving authorized restrictions on access and disclosure for proprietary information. For example, an online banking system is required to keep the account number and password of a user secret when the information is transmitted from the user to the bank. The system may attempt to enforce confidentiality by encrypting the account number and hashing the password, by limiting the places where it might appear (in databases, log files, backups, printed receipts, and so on), and by restricting access to the places where it is stored. If an unauthorized party obtains the account number and password in any way, the confidentiality of the information has been violated.

There are many forms in which confidentiality can be violated. Confidential information displayed on a screen may be peeked; secret data stored in a computer may be stolen; and sensitive information transmitted through a channel may be overheard. In this thesis, we consider only the situation where confidential information is transmitted through open wireless channels. Since the open wireless channels are particularly vulnerable to eavesdropping, the confidentiality of sensitive information may face more malicious threats. Link-to-link encryption can be applied to wireless channels to ensure the confidentiality of sensitive information transmitted through the channels. End-to-end encryption is another choice to ensure the confidentiality of information if remote mutual authentication and secret key exchange are supported.

Confidentiality is necessary (but not sufficient) for preserving the privacy of users. We will discuss privacy preservation in Section 2.2.4.

2.2.2 Integrity

In information security, the property that data cannot be modified without authorization is called “integrity”, which is different from the referential integrity in databases. For example, we can say that the integrity is breached if a computer is infected and damaged by a virus, if

an important data file is deleted accidentally or intentionally, or if some words or values in a file are modified.

Integrity may be breached in many forms with or without malicious intention. The address information of a person may be incorrectly typed into a computer system; information updates to a database may modify data in a wrong way; and data being transmitted in a wireless channel may be altered by an interfering signal. Since wireless channels are vulnerable to interference, the data integrity should be appropriately protected. The integrity requirement should ensure that messages cannot be altered after the propagation through wireless channels. Data integrity can be easily achieved through lightweight hash functions or digital signatures.

2.2.3 Availability

The term “availability” is used to describe the property of an information system that the information must be available when it is needed [17]. In other words, availability means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to transmit it must be functioning correctly. High availability systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, system upgrades, or channel interruptions. Ensuring availability also involves preventing denial-of-service attacks. In this thesis, we consider only the situation of service disruption due to channel interruptions.

2.2.4 Privacy

The term “privacy” is used to describe the state of preventing unauthorized access and disclosure of individual related information. In other words, privacy can be regarded as the confidentiality of personal information. In traditional information security system, privacy is considered as a part of confidentiality. With the development of the Internet and various personal information related applications, privacy issues have gained more and more

attentions from both academia and industry. Thus, in this thesis, we emphasize privacy protection by separating privacy from confidentiality.

As far as the information content is concerned, privacy can be classified into the following four categories: identity privacy [21], data privacy [22], location privacy [23], and activity privacy [24]. Identity privacy is a kind of protection over personal information or personally identifiable information (PII) from third parties and possibly also from the other parties in communication activities [25]. In other words, identity privacy can be regarded as the confidentiality of personal information or personal identifiable information, which is directly related to or indirectly derivable to the identity of an individual. According to [26], identity privacy can be classified into five levels: pseudonymity, unlinkability, anonymity, unobservability, and undetectability.

Data privacy, also known as content privacy, is a special kind of privacy requirements, which is very similar to data confidentiality [27]. Confidentiality usually requires that the message content to be kept secret from only the third parties other than the sender and the recipient(s). Data privacy is differentiated from confidentiality by posing an additional requirement on the details of message content, where the details of message content are kept confidential to the recipient(s) instead of only to the third parties. Traditional simple encryption methods, no matter whether they are symmetric key encryption or public key encryption, are difficult to achieve data privacy. Currently, data privacy is an important privacy related research topic.

Location privacy can be defined as the confidentiality of personal location information [28]. Location privacy is another kind of special privacy requirements due to the distinctiveness of location information, which can be obtained in many means (direct localization, calculation, or eavesdropping). Thus, traditional methods designed for data confidentiality cannot protect personal location privacy [29]. As far as the party is concerned, location privacy can be divided into two types: source (sender) location privacy or sink

(recipient) location privacy. In this thesis, we put our emphasis on the location privacy protection.

Activity privacy, as the name implies, is the privacy preservation on personal activity information, such as when and where who makes what actions [30]. Activity privacy is usually under the shelter of the former three types of privacy and has received little attention. However, the violation of activity privacy may lead to the violation of other types of privacy such as identity privacy or location privacy. Activity privacy is very important to privacy related applications or real-world business. For example, if the communication activities between two companies suddenly increase, the two companies may be negotiating an important cooperation or contract. Currently, maybe due to the above-mentioned reasons, activity privacy is also receiving more and more attention from both academia and industry.

The above four types of privacy are not isolated from each other, and, instead, they are usually correlated with each other. For example, the breach of activity privacy may disclose the identity privacy, and the violation of location privacy combined with on-site observation may also disclose the identity privacy. In this sense, privacy preservation is a very challenging issue and needs comprehensive consideration from many aspects.

2.3 Characteristics of MWNs

In this section, we will discuss and summarize the characteristics of MWNs as follows.

2.3.1 Open Wireless Channels

A major characteristic of MWNs is the open nature of wireless channels. Wireless signals are broadcast in nature, which allows and attracts a lot of malicious attacks. Compared with wired cables, wireless channels make eavesdropping much easier since an adversary does not need any special wiretapping tools or doing any wiretapping operations. A common user who has a wireless interface can easily eavesdrop to wireless signals being transmitted around.

The open wireless channels greatly increase the chance of eavesdropping attacks, which may lead to the violation of confidentiality.

In addition, the open nature of wireless channels allows and induces more active attacks such as data injection/modification. Same as the eavesdropping attacks, data injection/modification attacks also do not need extra special hardware other than a wireless interface. A malicious node can inject a fake packet into a communication flow or modify the content of an authentic packet in a flow at any time. These attacks may breach the integrity of data and communication flows.

Furthermore, open wireless channels may incur interference attacks. Different from those of the above two types of attacks, the objective of interference attacks is to jam the wireless channels and finally achieve the denial of service (DoS) [31]. Interference attacks may greatly reduce the availability of the wireless system.

In summary, open wireless channels allow and induce more passive and active attacks such as eavesdropping, data injection/modification, and jamming. These attacks make wireless channels more vulnerable to malicious abuse and destruction. In addition, the potential profit of these attacks may induce more advanced attack techniques such as flow tracing and traffic analysis, which pose severer threats to user privacy.

2.3.2 Vulnerability of Intermediate Nodes

Due to the limited effective transmission range of wireless signals, nodes in MWNs are to be deployed much denser than nodes in other networks such as wired networks. The denser the network nodes, the cheaper each node will be. Due to the practical economic reasons, each node may not be powerfully equipped to resist all kinds of attacks. In addition, wireless network nodes are usually deployed outside of a building or in a public area, instead of being deployed inside a building as wired network nodes. Adversaries can easily get into these areas and then launch malicious attacks without the awareness of the administrators or

authentic users. All these factors make nodes in MWNs more vulnerable to node compromising attacks.

2.3.3 Multi-path Forwarding

For local wired networks, a common terminal user can only access the Internet through one routing path since the minimum wired cables are supposed to be deployed to save the wiring cost. For local MWNs such as community wireless mesh networks, the wiring cost is no longer a concern because network nodes are interconnected through wireless signals. In such a MWN, packets can be forwarded through multiple routing paths to increase the robustness and reliability of the network system [32]. In addition, multi-path packet forwarding can also be utilized to balance the network traffic load and to maximize the network throughput.

2.3.4 Resource Constraints

As mentioned above, nodes in MWNs are dense and cheap. Especially in asset monitoring wireless sensor networks, each node is only a cheap sensor equipped with a wireless transceiver. In such MWNs comprised of cheap nodes, various resources such computing capability, storage space, power supply, and radio spectrum may become critical constraints. Resource constraints are another major concern in the scheme design for information security. For example, a cheap sensor may have no sufficient computing capability to perform expensive public-key encryption on each message; sensor nodes may also have no enough storage space to keep all sensed data; limited power supply from a small battery may also require sensor nodes to transmit data as efficient as possible. In addition, resource constraints may induce more DoS attacks. Thus, many information security schemes may not be suitable for resource-constraint MWNs due to their high computation or transmission overhead, and new lightweight schemes need to be explored.

In summary, these characteristics of MWNs not only pose challenges to security design but also bring opportunities to solve these information security issues from a new perspective.

For example, the multi-path packet forwarding can help to enhance the robustness and reliability of a MWN. In the following section, we will specifically explore the research challenges of information security in MWNs.

2.4 Identified Research Challenges

Nowadays, a common user can be very powerful with its ability to access the Internet. Distributed computing services such as grid computing can provide a common user with a tremendous computing capability; search engines, e.g., google and baidu, can provide a common user with the ability to access enormous data and documents. A single attacker can also utilize these services to equip itself into a powerful and intelligent adversary, and it is becoming more and more difficult to defend against these adversaries.

In this section, we present several identified research challenges of information security in MWNs. Some of them such as traffic analysis attacks and internal adversaries have already been identified in traditional wired networks, but they may pose new threats to or exhibit new behaviors in MWNs. Some of them such as traffic explosion which may lead to DoS are newly found in MWNs.

2.4.1 Thwarting Traffic Analysis Attacks

Traffic analysis attacks [33] have already been identified in traditional wired networks, but they pose much severer threats to newly emerging MWNs. Wiretapping in traditional wired networks is not as easy as eavesdropping in wireless networks, and special hardware is required for the attacks. The easiness of eavesdropping induces much more traffic analysis attacks, posing severer threats to wireless networks, especially MWNs.

Traffic analysis attacks have many forms such as flow tracing, rate monitoring, etc, and flow tracing is one of the critical attacks related to MWNs. Flow tracing has two types, forward tracing and back tracing, which are used to compromise sink privacy and source privacy, respectively. Flow tracing can be performed based on advanced techniques such as

time correlation, size correlation, or message content correlation. In time correlation, for example, the attacker may observe the time order of incoming and outgoing packets and attempt to correlate them together to deduce the forwarding paths. These advanced flow tracing techniques pose great challenges to thwarting traffic analysis attacks in MWNs.

2.4.2 Thwarting Internal Adversaries

As discussed in 2.3.2, intermediate network nodes in MWNs are vulnerable to node compromising attacks due to many practical reasons such as transmission range, node density, and product cost. If an intermediate network node is compromised, the attacker may stay inside the node and become an internal adversary. An internal adversary may have obtained the secret keys stored at the node and have acquired the full control of the node, making it very challenging to thwart internal adversaries from compromising user privacy. Link-to-link encryption can not preserve user privacy since the secret keys have already been compromised; end-to-end encryption, no matter whether it is symmetric-key or public-key encryption, can not prevent adversaries from tracing to source or sink nodes since message ciphtertext may be correlated to deduce the forwarding paths. Traditional methods can not preserve user privacy, and new schemes are required for the information security of MWNs.

2.4.3 Preventing Traffic Explosion

As shown in [26], dummy messages are usually employed to achieve event source unobservability, which is a very attractive and desirable privacy objective. However, dummy messages may lead to a severe issue, traffic explosion, which can greatly degrade the performance of a networking system. Traffic explosion is not only a performance issue but also an information security issue since the availability of a networking system may be breached by the DoS due to traffic explosion. Preventing traffic explosion and at the same time achieving source unobservability is a very challenging issue.

2.4.4 Trade-off between Security and Performance

MWNs will certainly extend the radio coverage range and at the same time attract more users. To support more users, various network resources, such as computing capability and transmission bandwidth, may become a performance bottleneck. The consideration for performance requirements [34] makes it inevitable for researcher and designer to face the tradeoff between security and performance. In some cases, the performance must be sacrificed to satisfy a security requirement; in other cases, a strict security requirement can be loosened a little to trade for a great improvement in performance. Our design principle is to maximize the performance after satisfying the predefined security requirements. Even with this design principle, it is still very challenging to make a tradeoff between security and performance in a complex real-world situation.

2.5 Preliminaries

Most of our solutions for the above research challenges are based on the newly-emerging network coding technologies [35]. Thus, it is necessary to give a brief introduction to the network coding model. Homomorphic Encryption Functions (HEFs) are also introduced in this part.

2.5.1 Network Coding

Unlike other packet-forwarding systems, network coding allows intermediate nodes to perform computation on incoming messages, making outgoing messages be the mixture of incoming ones. This elegant principle implies a plethora of surprising opportunities, such as random coding [36]. As shown in Figure 2.1, an outgoing packet is formed by taking a random combination of packets in the current incoming buffer.

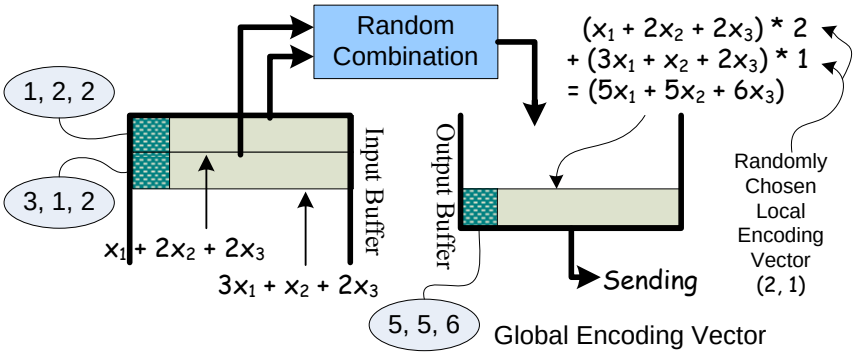

Figure 2.1: Random Coding (Mixing) at Intermediate Nodes

An overview of network coding and possible applications has been given in [37], and packet tagging and buffering are key techniques for practical network coding [38]. Packet tagging will be introduced later. In practical network coding, source information should be divided into blocks with h packets in each block. All coded packets related to the kth block belong to generation k and random coding is performed only among the packets in the same generation. Packets within a generation need to be synchronized by buffering for the purpose of network coding at intermediate nodes.

Consider an acyclic network (V, E, c) with unit capacity, i.e., $c(e)$=1 for all $e \in E$, meaning that each edge can carry one symbol per unit time, where V is the node set and E is the edge set. Assume that each symbol is an element of a finite field $\mathbb{F}_q$. Consider a network scenario with multicast sessions, where a session is comprised of one source $s \in V$ and a set of sinks $T \subseteq V$ (or one single sink $t \in V$). Let $h = MinCut(s,T)$ be the multicast capacity, and $x_1, \cdots, x_h$ be the h symbols to be delivered from s to T.

For each outgoing edge e of a node v, let $y(e) \in \mathbb{F}_q$ denote the symbol carried on e, which can be computed as a linear combination of the symbols $y(e')$ on the incoming edges e' of node v, i.e., $y(e) = \sum_{e'} \beta_{e'}(e) y(e')$. The coefficient vector $\boldsymbol{\beta}(e) = [\beta_{e'}(e)]$ is called *Local Encoding Vector* (LEV).

By induction, the symbol $y(e)$ on any edge $e \in E$ can be computed as a linear combination of the source symbols $x_1, \cdots, x_h$, i.e., $y(e) = \sum_{i=1}^{h} g_i(e) x_i$. The coefficients form

a *Global Encoding Vector* (GEV) $\boldsymbol{g}(e)=[g_1(e),...,g_h(e)]$, which can be computed recursively as $\boldsymbol{g}(e)=\sum_{e'}\beta_{e'}(e)\boldsymbol{g}(e')$, using the LEVs $\boldsymbol{\beta}(e)$. The GEV $\boldsymbol{g}(e)$ represents the code symbol $y(e)$ in terms of the source symbols $x_1,\cdots,x_h$.

Suppose that a sink $t\in T$ receives symbols $y(e_1),\cdots,y(e_h)$, which can be expressed in terms of the source symbols as

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}, \tag{2.1}$$

where G_t is called *Global Encoding Matrix* (GEM) and the ith row of G_t is the GEV associated with $y(e_i)$. Further, sink t can recover the h source symbols by inverting the matrix G_t and applying the inverse to $y(e_1),\cdots,y(e_h)$.

In general, each packet can be considered as a vector of symbols $\boldsymbol{y}(e)=[y_1(e),\cdots,y_N(e)]$. By likewise grouping the source symbols into packets $\boldsymbol{x}_i=[x_{i,1},\cdots,x_{i,N}]$, the above algebraic relationships carry over to packets. To facilitate the decoding at the sinks, each message should be tagged with its GEV $\boldsymbol{g}(e)$, which can be easily achieved by prefixing the ith source packet $\boldsymbol{x}_i$ with the ith unit vector $\boldsymbol{u}_i$. Then, each packet is automatically tagged with the corresponding GEV, since

$$\begin{aligned} [\boldsymbol{g}(e),\boldsymbol{y}(e)] &= \sum_{e'}\beta_{e'}(e)[\boldsymbol{g}(e'),\boldsymbol{y}(e')] \\ &= \sum_{i=1}^{h} g_i(e)[\boldsymbol{u}_i,\boldsymbol{x}_i] \end{aligned}. \tag{2.2}$$

The benefit of tags is that the GEVs can be found within the packets themselves, so that the sinks can compute G_t without knowing the network topology or packet-forwarding paths. Nor is a side channel required for the communication of G_t. Actually, the network can be dynamic, with nodes and edges being added or removed in an ad hoc way. The coding arguments can be time varying and random.

2.5.2 Homomorphic Encryption Functions

Homomorphic Encryption Functions (HEFs) have the property of homomorphism, which means operations on plaintext can be performed by operating on corresponding ciphertext. If $E(\cdot)$ is a HEF, $E(x+y)$ can be computed from $E(x)$ and $E(y)$ without knowing the corresponding plaintext *x* and *y*. To be applicable in the proposed scheme, a HEF $E(\cdot)$ needs to satisfy the following properties:

1) **Additivity**: Given the ciphertext $E(x)$ and $E(y)$, there exists a computationally efficient algorithm $Add(\cdot,\cdot)$ such that $E(x+y)=Add(E(x),E(y))$.

2) **Scalar Multiplicativity**: Given $E(x)$ and a scalar *t*, there exists a computationally efficient algorithm $Mul(\cdot,\cdot)$ such that $E(x\cdot t)=Mul(E(x),t)$.

Actually, the scalar multiplicativity can be deduced from the additivity, since $E(x\cdot t)=E(\sum_{i=1}^{t}x)$. Benaloh [39] and Paillier [40] cryptosystems are of such an additive HEF, where the addition on plaintext can be achieved by performing a multiplicative operation on the corresponding ciphertext, i.e., $E(x_1+x_2)=E(x_1)\cdot E(x_2)$. Further, the following two equations can be easily derived:

$$\begin{aligned} E(x\cdot t)&=E^{t}(x)\\ E\left(\sum_i x_i\cdot t_i\right)&=\prod_i E^{t_i}\left(x_i\right)\end{aligned}. \tag{2.3}$$

2.6 Summary

In this chapter, we have discussed the threats, requirements, and characteristics related to the information security of MWNs. We also present four research challenges, followed by the network coding model. In the following chapters, we will present our solutions to face those challenges.

Chapter 3
Network Coding Based Privacy Preservation against Traffic Analysis in Multi-hop Wireless Networks

In this chapter, we propose a novel network coding based privacy-preserving scheme against traffic analysis in MWNs. As we have discussed before, wireless access networks, such as Wi-Fi, have been widely deployed due to their convenience, portability, and low cost. However, they still suffer from inherent shortcomings such as limited radio coverage, poor system reliability, as well as lack of security and privacy preservation. Multi-hop Wireless Networks (MWNs) are regarded as a highly promising solution for extending the radio coverage range of the existing wireless networks, and they can also be used to improve the system reliability through multi-path packet forwarding. However, due to the open wireless medium, MWNs are susceptible to various attacks, such as eavesdropping, data modification/injection, and node compromising. These attacks may breach the security of MWNs, including confidentiality, integrity, and authenticity. In addition, some advanced attacks, such as traffic analysis and flow tracing, can also be launched by a malicious adversary to compromise users' privacy, including source anonymity and traffic secrecy. In this chapter, we focus on the privacy issue, i.e., how to prevent traffic analysis/flow tracing and achieve source anonymity in MWNs.

Among all privacy properties, source anonymity is of special interest in MWNs. Source anonymity refers to communicating through a network without revealing the identity or location of source nodes. Preventing traffic analysis/flow tracing and provisioning source anonymity are critical for privacy-aware MWNs, such as wireless sensor or tactical networks. Consider a simple example of multicast communication in military ad hoc networks (which can be regarded as a kind of MWNs), where nodes can communicate with each other through multi-hop packet forwarding. If an attacker can intercept packets and trace back to the source through traffic analysis, it may disclose some sensitive information such as the location of

critical nodes (e.g., the commanders) and then further it may impair the location privacy. Subsequently, the attacker can take a series of actions to launch the so-called *Decapitation Strike* to destroy these critical nodes [11], as shown in Figure 3.1 (A). Another example is the event reporting in wireless sensor networks, where flow tracing can help attackers to identify the location of concerned events, e.g., the appearance of an endangered animal in a monitored area, and then take subsequent actions to capture or kill the animals [41], as shown in Figure 3.1 (B).

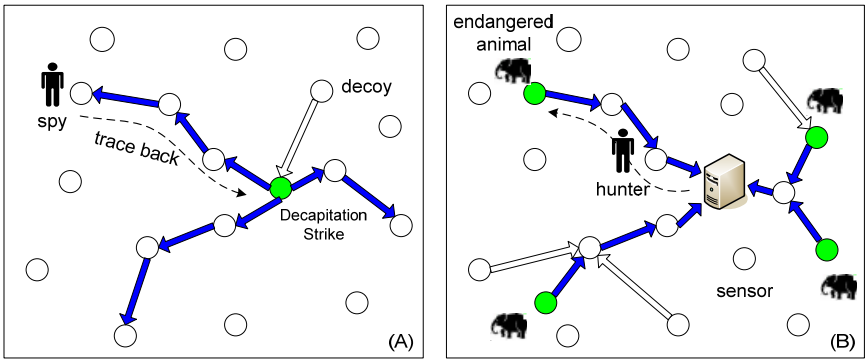

Figure 3.1: Privacy Threats in MWNs

How to efficiently thwart traffic analysis/flow tracing attacks and provide privacy protection in MWNs is a very challenging issue. Existing privacy-preserving solutions, such as proxy-based schemes [42], [43], Chaum's mix-based schemes [44], [45], and onion-based schemes [46], [47], may either require a series of trusted forwarding proxies or result in severe performance degradation in practice. Different from previous research, our study examines the privacy issue from a brand-new perspective: using network coding to achieve privacy preservation.

Network coding was first introduced by Ahlswede et al [48]. Subsequently, two key techniques, random coding [36] and linear coding [49], further promoted the development of network coding technologies. The random coding makes network coding more practical, while the linear coding is proven to be sufficient and computationally efficient for network coding. Currently, network coding has been widely recognized as a promising information

dissemination approach to improve network performance. Primary applications of network coding include file distribution and multimedia streaming on P2P overlay networks [50], data transmission in sensor networks [51], tactical communications in military networks [52], etc. Compared with conventional packet forwarding technologies, network coding offers, by allowing and encouraging coding/mixing operations at intermediate forwarders [48], several significant advantages such as potential throughput improvement [53], transmission energy minimization [54], and delay reduction [37]. In addition, network coding can also work as an erasure coding scheme to enhance the dependability of a distributed data storage system [55].

The deployment of network coding in MWNs can not only bring the above performance benefits, but also provide a feasible way to efficiently thwart the traffic analysis/flow tracing attacks since the coding/mixing operation is encouraged at intermediate nodes. Similar to Chaum's mix-based schemes [44], [45], network coding provides an intrinsic message mixing mechanism, which implies that privacy preservation may be efficiently achieved in a distributed manner. Moreover, the unlinkability between incoming packets and outgoing packets, which is an important privacy property for preventing traffic analysis/flow tracing, can be achieved by mixing the incoming packets at intermediate nodes. However, the privacy offered by such a mixing feature is still vulnerable, since the linear dependence between outgoing and incoming packets can be easily analyzed. A simple deployment of network coding cannot prevent traffic analysis/flow tracing since the explicit Global Encoding Vectors (GEVs, also known as tags) prefixed to the encoded messages provide a back door for adversaries to compromise the privacy of users. Once enough packets are collected, adversaries can easily recover the original packets and then conduct the traffic analysis/flow tracing attacks based on these original packets. A naïve solution to address this vulnerability is to employ link-to-link encryption. This method can prevent traffic analysis to a certain degree, but it introduces very heavy computational overhead and thus results in significant performance degradation of the whole network system. Additionally, it cannot preserve the

privacy of users once some intermediate nodes are compromised by adversaries. Such deficiencies motivate us to explore an efficient privacy-preserving scheme for MWNs.

In this chapter, based on network coding and Homomorphic Encryption Functions (HEFs) [39], [40], we propose an efficient privacy-preserving scheme for MWNs. Our objective is to achieve source anonymity by preventing traffic analysis and flow tracing. To the best of our knowledge, this is the first research effort in utilizing network coding to thwart traffic analysis/flow tracing and realize privacy preservation. The proposed scheme offers the following attractive features: 1) **Enhanced Privacy against traffic analysis and flow tracing**. With the employment of HEFs, the confidentiality of GEVs is effectively guaranteed, making it difficult for attackers to recover the plaintext of GEVs. Even if some intermediate nodes are compromised, the adversaries still cannot decrypt the GEVs, since only the sinks know the decryption key. Further, the confidentiality of GEVs brings an implicative benefit, i.e., the confidentiality of message content [57], because message decoding only relies on GEVs. On the other hand, with random recoding on encrypted GEVs, the coding/mixing feature of network coding can be exploited in a natural manner to satisfy the mixing requirements of privacy preservation against traffic analysis; 2) **Efficiency**. Due to the Homomorphism of HEFs, message recoding at intermediate nodes can be directly performed on encrypted GEVs and encoded messages, without knowing the decryption keys or performing expensive decryption operations on each incoming packet. The performance evaluation on computational complexity demonstrates the efficiency of the proposed scheme; and 3) **High Invertible Probability**. Random network coding is feasible only if the prefixed GEVs are invertible with a high probability. Theoretical analysis demonstrates that the influence of HEFs on the invertible probability of GEVs is negligible. Thus, the random coding feature can be kept in our network coding based privacy-preserving scheme.

The remainder of the chapter is organized as follows. Section 3.1 gives the threat models. In Section 3.2, the proposed privacy-preserving scheme is presented in detail. In Sections 3.3 and 3.4, security analysis and performance evaluation/optimization are

conducted, respectively. In Section 3.5, related work is surveyed, followed by a summary in Section 3.6.

3.1 Threat Models

In this chapter, we consider the following two attack models.

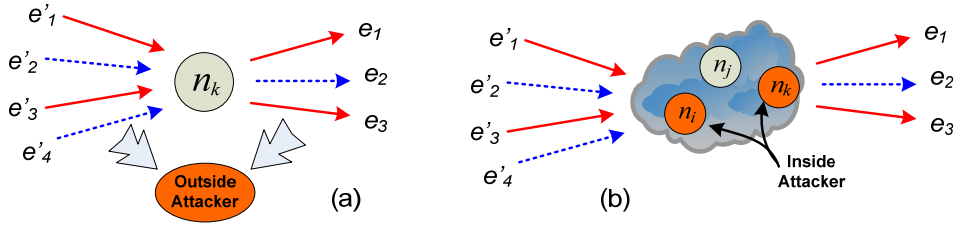

Figure 3.2: Attack Model: (a) Outside Attacker, (b) Inside Attacker

Outside Attacker: An outside attacker can be considered as a global passive eavesdropper who has the ability to observe all network links, as shown in Figure 3.2 (a). An outside attacker can examine the tags and message content, and thus link outgoing packets with incoming packets. Further, even if end-to-end encryption is applied to messages at a higher layer, it is still possible for a global outside attacker to trace packets by analyzing and comparing the message ciphertext.

Inside attacker: An inside attacker may compromise several intermediate nodes, as shown in Figure 3.2 (b). Link-to-link encryption is vulnerable to inside attackers since they may already have obtained the decryption keys and thus the message plaintext can be easily recovered.

Both inside and outside attackers may perform more advanced traffic analysis/flow tracing techniques, including size correlation, time correlation, and message content correlation [58]. Adversaries can further explore these techniques to deduce the forwarding paths [41] and thus to compromise user privacy.

Without loss of generality, we assume that an anonymous secure routing protocol [11] is deployed to assist network nodes to determine forwarding paths. The generation number of a

packet can be hidden in the secure routing scheme so that an attacker cannot find the generation of a packet for its further analysis.

3.2 Network Coding Based Privacy-Preserving Scheme for MWNs

In this section, we propose a novel network coding based privacy-preserving scheme for MWNs, which can efficiently thwart traffic analysis and flow tracing attacks, followed by theoretical analysis on the invertibility of GEMs.

3.2.1 The Proposed Privacy-Preserving Scheme

Though providing an intrinsic mixing mechanism, the original network coding cannot provide privacy guarantee due to explicit GEVs, since an adversary can recover the original messages as long as enough packets are collected. Link-to-link encryption is vulnerable to inside attackers since they may already have compromised several intermediate nodes and obtained the secret keys.

An intuitive way to resolve this issue is to keep GEVs confidential to intermediate nodes by encrypting the GEVs in an end-to-end manner, which can prevent compromised intermediate nodes from analyzing GEVs or recovering the original messages. Such an intuitive approach, however, cannot prevent the adversaries from tracking the message ciphertext since the "mixing" feature of network coding may be disabled by the end-to-end encryption.

To address this issue, we employ the Paillier cryptosystem [40] as the HEF to apply encryption to GEVs, since protecting GEVs is generally sufficient to ensure confidentiality network coded message content [57]. HEF can not only keep the confidentiality of GEVs, but also enable intermediate nodes to efficiently mix the coded messages. In the Paillier cryptosystem, given a message m and the public key (n, g), the encryption function is described as follows,

$$E(m) = g^m \cdot r^n (\mathrm{mod}\, n^2), \tag{3.1}$$

where r is a random factor in the Paillier cryptosystem. $E(m)$ satisfies the following homomorphic property:

$$E(m_1)\cdot E(m_2) = g^{m_1+m_2}\cdot(r_1 r_2)^n \,(\mathrm{mod}\, n^2) = E(m_1+m_2)\,. \tag{3.2}$$

With HEFs, intermediate nodes are allowed to directly perform linear coding/mixing operations on the coded messages and encrypted tags, as shown in Figure 3.3. In other words, due to the homomorphism of the HEF, one can achieve linear network coding by operating on encoded messages and encrypted GEVs, without knowing the decryption keys or performing the decryption operations.

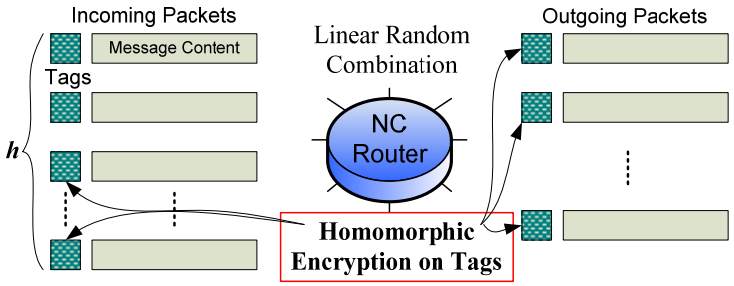

Figure 3.3: Homomorphic Encryption on Packet Tags

The proposed scheme primarily consists of three phases: source encoding, intermediate recoding, and sink decoding. Without loss of generality, we assume that each sink acquires two keys, the encryption key *ek* and the decryption key *dk*, from an offline Trust Authority (TA), and the encryption key *ek* is published to all other nodes. For supporting multicast, a group of sinks are required to obtain from the TA or negotiate the key pair in advance [59]; then, they can publish the encryption key and keep the decryption key private in the group.

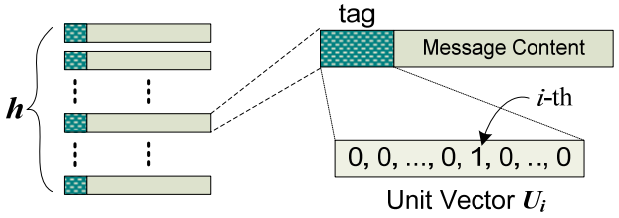

Figure 3.4: Packet Tagging before Source Encoding

Source Encoding: Consider that a source has h messages, say $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_h$, to be sent out. The source first prefixes h unit vectors to the h messages, respectively, as illustrated in Figure 3.4. After tagging, the source can choose a random LEV and perform linear encoding on these messages. Then, a LEV can produce an encoded message with the GEV (which is equal to the LEV temporarily) tagged.

To offer confidentiality for the tags, homomorphic encryption operations are applied as follows:

$$\begin{aligned} &c_i(e) = E_{ek}(g_i(e)), (1 \le i \le h) \\ &\boldsymbol{c}(e) = [c_1(e), c_2(e), \cdots, c_h(e)] \end{aligned}, \tag{3.3}$$

where the notation *ek* denotes the encryption key. Notice that we apply HEF to GEVs after (instead of before) linear encoding. We will discuss this strategy in Section 3.3 from the perspective of both the security and performance.

Intermediate Recoding: After receiving a number of packets of the same generation, an intermediate node can perform random linear coding on these packets. To generate an outgoing packet, firstly, a random LEV $[\beta_1, \cdots, \beta_h]$ is chosen independently; then, a linear combination of message content of the incoming packets is computed as the message content of the outgoing packet, as shown in Figure 2.1.

Since the tags of the h incoming packets are in ciphertext format, and an intermediate node has no knowledge of the corresponding decryption keys, it is difficult for the intermediate node to perform functions such as earliest decoding to get the original message content. However, due to the homomorphism of the encryption function, a linear transformation can be directly performed on the encrypted tags of the incoming packets to generate a new tag for the outgoing packet, namely,

$$g(e) = \sum_{i=1}^{h} \beta_i(e) g(e'_i). \tag{3.4}$$

The GEV of a new outgoing packet can be calculated according to equation (7). By utilizing the homomorphic characteristic of the encryption on GEVs, the ciphertext of the new GEVs for outgoing packets can be calculated as follows:

$$
\begin{aligned}
E_{ek}(g(e)) &= E_{ek}(\textstyle\sum_{i=1}^{h} \beta_i(e) g(e'_i)) \\
&= \textstyle\prod_{i=1}^{h} E_{ek}(\beta_i(e) g(e'_i)) \\
&= \textstyle\prod_{i=1}^{h} E_{ek}^{\beta_i(e)}(g(e'_i))
\end{aligned}
\tag{3.5}
$$

The ciphertext of new GEVs can be computed from the ciphertext of GEVs of incoming packets without the knowledge of the decryption key. Finally, the ciphertext of a new GEV is prefixed to the corresponding message content to form a new outgoing packet, which is sent out to downstream nodes.

Sink Decoding: After receiving a packet, the sink first decrypts the packet tag using the corresponding decryption key *dk*.

$$
\begin{aligned}
&g_i(e) = D_{dk}(c_i(e)) \quad (1 \le i \le h) \\
&\boldsymbol{g}(e) = [g_1(e), g_2(e), \cdots, g_h(e)]
\end{aligned}
\tag{3.6}
$$

Once enough packets are received, a sink can decode the packets to get the original messages. Then, the sink derives the decoding vector, which is the inverse of the GEM, as shown in the following equations.

$$
\begin{aligned}
&\boldsymbol{G}^{-1} \cdot \boldsymbol{G} = \boldsymbol{U} \pmod{n} \\
&\boldsymbol{G} = [\boldsymbol{g}(e_1), \boldsymbol{g}(e_2), \cdots, \boldsymbol{g}(e_h)]^T
\end{aligned}
\tag{3.7}
$$

Finally, the sink can use the inverse to recover the original messages, shown as follows.

$$
\begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_h \end{bmatrix} = \boldsymbol{G}^{-1} \begin{bmatrix} \boldsymbol{y}(e_1) \\ \vdots \\ \boldsymbol{y}(e_h) \end{bmatrix}. \tag{3.8}
$$

For random network coding, a key issue is the invertibility of a GEM. We discuss in detail the invertibility of a GEM as follows.

3.2.2 Invertibility of a GEM

Let GEM A be comprised of h GEVs with h elements in each GEV. $|A|$ and A^* are the determinant and the adjoint of the matrix A, respectively. According to the theory of linear algebra, finding the inverse of a square matrix A is equivalent to solving the corresponding

system of linear equation with A being the coefficient matrix. Gaussian elimination can be applied to solve a system of linear congruence equations. Due to the homomorphism of the modulo congruence in terms of the addition, subtraction, and multiplication operations, a system of linear congruence equations can be separated into several single equations with one unknown in each equation as follows:

$$|A|x_i = \sum_{j=1}^{h} (-1)^{i+j} y_j M_{ij} (\mathrm{mod}\, n). \tag{3.9}$$

A system of linear congruence equations is solvable if and only if every independent equation is solvable. The difference between solving a system of linear equations and solving a system of linear congruence equations lies in finding the inverse of $|A|$ modulo n. In order to further discuss the solutions, we formulate the linear congruence equations as follows:

$$|A|x_i = \left|\tilde{A}_i\right| (\mathrm{mod}\, n) \quad (i = 1, \cdots, h), \tag{3.10}$$

where $\left|\tilde{A}_i\right| = \sum_{j=1}^{h} (-1)^{i+j} y_j M_{ij} (\mathrm{mod}\, n)$.

Theorem 1: A system of linear congruence equations has a unique solution only if $|A| \neq 0$.

Proof: From the theory of modulo congruence, for $n > 1$, the mapping $f : \mathbb{Z} \mapsto \mathbb{Z}_n$ is a homomorphic mapping in terms of the addition, subtraction, and multiplication operations. Therefore, for each solution to $|A|x_i = \left|\tilde{A}_i\right| (\mathrm{mod}\, n)$, there must be one or more solutions to $|A|x_i = \left|\tilde{A}_i\right| + k \cdot n (k \in \mathbb{Z})$.

According to the theory of linear algebra, the necessary and sufficient condition for $|A|x_i = \left|\tilde{A}_i\right| + k \cdot n (k \in \mathbb{Z})$ to have a unique solution is $|A| \neq 0$. Thus, the necessary condition for a system of linear congruence equations to have a unique solution is $|A| \neq 0$. ■

However, this condition is not sufficient for a system of linear congruence equations to have a unique solution, because a solution for $|A|x_i = \left|\tilde{A}_i\right|$ does not imply a corresponding solution for $|A|x_i = \left|\tilde{A}_i\right| (\mathrm{mod}\, n)$.

Theorem 2: A system of linear congruence equations has d^h solutions if:

$$\begin{cases} |A| \neq 0 \\ \left|\tilde{A}_i\right| \equiv 0 (\bmod d) \quad (i = 1, 2, \cdots, h) \end{cases} \tag{3.11}$$

where $d = \gcd(|A|, n)$.

Proof: According to **Theorem 1**, $|A| \neq 0$ is a necessary condition for a system of linear congruence equations to have a unique solution. Therefore, the original system of linear congruence equations can be transformed to $|A| x_i = \left|\tilde{A}_i\right| (\bmod n) (i = 1, 2, \cdots, h)$ by only applying addition and multiplication operations, which are preserved by the homomorphic mapping $f : \mathbb{Z} \mapsto \mathbb{Z}_n$.

In equations $|A| x_i = \left|\tilde{A}_i\right| (\bmod n)$, variables $x_i (i = 1, 2, \cdots, h)$ can be solved independently by employing the theory of linear congruence equations. A linear congruence equation $ax = b (\bmod n)$ is solvable if and only if the congruence $b = 0 (\bmod d)$ with $d = \gcd(a, n)$ is solvable, where $\gcd(a, n)$ is the greatest common divisor of a and n. Let one solution to the linear congruence equation be $x_0 < n / d$. Then, the solutions are $x = x_0, x_0 + n / d, \cdots, x_0 + (d - 1) n / d$. If $d = 1$, there is only one unique solution which is less than n. According to the theory of linear congruence equations, if $\left|\tilde{A}_i\right| \equiv 0 (\bmod d)$ $(i = 1, 2, \cdots, h)$, every equation $|A| x_i = \left|\tilde{A}_i\right| (\bmod n)$ has d solutions. The solutions to a system of linear congruence equations are the combinations of these independent solutions. The combination number is d^h. ■

Corollary 1: A system of linear congruence equations has a unique solution if and only if:

$$\begin{cases} |A| \neq 0 \\ \gcd(|A|, n) = 1 \end{cases} \tag{3.12}$$

and $x_i = |A|^{-1} \left|\tilde{A}_i\right| (\bmod n) \quad (i = 1, 2, \cdots, h)$.

Proof: From **Theorem 2**, if $d = \gcd(|A|, n) = 1$, a system of linear congruence equations has $d^h = 1$ solution, which is the unique solution to the system. In addition, according to the congruence theory, when $\gcd(|A|, n) = 1$, the modular inverse of the integer $|A|$, which is

denoted as $|A|^{-1}$, can be calculated using the extended Euclidean algorithm. The result satisfies the following equation: $|A|^{-1}|A| = 1 (\mathrm{mod}\, n)$. The unique solution to a system of linear congruence equations is $x_i = |A|^{-1}|\tilde{A}_i| (\mathrm{mod}\, n)$ $(i = 1, 2, \cdots, h)$. The solution can also be expressed in a matrix form as $|A|^{-1}|A|X = X = |A|^{-1} A^* Y (\mathrm{mod}\, n)$. ■

Theorem 2 and **Corollary 1** indicate that a solvable system of linear congruence equations does not imply the invertibility of the corresponding coefficient matrix. A stronger condition, i.e., $\gcd(|A|, n) = 1$, is required for the invertibility of a coefficient matrix A modulo n. The above theorems and corollary generally hold whether n takes the value of a prime number q or the product of two prime numbers p and q. In section 3.4, we will further give a quantitative analysis on the invertible probability of a coefficient matrix.

3.3 Security Analysis

The proposed scheme can provide privacy preservation by means of resisting traffic analysis/flow tracing attacks such as size correlation, time correlation, and message content correlation. **Size correlation** can be naturally prevented since each message is trimmed to be of the same length in network coding based schemes. **Time correlation** can be effectively resisted by the inherent buffering technique [37] of network coding. Let the time length of buffering periods be T_b and the average arrival rate of coded packets be λ. The time correlation attack can succeed only when exactly one packet arrives in the buffering period T_b, since zero packets make the attack meaningless and more than one packet can induce the "mixing" operation, making time correlation useless. If coded packets arrive following the Poisson distribution, the probability of a successful time correlation attack can be given as follows:

$$\Pr(1, \lambda \cdot T_b) = \lambda \cdot T_b \cdot e^{-\lambda \cdot T_b}. \tag{3.13}$$

From Eq. (3.13), it can be seen that the probability decreases exponentially with the time period T_b. On the other hand, the transmission delay increases linearly with the time period

T_b. In practice, we can adaptively adjust parameter T_b according to the security and delay requirements.

Message content correlation can be resisted by the "mixing" feature of network coding. With the assistance of HEF, GEVs are kept confidential to eavesdroppers, making it difficult for adversaries to perform linear analysis on GEVs. In addition, HEF keeps the random coding feature, making the linear analysis on message content almost computationally impossible. Let the number of intercepted packets be w. The computational complexity for attackers to examine if a packet is a linear combination of h messages is $\boldsymbol{O}(h^3 + h \cdot l)$ in terms of multiplication. Thus, the computational complexity to analyze the intercepted w packets is $O(C_w^h(h^3 + h \cdot l))$, which increases exponentially with w, as shown in Figure 3.5. Compared to the conventional network coding with explicit GEVs, the proposed scheme significantly enhances privacy so that the traffic analysis attacks are almost computationally impossible.

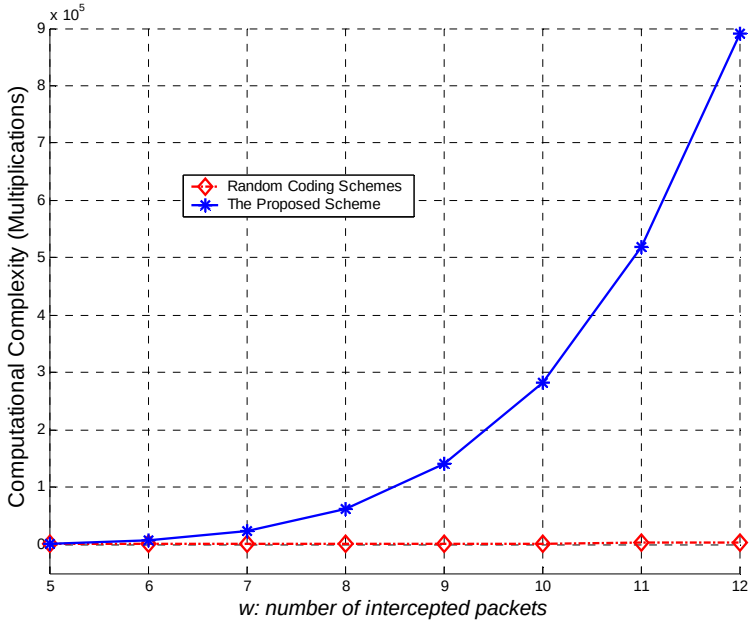

Figure 3.5: Privacy Enhancement in terms of Computational Complexity (h=5, l=200)

In the source encoding phase, we apply HEFs to GEVs after (instead of before) linear encoding. From security perspective, this choice is more secure since independent random factors can be chosen for each encryption operation; these random factors can bring more randomness to the ciphertext of GEVs and make content correlation more difficult. From performance perspective, it is argued that source encoding may be more lightweight if HEFs are applied before linear coding and independent random factors are only chosen for different GEV elements. This argument is not proper since, for each new GEV element, linear coding after encryption requires averagely about *h* exponentiations and *h-1* multiplications, which are computationally much more expensive than those of linear coding before encryption (which requires 2 exponentiations and 1 multiplication).

3.4 Performance Evaluation and Optimization

In this section, we evaluate the performance of the proposed scheme in terms of invertible probability and computational overhead. A performance optimization framework is also developed to minimize the statistical computational overhead.

3.4.1 Invertible Probability

Let each element of a LEV be randomly chosen from a field $\mathbb{F}_q$. The following two theorems hold.

Theorem 3: For a Local Encoding Matrix (LEM), which is comprised of *h* LEVs with *h* elements in each LEV and each element is from the finite field $\mathbb{F}_q$, the invertible probability of a GEM (also with *h* vectors) is degraded by $s_q = \prod_{i=1}^{h}(1-q^{-i})$.

Proof: The invertibility factor of *h* LEVs depends only on the linear dependence of the *h* LEVs themselves. Firstly, the elements in the first LEV can be any combinations except for all zeros. Therefore, the invertibility factor of the first LEV is $1-q^{-h}$. The second LEV should be linearly independent on the first one, where the all-zero vector is also excluded; thus, the invertibility factor of the second LEV is $1-q/q^{h}=1-q^{1-h}$. The third LEV should

be linearly independent on the former two; thus, the invertibility factor is $1-q^2/q^h=1-q^{2-h}$. Similarly, for the remaining LEVs, the invertibility factors are $1-q^{3-h}$, $1-q^{4-h}$, $\cdots$, $1-q^{-1}$, respectively. Therefore, the overall invertibility factor of the whole LEM is the product of these individual factors: $s_q=\prod_{i=1}^{h}(1-q^{-i})$. ■

Corollary 2: The invertibility factor s_q of an $h\times h$ LEM can be approximated to $1-q^{-1}-q^{-2}$ when $h\geq 4$, and the error of this approximation is within the magnitude of $\boldsymbol{O}(q^{-5})$.

This corollary can be easily proven by expanding the multiplication of the polynomials. This corollary gives two important implications. Firstly, in practical network coding, the min-cut capacity h is much larger than the condition in ***corollary 2*** and, thus, this corollary can be safely used. Secondly, the field size q is relatively a large number. Therefore, an amount in the magnitude $\boldsymbol{O}(q^{-5})$ is very small and can be omitted. Based on ***Theorem 3*** and ***Corollary 2***, the invertible probability of a GEM can be easily calculated. For a network coding system with a min-cut capacity $h\,(h\geq 4)$, the invertible probability can be approximated as $(1-q^{-1}-q^{-2})^t$, where q is the field size and t is the total coding time from the source to sinks. In practical network coding, since q is a relatively large prime number, the above invertible probability can be further approximated to $1-tq^{-1}$.

Theorem 3 does not apply to the Paillier cryptosystem, since elements in the cryptosystem are chosen from a ring $\mathbb{R}_n$ (another algebraic structure), where $n=pq$ and p, q are two prime numbers.

Theorem 4: For a LEM (comprised of h LEVs), where the elements are randomly chosen from a ring $\mathbb{R}_n$ ($n=pq$), the invertible probability of a GEM (also with h vectors) is degraded by $s_p+s_q-s_n$.

Proof: The problem can be decomposed into two separate sub-problems in terms of the prime numbers p and q, respectively. As for the sub-problem in terms of p, there is a mapping from the original problem to the problem modulo p. According to ***Theorem 3***, the

invertibility factor is $s_p = \prod_{i=1}^{h}(1-p^{-i})$. Similarly, the invertibility factor of the sub-problem in terms of q is $s_q = \prod_{i=1}^{h}(1-q^{-i})$. The above two sub-problems have overlap, which occurs at these points where the number is congruent to zero modulo n. The invertibility factor related to the overlap area is $s_n = \prod_{i=1}^{h}(1-n^{-i})$. According to the union principle of the set theory, the overall invertibility factor is: $1-((1-s_p)+(1-s_q)-(1-s_n)) = s_p + s_q - s_n$. ■

If $|p| \approx |q|$, the integral invertibility factor can be approximately reduced to $1-p^{-1}-q^{-1}$, with the error confined in $\boldsymbol{O}(p^{-2}+q^{-2})$. If a session performs totally t times of random coding, the invertible probability of GEVs at sinks is $(1-p^{-1}-q^{-1})^t$. Since p and q are relatively large prime numbers in practical network coding, the invertible probability can be approximately reduced to $1-t(p^{-1}+q^{-1})$. It can be seen that the invertible probability is dependent on the random coding times, instead of the number of sinks.

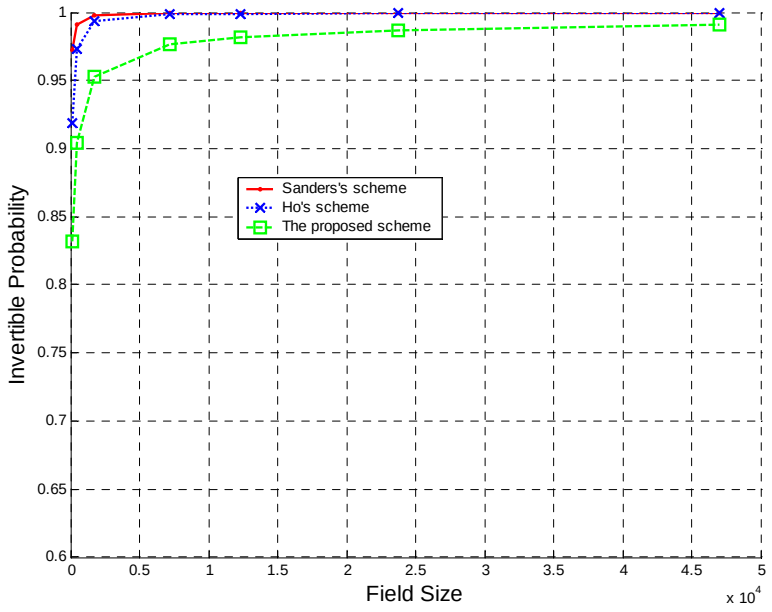

Figure 3.6: Invertible Probability vs. Field Size

We compare the invertible probability of the proposed scheme with those of the random coding schemes in [36] and [49], as shown in Figure 3.6. It can be seen that the invertible probability of the proposed scheme is similar to those of the random coding schemes, however, the proposed scheme can offer significant privacy enhancement, which is very critical in practical applications. In addition, Figure 3.7 shows the invertible probability decreases with the increase of the random coding time, and the simulation results match the analytical results very well, thus demonstrating the validity of the performance analysis.

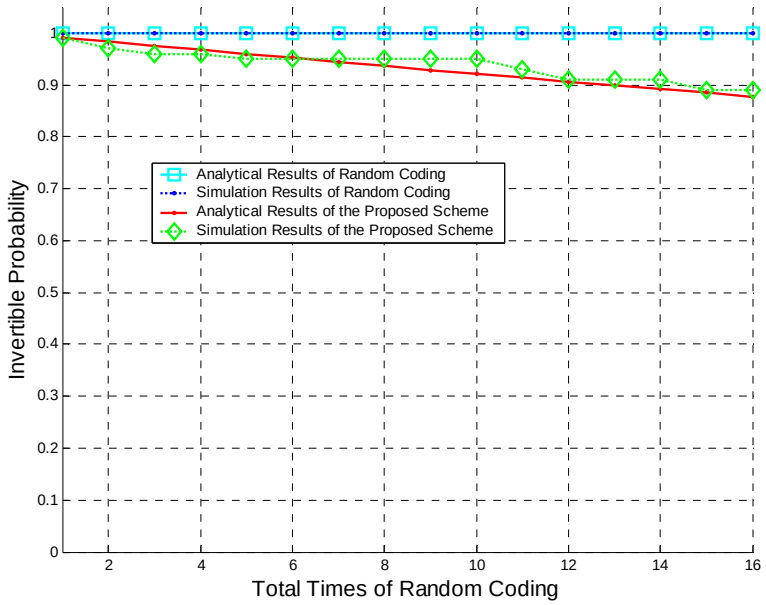

Figure 3.7: Invertible Probability vs. Total Times of Random Coding (p=241, q=251)

3.4.2 Computational Overhead

The computational overhead of the proposed scheme can be investigated respectively from three aspects: source encoding, intermediate recoding, and sink decoding. Since the computational overhead of the proposed scheme is closely related to the specific

homomorphic encryption algorithm, in the following analysis, we will take the Paillier cryptosystem as the encryption method when necessary.

Source Encoding Overhead: Consider h GEVs with h elements in each GEV, which form an $h \times h$ GEM. After source encoding, every element in the GEM is encrypted one by one. Thus, the computational overhead is $\boldsymbol{O}(h^2)$ in terms of encryption operations. Every encryption operation requires 2 exponentiations, 1 multiplication, and 1 modulus operation in the Paillier cryptosystem. Therefore, the computational complexity is $\boldsymbol{O}(h^2 \cdot \log n)$ in terms of multiplication operations.

Intermediate Recoding Overhead: In intermediate nodes, linear transformation on the elements of GEVs can be performed only by manipulating the ciphertext of these elements because intermediate nodes have no knowledge of decryption keys. According to Eq. (3.8), the computational complexity of producing one element in new GEVs is h exponentiations and $h-1$ multiplications on the ciphertext, which is $\boldsymbol{O}(h \cdot \log n)$ in terms of multiplications together. Thus, the computational complexity is $\boldsymbol{O}(h^2 \cdot \log n)$ for a GEV and $\boldsymbol{O}(h^3 \cdot \log n)$ for a GEM with h GEVs in terms of multiplication.

Sink Decoding Overhead: After receiving an encoded message, a sink can decrypt the elements in the GEV. According to the Paillier cryptosystem, decrypting an element requires 1 exponentiation, 1 multiplication, and 1 division operation. Therefore, the computational complexity of decrypting a GEV is $\boldsymbol{O}(h \cdot \log n)$ in terms of multiplication operations. Thus, for a whole GEM with h GEVs, the computational overhead is $\boldsymbol{O}(h^2 \cdot \log n)$ in terms of multiplication.

3.4.3 Performance Optimization

As described in the previous subsections, the invertible probability and computational overhead of the proposed scheme are $1-t(p^{-1}+q^{-1})$ and $\boldsymbol{O}(h^3 \cdot \log n)$, respectively. Thus, the statistical computational overhead for a GEM can be expressed in terms of multiplications as follows:

$$CO = \frac{h^3 \cdot \log n}{1 - t(p^{-1} + q^{-1})}. \tag{3.14}$$

From Eq. (3.14), we can see that the computational overhead of the proposed scheme is a monotonically increasing function of *h*, i.e., the length of a GEV, for any given *n* and *t*. As discussed in Section 3.3, the security of the proposed scheme is also monotonically increasing with the increase of *h*. Thus, a tradeoff between the security and the computational overhead should be considered in practical deployment. A typical way to deal with this tradeoff is to set the security requirements first and then choose the minimum *h* to meet the requirements. In this way, the minimum computational overhead can be achieved.

On the other hand, noticing that $n = pq$ and $|p| \approx |q|$, we can approximate Eq. (3.14) for any given *h* and formulate it as the following optimization problem:

$$\begin{aligned} &\textit{Minimize} \quad g(n,t) = \frac{\log n}{1 - 2t/\sqrt{n}} \\ &\textit{subject to}: \quad n \geq 4, \textit{n is an integer}, \\ &\textit{where} \qquad t \geq 1, \textit{t is an integer}. \end{aligned} \tag{3.15}$$

By solving the ordinary differential equation $\frac{\partial g}{\partial n} = 0$, we have: $n_1 = 4t^2 LambertW(-1/2e \cdot t)^2$ and $n_2 = 4t^2 LambertW(1/2e \cdot t)^2$. Since the *Lambert-W* function has infinite branches in the complex plane, we only consider the branches which have real-valued solutions with real arguments. In addition, since $t \geq 1$ (*t* is the total coding time) and the *Lambert-W* function is single-valued in the real plane for a positive argument, we can determine that $n_2 < 4t^2 \cdot (1/2e \cdot t)^2 = 1/e^2$, which is in conflict with the condition $n \geq 4$. Thus, n_2 can be excluded for further consideration.

n_1 is double-valued in the real plane since the argument of the function *LambertW*(x), $x = -1/2e \cdot t$, is in the region of $(-1/e, 0)$. We denote the double-valued results as

$$n_1(k,t) = 4t^2 LambertW(k, -1/2e \cdot t)^2, k = -1, 0, \tag{3.16}$$

where *k* can be any integer in the complex plane. For a real-valued solution of n_1, *k* can only be 0 and -1. Similarly, we can determine that $n_1(0,t) < n_1(0,t)\big|_{t=1} = 0.215$ and this principal value does not accord with the prescribed condition $n \geq 4$. Finally, we can determine by

checking that the result $n_1(-1,t)$ is the point where the objective function $g(n,t)$ achieves its minimum for any given parameter t. For example, given $t=5$, we can get three real-valued results as follows:

$$\begin{cases} n_1(-1,5) = 2390.936334, \\ n_1(0,5) = 0.1460863321, \\ n_2(0,5) = 0.1260584706, \end{cases} \tag{3.17}$$

where only $n_1(-1,5)$ meets the prescribed condition $n \geq 4$. Figure 3.8 shows the computational overhead versus the size of algebraic structure.

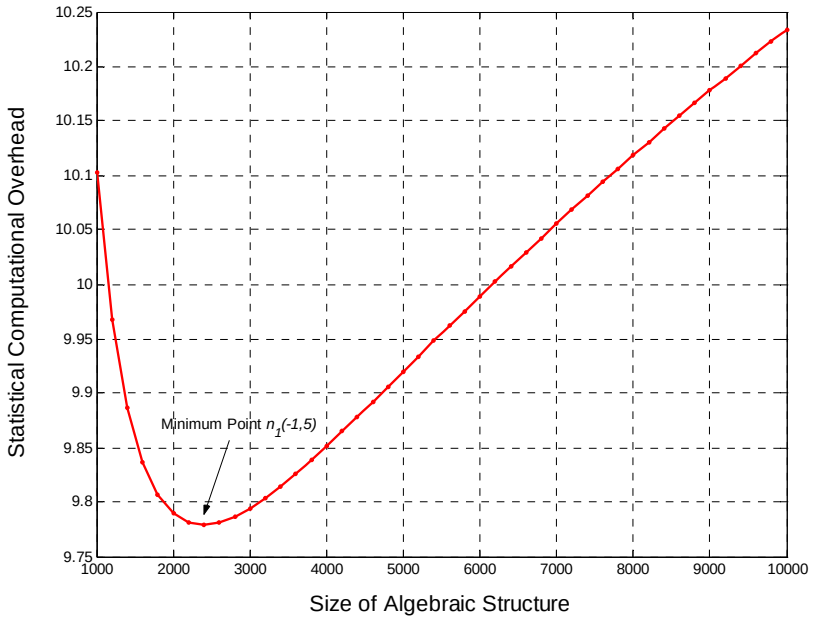

Figure 3.8: Computational Overhead vs. Size of Algebraic Structure (t=5)

After obtaining the minimum point $n_1(-1,t)$ of the objective function, we can find the closest positive integer n which is the product of two primes p and q, i.e., $n = pq$. The integer n can then be incorporated into Eq. (3.14) to achieve the minimum computational overhead.

3.5 Related Work

Several privacy-preserving schemes have been proposed, and they can be classified into three categories: proxy-based, mix-based, and onion-based. Proxy-based schemes include Crowds [42] and Hordes [43]. The common characteristic of these schemes is that they all employ one or more network nodes to issue service requests on behalf of the originator. In Crowds, for example, servers and even crowd members cannot distinguish the originator of a service request, since it is equally likely originating from any member of the crowd.

Chaum's mix based schemes include MorphMix [44] and Mixminion [45]. These schemes commonly apply techniques such as shaping, which divides messages into a number of fixed-sized chunks, and mixing, which caches incoming messages and then forwards them in a randomized order. These two techniques can be used to prevent attacks such as size correlation and time correlation.

Onion-based schemes include Onion Routing [46] and Onion Ring [47]. The common feature of these schemes is the chaining technique, which chains onion routers together to forward messages hop by hop to the intended recipient. The characteristic of this technique is that every intermediate onion router knows only about the router directly in front of and behind itself, respectively, which can protect user privacy if one or even several intermediate onion routers are compromised.

Network coding has privacy-preserving features, such as shaping, buffering, and mixing. However, network coding suffers from two primary types of attacks, *pollution attacks* [60] and *entropy attacks* [61]. *Pollution attacks* can be launched by untrusted nodes or adversaries through injecting polluted messages or modifying disseminated messages, which is fatal to the whole network due to the rapid propagation of pollution. In *entropy attacks*, adversaries forge non-innovative packets that are linear combinations of "stale" ones, thus reducing the overall network throughput.

To secure network coding, some solutions have been proposed and they can be classified into two categories according to different theoretical bases. Information-theory

based schemes [52] can detect or filter out polluted messages only at sinks, not at forwarders. Cryptography-based solutions include homomorphic hashing [61], homomorphic signatures [60], and secure random checksum [61]. These solutions either require an extra secure channel [61], or incur high computation overhead [60].

In summary, existing studies on secure network coding mainly focus on detecting or filtering out polluted messages [60]. Homomorphic cryptosystems are utilized to produce homomorphic signatures for the purpose of protecting the integrity of messages, which is substantially different from the objective of this research; in this research, we utilize homomorphic encryption functions in homomorphic cryptosystems to offer the confidentiality for GEVs; in this way, the proposed scheme can provide the packet flow untraceability and message content confidentiality, and finally achieves our privacy objective, source anonymity. Little attention has been paid to the privacy issues, especially to protecting the encoded messages from tracking or traffic analysis.

3.6 Summary

In this chapter, we have proposed an efficient network coding based privacy-preserving scheme against traffic analysis and flow tracing in multi-hop wireless networks. With the light-weight homomorphic encryption on Global Encoding Vectors (GEVs), the proposed scheme offers two significant privacy-preserving features, packet flow untraceability and message content confidentiality, which can efficiently thwart traffic analysis/flow tracing attacks. Moreover, with homomorphic encryption, the proposed scheme keeps the essence of random linear network coding, and each sink can recover the source messages by inverting the GEVs with a very high probability. The quantitative analysis and simulative evaluation on privacy enhancement and computational overhead demonstrate the effectiveness and efficiency of the proposed scheme.

Chapter 4
Preventing Traffic Explosion and Achieving Source Unobservability in Multi-hop Wireless Networks using Network Coding

In this chapter, we focus on preventing traffic explosion to ensure system availability while achieving the privacy objective of source unobservability in MWNs at the same time. As mentioned above, multi-hop wireless networks (MWNs) are very attractive and promising in many application scenarios, such as wireless sensor networks, military ad hoc networks, and wireless mesh networks. However, most of these application scenarios are security/privacy-sensitive. For example, a generic asset monitoring application, which is also called the *Panda-Hunter Game* [9], is susceptible to the leakage of source location privacy since it may lead to the loss of monitored assets. Another example is the command communication in military ad hoc networks, where an attacker may utilize back-tracing techniques to obtain the location information of critical nodes (e.g., the commanders) and then launch the so-called *Decapitation Strike* to destroy those critical nodes [11].

For those privacy-sensitive application scenarios, source unobservability is an attractive and desirable privacy property. Unobservability is the state of items of interest (IOIs) being indistinguishable from any other IOI of the same type [26]. Compared with anonymity, unobservability is a stronger privacy objective since, with respect to the same attacker, unobservability reveals always only a subset of the information that anonymity reveals. Specially, source unobservability means that it is not noticeable whether or not any source node transmits. It is very challenging to achieve source unobservability in MWNs. Most existing privacy-preserving solutions [42]-[47] are not designed for MWNs and can offer only a relatively weak privacy protection, such as anonymity or pseudonymity. Only a few schemes can achieve source unobservability by using dummy messages.

Shao et al propose a scheme called *FitProbRate* by utilizing dummy messages to achieve source unobservability, which is referred to as the statistically strong source

anonymity in [41]. *FitProbRate* can prevent global external attackers from conducting time correlation/rate monitoring attacks, but it cannot thwart message content correlation based traffic analysis or flow tracing even if messages are encrypted in an end-to-end manner [62]. If assisted by link-to-link encryption on each-hop transmission, *FitProbRate* can prevent external attackers from successfully launching these two attacks. However, this scheme is still vulnerable to internal attackers since they may have the access to secret keys. More importantly, *FitProbRate* has not taken into consideration the problem of the explosion of network traffic, which may lead to severe performance degradation or even service denial.

Yang et al propose two schemes, *PFS* (Proxy-based Filtering Scheme) and *TFS* (Tree-based Filtering Scheme), to prevent the explosion of network traffic while achieving source unobservability [63]. However, both the schemes rely on trusted sensor proxies to proactively filter dummy messages, making those proxies become performance bottlenecks and privacy-critical nodes. In other words, if a proxy node is compromised, all sensor nodes in its proximity will suffer privacy leakage. Such deficiencies in existing schemes motivate us to explore a more secure and more efficient privacy-preserving scheme to prevent the explosion of network traffic while achieving source unobservability.

Generally, if we use dummy messages to achieve source unobservability, it is argued that there is no way to filter dummy messages while preventing internal attackers at proxy nodes. The statement seems plausible since proxy nodes must be able to distinguish dummy messages from real ones to selectively drop dummy messages for the prevention of traffic explosion. The ability of proxy nodes may be utilized by the internal attackers to identify and track real messages for compromising source privacy.

We investigate the traffic explosion problem from a brand-new perspective. Instead of filtering or dropping dummy messages at trusted sensor proxies, we utilize network coding to absorb dummy messages at intermediate forwarders. To the best of our knowledge, this is the first research on how to utilize network coding to prevent the explosion of network traffic. The basic idea is to combine dummy and real messages in accordance with the network

coding principles to produce coded outgoing packets. In this way, dummy messages can be absorbed automatically by network coding at intermediate forwarders.

Network coding was first introduced by Ahlswede et al [48] and has been widely recognized as a promising information dissemination approach. Compared with conventional packet forwarding technologies, network coding allows coding/mixing operations at intermediate forwarders [48] and, thus, offers several significant advantages such as potential throughput improvement [53], transmission energy minimization [54], and delay minimization [37]. Network coding can also work as an erasure coding scheme to enhance the dependability of a distributed data storage system [55].

The deployment of network coding in MWNs can not only bring the above performance benefits, but also provide a possible way to efficiently prevent the explosion of network traffic while achieving source unobservability. Since coding operation is encouraged at intermediate forwarders, dummy messages can be combined with real ones and, thus, be absorbed into real traffic. In this way, internal attackers can be naturally thwarted since there is no necessity for intermediate forwarders to distinguish and drop dummy messages. However, all dummy messages cannot be combined into real traffic. An arbitrary dummy message may destroy coded packets when combined with them, leaving no way to recover real messages. Thus, only specially designed dummy messages can be used for the purpose of network coding and, further, can be absorbed by network coding.

In this chapter, we propose a privacy-preserving scheme, SUNC (Source Unobservability using Network Coding), which can prevent the explosion of network traffic while achieving source unobservability. With specially designed dummy messages, SUNC has the following three attractive properties.

1) Preventing Traffic Explosion: According to the network coding principle, SUNC can combine dummy messages with real ones at intermediate forwarders. With specially designed dummy messages, the combination can only be related to the real messages. We introduce a new concept called *dummy nullity* to describe this property of the specially

designed dummy messages. With this property, dummy messages do not destroy real messages when combined with them. In this way, dummy messages can be absorbed by network coding at intermediate forwarders; thus, the explosion of network traffic can be naturally prevented.

2) Source Unobservability: SUNC is based on network coding. As we know from Chapter 3, SUNC has many attractive privacy-preserving characteristics, such as shaping, buffering, and mixing. Shaping can resist size correlation attacks, buffering can resist time correlation attacks, and mixing can resist content correlation attacks. These three characteristics together offer unlinkability between incoming and outgoing packets. Unlinkability is a critical privacy feature for thwarting traffic analysis/flow tracing attacks and achieving source anonymity. According to [26], unobservability can be achieved when a mechanism used to achieve anonymity is appropriately combined with dummy traffic. Thus, by utilizing dummy messages, SUNC can achieve source unobservability.

3) Forwarder Blindness: Since dummy messages are absorbed by network coding, there is no need for intermediate forwarders to distinguish dummy messages from real ones. Thus, forwarders can be designed to be unaware of the authenticity of a message. We call this property "forwarder blindness", which means that a forwarder cannot distinguish dummy messages from real ones. The "forwarder blindness" property is attractive since, even if all forwarders are compromised, adversaries still cannot tell the authenticity of a message and, therefore, cannot compromise source unobservability. In SUNC, we utilize homomorphic encryption functions to provide forwarder blindness.

The remainder of this chapter is organized as follows. Section 4.1 gives the threat models. In Section 4.2, the proposed privacy-preserving scheme is presented in detail. Security analysis and performance evaluation are given in Section 4.3 and 4.4, respectively, followed by a summary in Section 4.5.

4.1 Threat Models

In this chapter, we consider the following two kinds of attackers.

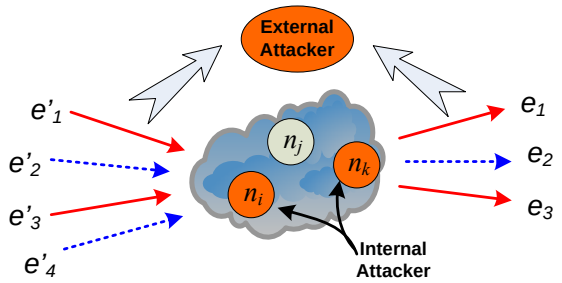

Figure 4.1: Attack Model: External Attacker and Internal Attacker

External Attacker: An external attacker can be a global passive eavesdropper who has the ability to observe all network links, as shown in Figure 4.1. An external attacker can examine the tags and message content, and, thus, link outgoing packets with incoming ones. Further, even if messages are encrypted in an end-to-end manner, it is still possible for a global external attacker to trace packets by analyzing and comparing the message ciphertext.

Internal attacker: An internal attacker may compromise many intermediate nodes (excluding source and sink nodes), as shown in Figure 4.1. Link-to-link encryption is vulnerable to internal attackers since they may already have obtained the decryption keys, and, thus, the message plaintext may be easily recovered for analysis.

Both internal and external attackers may perform more advanced traffic analysis/flow tracing techniques, including size correlation, time correlation/rate monitoring, and message content correlation [58]. Thus, they can further explore these techniques to deduce the forwarding paths [41], and then back-trace to the source node.

4.2 The Proposed Privacy-Preserving Scheme

In this section, we propose a novel privacy-preserving scheme (SUNC) for MWNs to prevent the explosion of network traffic while achieving source unobservability. As a multi-source

network coding scheme [64], SUNC can be divided into four functional modules: parameter setting, generating dummy traffic, embedding real traffic, and recovering real messages.

4.2.1 Parameter Setting

Since random coding can only be performed on the packets in the same generation, the generation management is the first issue to be addressed. In this study, we utilize a coarse synchronization mechanism [65] to determine the generation number of a packet. Each node can be equipped with a low-cost timer, and the time is divided into fixed time slots. All messages, no matter whether they are real or dummy, sent out in a time slot belong to the same generation, and each message is prefixed with the corresponding generation number for further recoding at intermediate forwarders.

Similar to the research in Chapter 3, we choose the Paillier cryptosystem as the HEF for SUNC. In the Paillier cryptosystem [40], given a message *m* and the public/encryption key (*n, g*), the encryption function can be expressed as follows,

$$E(m) = g^m \cdot r^n \pmod{n^2}. \tag{4.1}$$

where $r \in \mathbb{Z}_n^*$ is a random factor. The Paillier cryptosystem can keep the random coding feature of SUNC, since

$$E(m_1) \cdot E(m_2) = g^{m_1+m_2} \cdot (r_1 r_2)^n \pmod{n^2} = E(m_1 + m_2). \tag{4.2}$$

In addition, the random factor *r* in the Paillier cryptosystem can be employed to generate random dummy messages of nullity, which can be achieved by setting *m* to be zero.

For a unicast application, e.g., the endangered animal monitoring in wireless sensor networks, the sink node is preloaded with a decryption/private key *dk*, and the corresponding encryption/public key *ek* is published to all other nodes. For a multicast application, e.g., the command communication in military ad hoc networks, each sink node in a multicast session is preloaded with a same decryption key *dk*, and the corresponding encryption key is published to all nodes in the network.

4.2.2 Generating Dummy Traffic

Since network coding and dummy traffic are employed to provide source unobservability, how to generate dummy messages and how to send out them are critical. Several schemes, such as *ConstRate* [66] [67], *ProbRate* [63], and *FitProbRate* [41], have already been proposed for sending out dummy messages to thwart rate monitoring attack. Here, we focus on how to generate dummy messages.

In SUNC, dummy messages should have the following three properties: homomorphism, randomness, and dummy nullity. Since messages in plaintext are easy to analyze, we make dummy messages in SUNC always appear in their ciphertext format. Thus, homomorphism is the first requirement for the possibility of coding through operating the ciphertext directly. Secondly, dummy messages should have randomness for thwarting content correlation attacks even if they are designed to appear only in their ciphertext format; because if attackers can track dummy messages, real messages can be easily exposed. Thirdly, to thwart internal attackers, the authenticity of a message is concealed from intermediate nodes. For an intermediate node to conduct random coding operations on the authenticity-unknown messages, the property of dummy nullity is required. Only with dummy nullity, dummy messages can be combined with real ones without destroying the coded packets; real messages can be kept mint in coded packets and, finally, be recovered intact from the coded packets. Dummy nullity can be implemented through, when a dummy message is generated, setting the message plaintext to be null.

Let the dimension of GEVs be h and the length of real messages be l in terms of codewords. Dummy messages should have the same length as a tagged real message, i.e., $h+l$. Thus, a dummy message can be generated as follows.

Algorithm 1: Generating Dummy Traffic
Input: the length of a dummy message $h+l$;
Output: a dummy message $\boldsymbol{c}$.
Procedure GDT:
1. choose $h+l$ random factors $(r_1,\cdots,r_{h+l})$ from $\mathbb{Z}_n^*$;
2. calculate dummy codewords of nullity by letting $m_j = 0$:
$$d_j = E_{ek}(m_j)\Big|_{m_j=0,r=r_j} = g^{m_j}\cdot r_j^{\,n} (\mathrm{mod}\, n^2) = r_j^{\,n} (\mathrm{mod}\, n^2);$$
3. unite d_j to form a dummy message $(d_1,\cdots,d_{h+l})$;
4. return $\boldsymbol{c} = (d_1,\cdots,d_{h+l})$.

The former h codewords can be regarded as the GEV of this dummy message, while the latter l codewords can be regarded as the content of this dummy message.

The steps to generate a real message are the same as those shown in Algorithm 1 except that the codewords of real messages are used instead of a zero vector. The former h codewords are the GEV (normally a unit vector) prefixed to this real message and the latter l codewords are the content of the real message, as shown in Eq. (4.3).

$$\boldsymbol{m} = (\boldsymbol{g}, \boldsymbol{x}) = (g_1,\cdots,g_h,x_1,\cdots,x_l) \tag{4.3}$$

How to determine the dimension h of GEVs and choose a GEV is a critical performance issue, which will be discussed in Section **4.4**. Notice that real messages also have the former two properties, i.e., homomorphism and randomness, but they do not have the third property, dummy nullity.

4.2.3 Embedding Real Traffic

According to the rate control policies in [66] [67] [63] [41], a node needs to send out dummy or real messages at the intervals of a specified probability distribution. If a node has not received any message from its neighbors, it sends out a dummy message, which can be generated according to Algorithm 1. If the node has already received several messages from its neighbors, it can compute the outgoing message from those incoming ones according to the principles of random linear network coding. Real messages, if exist, are thus embedded into the traffic flow in the network.

For the sake of presentation, we define a message as:

$$\boldsymbol{c_i} = g^{\boldsymbol{m_i}} \cdot \boldsymbol{r}_i^{n} \pmod{n^2}, \tag{4.4}$$

where $\boldsymbol{m_i}$ is a vector of codewords representing the message plaintext, and $\boldsymbol{r_i}$ is a vector of random factors. For a real message, the corresponding $\boldsymbol{m_i}$ is not a null vector; while for any dummy message, the corresponding $\boldsymbol{m_i}$ is a null vector.

Algorithm 2: Embedding Real Traffic
Input: incoming messages $\boldsymbol{c_i}$'s;
Output: an outgoing message $\boldsymbol{c}$.
Procedure ERT:
1. choose a random factor $\beta_i \in \mathbb{Z}_{n^2}^*$ for each $\boldsymbol{c_i}$;
2. calculate the outgoing message $\boldsymbol{c}$:
$$\boldsymbol{c} = \prod_i \boldsymbol{c_i}^{\beta_i} \pmod{n^2};$$
3. return $\boldsymbol{c}$.

Algorithm 2 shows the procedure of embedding real traffic, if there is, into dummy traffic. From the algorithm, we have:

$$\begin{aligned} \boldsymbol{c} &= \prod_i \boldsymbol{c_i}^{\beta_i} \pmod{n^2} \\ &= \prod_i (g^{\boldsymbol{m_i}} \cdot \boldsymbol{r}_i^{n})^{\beta_i} \pmod{n^2} \\ &= \prod_i g^{\beta_i \boldsymbol{m_i}} \cdot \prod_i \boldsymbol{r}_i^{\beta_i n} \pmod{n^2} \\ &= g^{\sum_i \beta_i \boldsymbol{m_i}} \cdot (\prod_i \boldsymbol{r}_i^{\beta_i})^{n} \pmod{n^2} \end{aligned} . \tag{4.5}$$

If letting $\boldsymbol{m} = \sum_i \beta_i \boldsymbol{m_i}$ and $\boldsymbol{r} = \prod_i \boldsymbol{r}_i^{\beta_i}$, we get

$$\boldsymbol{c} = g^{\boldsymbol{m}} \cdot \boldsymbol{r}^{n} \pmod{n^2}. \tag{4.6}$$

From Eq. (4.6), the outgoing message is actually the ciphertext of message $\boldsymbol{m}$, which is a linear combination of those incoming messages $\boldsymbol{m_i}$. This computation is feasible ascribing to the homomorphism of the Paillier cryptosystem.

If all $\boldsymbol{c_i}$'s are dummy messages, all corresponding $\boldsymbol{m_i}$'s are null vectors; the procedure in algorithm 2 mixes all the incoming dummy messages together to produce a new outgoing dummy message, since the outgoing message plaintext is $\boldsymbol{m} = \sum_i \beta_i \boldsymbol{m_i} = \boldsymbol{0}$. Otherwise, the

outgoing message plaintext is actually a linear combination of those incoming real messages $\boldsymbol{m}_i$, since for those incoming dummy messages, $\boldsymbol{m}_i = \boldsymbol{0}$.

Notice that a node can execute Algorithm 2 without knowing the authenticity of an incoming packet; neither does it know the authenticity of the outgoing packet. This so-called *forwarder blindness* is really attractive and beneficial to privacy protection. Only the source node knows the authenticity of a message; the sink node will know it after applying the decryption operation on the message. We will have a detailed discussion in Section **4.3** about the impacts of these precious properties to system security.

4.2.4 Recovering Real Messages

After receiving a message, which is in its ciphertext format, a sink node can decrypt it to get the message plaintext $\boldsymbol{m}_i$, as shown in Eq. (4.7).

$$\begin{aligned} &(d_{i,1}, \cdots, d_{i,h+l}) = \boldsymbol{c}_i \\ &m_{i,j} = D_{dk}(d_{i,j}), \quad j \in [1, h+l] \\ &\boldsymbol{m}_i = (m_{i,1}, \cdots, m_{i,h+l}) \end{aligned} \tag{4.7}$$

If the received message is a dummy message, the decrypted message plaintext $\boldsymbol{m}_i$ is a null vector; otherwise, $\boldsymbol{m}_i$ is a linear combination of related original messages.

$$\begin{aligned} &G^{-1} \cdot G = I \pmod{n} \\ &G = (\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_h)^T \\ &\boldsymbol{g}_i = (m_{i,1}, \cdots, m_{i,h}) \end{aligned} \tag{4.8}$$

After collecting enough (typically h) coded messages, the sink node can find the inverse matrix of the prefixed GEVs in those coded messages, as shown in Eq. (4.8), and then apply the inverse matrix to the coded messages to recover the original messages, as shown in Eq. (4.9).

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_h \end{bmatrix} = G^{-1} \cdot \begin{bmatrix} m_{1,h+1} & \cdots & m_{1,h+l} \\ \vdots & \ddots & \vdots \\ m_{h,h+1} & \cdots & m_{h,h+l} \end{bmatrix} \tag{4.9}$$

In SUNC, due to the existence of dummy messages, the GEM is probably invertible if the number of received GEVs is no less than the rank of the GEM. In other words, if the number of received non-dummy messages is not less than the number of detected real events, the original real messages are probably able to be recovered. A critical performance issue is the invertible probability of the GEVs. We will discuss in Section **4.4** on how to analyze and enhance the invertible probability.

4.3 Security Analysis

In this section, we will discuss the security properties of SUNC, i.e., what kind of objectives can be achieved, what kind of adversaries can be thwarted, what kind of attacks can be prevented, and how well the privacy protection can be provided.

4.3.1 Information Security and Privacy

SUNC can prevent the explosion of network traffic while providing source unobservability for privacy-sensitive MWNs, such as asset monitoring wireless sensor network [9] and military ad hoc networks [11]. Specially, the precious privacy property of source unobservability provided by SUNC relies on the following features.

Availability: The explosion of network traffic is almost an inevitable subsequence of using dummy traffic to achieve source unobservability. Traffic explosion may lead to severe performance degradation or even service denial and greatly affect the availability of the communication system. To guarantee the availability of the applications, the explosion of network traffic must be prevented. SUNC utilizes network coding to absorb specially designed dummy messages at intermediate forwarders. In this way, network traffic is made under a full control, traffic explosion can be naturally prevented, and, thus, the availability can be guaranteed.

Source Unobservability: SUNC is based on network coding, and the inherent mixing feature of network coding provides SUNC with packet unlinkability, which eliminates the

possibility of linear analysis on incoming and outgoing messages. Packet unlinkability can effectively prevent attacks such as back-tracing, and, thus, can help SUNC to achieve source anonymity. When combined with dummy messages, SUNC can achieve provide source unobservability [26].

Forwarder Blindness: Intermediate nodes may be compromised, and adversaries may reside in the nodes to become an internal attacker. To prevent the possible privacy leakage at intermediate nodes, SUNC provides the message authenticity blindness to those intermediate forwarders by employing the public-key HEF of the Paillier cryptosystem. In SUNC, forwarders can blindly conduct coding operations on messages without knowing the decryption keys or performing the expensive decryption operations; neither are they required to know the authenticity of a message, since they do not need to discard the dummy messages. Through network coding, dummy messages are automatically absorbed at intermediate forwarders with only the random factors included into the newly coded messages. With forwarder blindness, the privacy leakage at intermediate nodes can be effectively prevented, and SUNC can resist internal attackers.

In summary, preventing traffic explosion helps to guarantee the availability of the communication system, source unobservability is achieved by utilizing network coding and dummy messages, and forwarder blindness is provided by HEFs. The three information security and privacy features together assist SUNC to successfully achieve its final objectives: providing communication service without privacy leakage. In addition, according to [26], the source unobservability is a very strong privacy feature and implies a lot of privacy properties, such as source anonymity, relation unobservability, source undetectability, etc.

4.3.2 Adversaries and Attacks

SUNC can prevent global external attackers, since external attackers can neither directly determine the authenticity of a message directly nor find the possibility for a message to be authentic through packet tracing or analysis. In addition, SUNC can thwart internal attackers,

because internal attackers at intermediate forwarders do not have decryption keys, and neither can they link incoming and outgoing packets together.

As described above, some rate control schemes, such as *ConstRate* [67], *ProbRate* [63], and *FitProbRate* [41], can be employed in combination with SUNC. Thus, the rate monitoring attack can be effectively resisted since adversaries cannot distinguish the real source from dummy sources through statistically monitoring their data rates.

Traffic analysis and flow tracing attacks can also be effectively resisted since the mixing feature of network coding is extensively exploited and packet unlinkability is achieved against not only global external attackers but also internal attackers. Specially, the commonly-used techniques of traffic analysis and flow tracing attacks, such as size correlation, time correlation, and message content correlation, are effectively prevented: size correlation becomes useless in network coding based system since each packet in the system is chopped of the same length; time correlation is statistically thwarted due to the existence of buffering and rate control; message content correlation is completely disabled by the mixing feature of network coding combined with HEFs.

Another typical attack, called message content based back-tracing attack, is also effectively prevented by SUNC. Message content based back-tracing attack is conducted by comparing message content no matter whether it is in plaintext or ciphertext format, and tracing messages up to several hops to distinguish real messages from dummy ones, since dummy messages are usually discarded after one hop transmission while real messages will probably be forwarded hop-by-hop to its destination. Once a real message is distinguished from dummy ones, the corresponding source node may be easily revealed by tracing the message backward to the originator. This attack is quite dangerous to some existing schemes [41], while SUNC can effectively resist this attack since the mixing feature of network coding is extensively exploited and the packet unlinkability is achieved especially against content correlation based back-tracing.

4.4 Performance Evaluation

In this section, we will conduct performance analysis and evaluation on SUNC in terms of invertible probability of GEVs, communication efficiency, and computational overhead. Moreover, the linear dependence of received GEVs is also extensively analyzed since it is closely related to the computational overhead of message recovering at sink node.

4.4.1 Invertible Probability

Generally, the invertible probability of GEVs is closely related to three parameters: the size of algebraic structure n, the dimension of GEVs h, and the total times of random coding t [36] [49]. The results on invertible probability achieved in [56] are still applicable to SUNC. According to ***Corollary 2*** in [56], for example, the influence of changing h on invertible probability is confined within the magnitude of $\boldsymbol{O}(n^{-5})$ when $h \geq 4$ and, thus, can be neglected.

However, some other factors need to be considered. A significant factor is the real traffic rate of a node. Consider an event reporting wireless sensor network [41] with a deterministic GEV management policy, i.e., each node in the network being assigned a specific codeword in GEVs.

As we know, SUNC is a multi-source network coding scheme [64], and generations of packets are divided according to the specified time slots. Let the real traffic rate of a node be λ messages per second, and the specified time slots be of the length $1/\mu$ seconds. If *ConstRate* [66] is adopted for rate control, the invertible probability of GEVs is degraded by

$$DF_{ConstRate} = \begin{cases} 1 & \lambda \leq \mu \\ 2 - \lambda / \mu & \mu < \lambda < 2\mu \\ 0 & \lambda \geq 2\mu \end{cases}, \tag{4.10}$$

where *DF* means the degrading factor of invertible probability. If *ProbRate* [63] or *FitProbRate* [41] is employed, the degrading factor is

$$DF_{ProbRate} = (1 + \lambda / u) \cdot e^{-\lambda / u}. \tag{4.11}$$

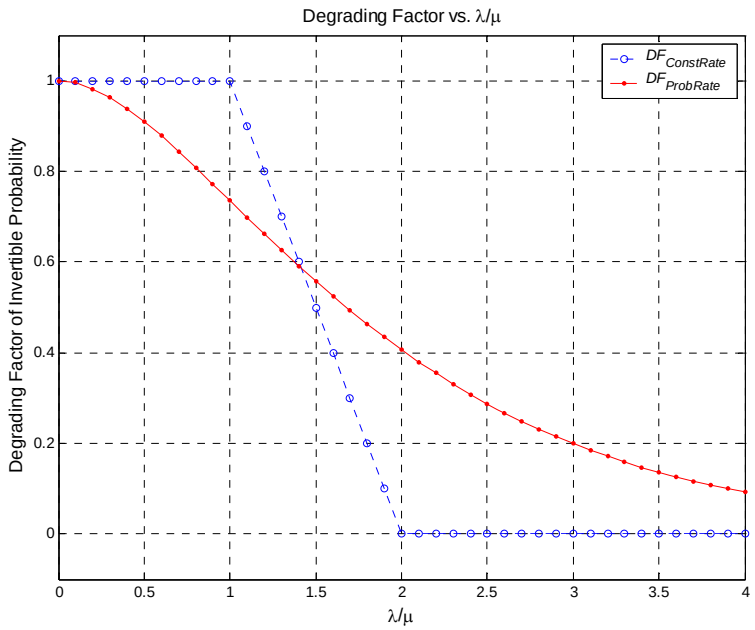

Figure 4.2: Degrading Factor of Invertible Probability vs. λ / μ

The analytical results are plotted in Figure 4.2. It can be seen that *ConstRate* outperforms *ProbRate* and *FitProbRate* in terms of invertible probability when $\lambda / u < 1.4$, while otherwise *ConstRate* performs worse than them. Normally, high invertible probability is preferred; thus, the condition $\lambda / u < 1.4$ is inclined to be satisfied.

For practical applications, several methods can be employed to improve the invertible probability. From Figure 4.2, reducing the value of λ / u can increase the invertible probability. To achieve this goal, we can set a larger value for parameter μ or discard some real messages to reduce λ.

4.4.2 Communication Efficiency

In SUNC, the communication traffic is relatively steady, since dummy messages will be produced and sent out if no real events are detected. Thus, the actual communication efficiency depends on the practical demands. However, practical demands cannot be satisfied infinitely and the upper bound of communication efficiency can be calculated.

The upper bound can be calculated independently by three steps. The first step is called encryption efficiency E_e. In the Paillier cryptosystem, each codeword is expanded from $\mathbb{Z}_n$ to $\mathbb{Z}_{n^2}^*$ after encryption, doubling the message length. Thus, the encryption efficiency E_e is 1/2. The second step is called tagging efficiency E_t. As indicated in SUNC, the tagging efficiency E_t can be calculated as $E_t = l/(h+l)$. In other words, the tagging overhead is $O_t = (1-E_t)/E_t = h/l$. The third step is called dummy efficiency E_d. The dummy efficiency E_d can be bounded as $E_d \le R_{\text{sink}} \Big/ \sum_i r_i$, where R_{sink} is the message arrival rate of sink node, and r_i is the message departure rate of node i. In summary, the overall communication efficiency E_c can be calculated as follows:

$$\begin{aligned} E_c &= E_e \cdot E_t \cdot E_d \\ &= l \cdot R_{\text{sink}} / 2(h+l) \cdot \sum_i r_i \end{aligned} . \tag{4.12}$$

4.4.3 Computational Overhead

The computational overhead of SUNC can be investigated respectively from the following four aspects: generating dummy traffic, generating real traffic, embedding real traffic, and recovering real traffic.

Generating Dummy Traffic: According to Algorithm 1, generating a dummy message requires $(h+l)$ exponentiation operations and $(h+l)$ modulus operations, which is equivalent to $(h+l)\cdot \log n^2$ multiplication operations. Fortunately, dummy messages can be generated offline, and no online computational overhead is required.

Generating Real Traffic: According to Algorithm 1 and Eq. (4.3), generating a real message requires $(h+l)$ encryption operations, which is equivalent to $2\cdot(h+l)$ exponentiations, $(h+l)$ multiplications and $(h+l)$ modulus operations. Considering the GEV is normally a unit vector, we can reduce the number of exponentiations to $(h+2\cdot l)$, in which $(h+l)$ exponentiations can be calculated offline and the other l exponentiations must be computed online. Thus, the overall computational overhead is $(h+l)\cdot \log n^2$ online and $l \cdot \log n^2$ offline multiplications.

Embedding Real Traffic: Let the number of incoming messages in a generation be θ, which is closely related to the sending rates of source nodes. According to Algorithm 2, generating an outgoing message requires $(h+l)\cdot\theta$ exponentiation operations, $(h+l)\cdot(\theta-1)$ multiplications, and $(h+l)$ modulus operations. In practice, almost all these operations can be performed offline. Thus, the computational overhead is $(h+l)\cdot\theta\cdot\log n^2$ offline multiplications.

Recovering Real Traffic: According to Eq. (4.7), the computational overhead to decrypt a message is $(h+l)$ exponentiations, $(h+l)$ divisions, $(h+l)$ multiplications, $(h+l)$ subtractions, and $2\cdot(h+l)$ modulus operations. Thus, the overall computational overhead to decrypt a message is $(h+l)\cdot\log n^2$ multiplications.

Notice that for applications such as the event reporting wireless sensor networks, the real traffic is pretty sparse and many received messages are actually dummy messages. A dummy message can be discerned through decrypting and checking if the prefixed GEV, i.e., the former h codewords, is a null vector. Thus, the corresponding computational overhead can be reduced to $h\cdot\log n^2$ multiplications.

For a real message, we can analyze the linear dependence of this message on other received messages immediately after decrypting the prefixed GEV, since the linear dependence among these messages only relies on their GEVs. If a real message is dependent on other received messages, it is not necessary to decrypt the latter l codewords, since this message has no innovation and the whole message can be discarded. The computational overhead of decrypting such a message can be greatly reduced to $h\cdot\log n^2$ multiplications.

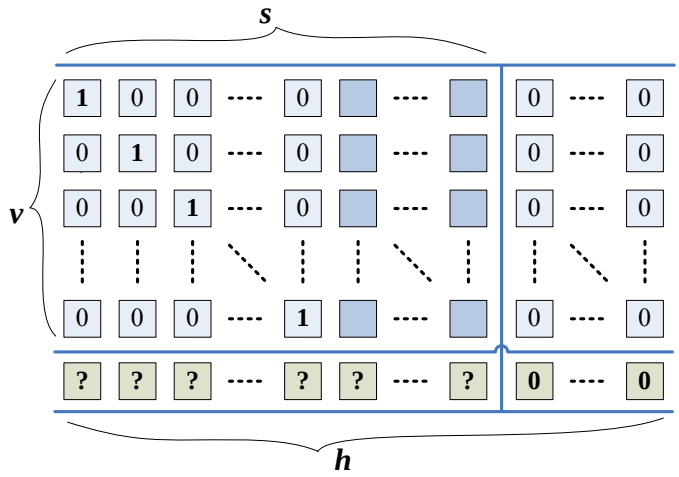

Figure 4.3: Linear Dependence Analysis

Linear dependence analysis can be conducted based on the Gaussian Elimination method. Generally speaking, the coefficients above the leading coefficient in each row cannot be guaranteed to be zeros in the row echelon form of GEVs; however, we can extend the general Gaussian Elimination method to uniform those coefficients as follows. Let the number of real events be *s*, and the number of real messages that arrived at the sink node be *v*. When another message arrives, linear dependence between the message and the *v* buffered messages can be bounded according to the following steps.

Step 1 is to eliminate the first *v* codewords of the newly arrived message, which can be done by multiplying the corresponding row vector by a corresponding value, and then subtracting the result from the codeword vector of the newly arrived message. For example, to eliminate the first codeword, we multiply the first vector by the value of the first codeword, and then subtract the result from the vector of the newly arrived message, as shown in Figure 4.3. This step requires $v \cdot (s - v)$ multiplications (we do not calculate subtractions here).

If the newly arrived message turns into a zero vector after the step 1, it is linearly dependent upon the buffered messages. Thus, it has no innovation to the sink node and can be discarded immediately. Otherwise, we will continue to change the messages into row echelon form for future analysis. Step 2 is to normalize the (*v*+*1*)th codeword, which can be

done by multiplying its inverse. This step requires 1 inversion and $(s-v-1)$ multiplications. Step 3 is to eliminate the (v+*1*)th codewords of buffered messages. The operations are similar to those in step 1. The difference is that in step 3, the codewords from different buffered messages are eliminated. This step requires $v\cdot(s-v-1)$ multiplications.

In summary, for a newly arrived message which is linearly dependent on buffered messages, the computational overhead to determine the linear dependence is $v\cdot(s-v)$ multiplications; for a message which is linearly independent on buffered messages, the computational overhead is 1 inversion and $v(s-v)+(v+1)(s-v-1)$ multiplications. If letting $w=s-v$, we get

$$\delta(v,w)=v\cdot w+(v+1)\cdot(w-1), \tag{4.13}$$

where $\delta(v,w)$ denotes the number of multiplications required for analyzing the (v+*1*)th real message.

If all messages at the sink node are linearly independent messages, we have

$$\begin{aligned}\Delta(v+1,w-1)&=\Delta(v,w)+\delta(v,w)\\&=\Delta(v,w)+v\cdot w+(v+1)\cdot(w-1)\end{aligned} \tag{4.14}$$

where $\Delta(v,w)$ is the total number of multiplications required after v real messages are received. Thus, the total number of multiplication required for analyzing all messages is

$$\Delta(s,0)=2\cdot\sum_{v=1}^{s}v\cdot(s-v)=s\cdot(s^2-1)/3. \tag{4.15}$$

In practice, however, many messages are linearly dependent messages. Let the number of linearly dependent messages be k when the sink node has v messages. We have

$$\Delta(v+1,w-1)=\Delta(v,w)+\delta(v,w)+k\cdot v\cdot(s-v), \tag{4.16}$$

where k is a random variable independent of v if the arrival of real events is a Poisson process. Thus, the average number of multiplications required for linear dependence analysis is

$$\begin{aligned}\Delta(s,0)&=(2+\bar{k})\cdot s\cdot(s^2-1)/6\\&=(1+\mu/\lambda)\cdot s\cdot(s^2-1)/6\end{aligned}, \tag{4.17}$$

where $\bar{k}$ is the average value of k, λ is the arrival rate of real events, and μ is the arrival rate of messages at sinks.

It is well known that the inversion process can be done by extend the GEM with an augmented unit matrix. Thus, the computational overhead of decoding and recovering original messages is $\boldsymbol{O}(s^3)$, which is much less than that of conventional decoding operations in network coding, $\boldsymbol{O}(h^3)$, when real events are sparse, since $\boldsymbol{O}(s^3) \ll \boldsymbol{O}(h^3)$ when $s < h$.

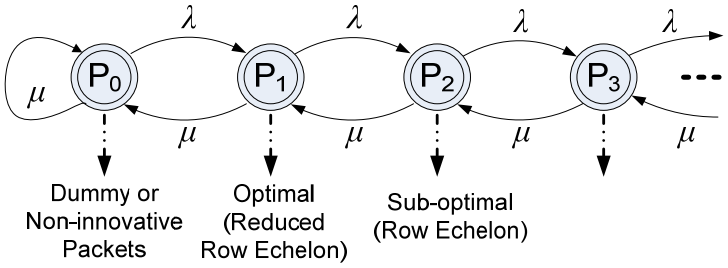

Figure 4.4: The Average Complexity of Linear Dependence Analysis

Notice that in SUNC the arrival of real events is sparse and distributed, making the arrival of innovative packets at the sink sparse and distributed too. Thus, the received GEVs usually form a matrix in reduced row echelon form, which greatly reduces the complexity of linear dependence analysis on GEVs. For example, if received GEVs naturally exhibit a reduced row echelon form, the linear independence of them can be determined immediately by observing the matrix form, without inducing any computational overhead.

Since the echelon form of GEM is closely related to the complexity of linear dependence analysis, we simulate the process of real event arrival and message sending as a queuing model shown in Figure 4.4. From the model, we can derive the average computational complexity as:

$$\Delta^e = \sum_{i=1}^{s} (C_i(q) + \bar{k} \cdot D_i(q)) \cdot P_i(q), \tag{4.18}$$

where $P_i(q)$ is the probability for the queue length being q in time slot i, $C_i(q)$ and $D_i(q)$ are the computational overhead of handling an innovative and a non-innovative message, respectively, both in terms of multiplications. Following the above analysis steps, we can get:

$$C_i(q) = q \cdot (q-1), \tag{4.19}$$

$$D_i(q) = \begin{cases} q-1, & q > 0 \\ 0 & q = 0 \end{cases}. \tag{4.20}$$

If *ConstRate* [66] is adopted as the rate control, we can get:

$$P_i(q) = (\lambda / \mu)^q \cdot e^{-\lambda/\mu} / q!. \tag{4.21}$$

Thus, the average computational complexity is

$$\Delta^e_{ConstRate} = s \cdot (\frac{\lambda^2}{\mu^2} - \frac{(u-\lambda)^2}{\lambda \cdot \mu} + \frac{\mu - \lambda}{\lambda \cdot e^{\lambda/\mu}}). \tag{4.22}$$

If *ProbRate* [63] or *FitProbRate* [41] is employed, we get:

$$P_i(q) = (1 - \lambda / \mu) \cdot (\lambda / \mu)^q. \tag{4.23}$$

Thus, the average computational overhead is

$$\Delta^e_{ProbRate} = s \cdot (\frac{\lambda \cdot (\lambda^2 + \mu^2)}{\mu \cdot (\mu - \lambda)^2}). \tag{4.24}$$

The average computational overhead of linear dependence analysis in the above two cases is plotted in Figure 4.5. It can be seen that the average computational overheads both increase with the ratio of λ and μ, and that of *ConstRate* increases much slower than that of *ProbRate* and *FitProbRate*.

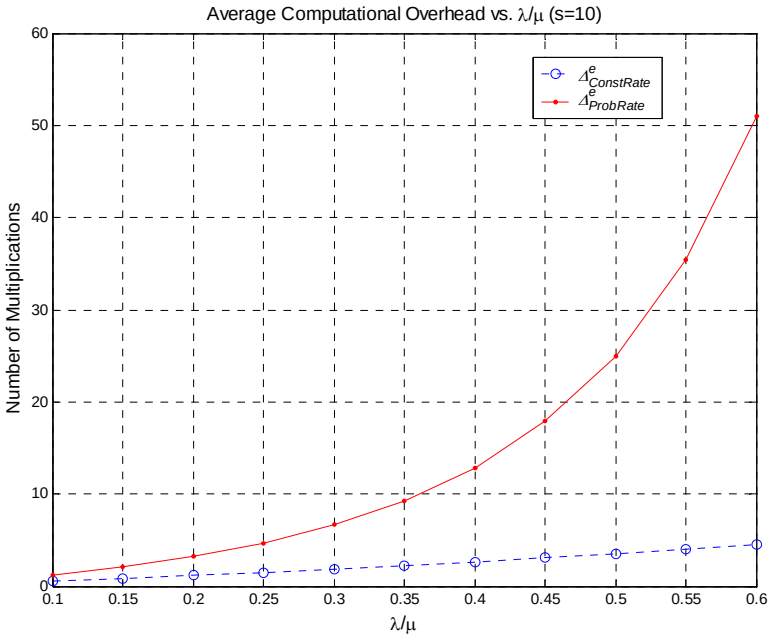

Figure 4.5: Average Computational Overhead

In summary, the average computational overhead of linear dependence analysis is coarsely given in Eq. (4.17); if *ConstRate* is chosen, the average computational overhead of linear dependence analysis is refined in Eq. (4.22); if *ProbRate* or *FitProbRate* is chosen, the average computational overhead is given in Eq. (4.24), all in terms of multiplications. From Eq. (4.22) and (4.24), it can be seen that both average computational overhead is linear to the number of real events *s*, meaning that the average computational overhead is far below the worst case.

4.5 Summary

In this chapter, we have proposed a privacy-preserving scheme, called SUNC, to prevent traffic explosion while achieving source unobservability. Dummy messages which are used for privacy preservation can be absorbed by network coding at intermediate forwarders, and, thus, the explosion of network traffic can be naturally prevented. On the other hand, network

coding helps to offer source anonymity, which can yield source unobservability when combined with dummy messages. In addition, assisted by homomorphic encryption functions, SUNC exhibits several attractive features such as forwarder blindness. With this feature, SUNC can not only thwart global external attackers, but also resist internal attackers.

Chapter 5
Cooperative Peer-to-Peer Information Exchange in Network Coding Enabled Wireless Networks

In this chapter, we emphasize how to enhance the data availability among peers if the remote base station is temporarily unavailable, and propose a cooperative peer-to-peer information exchange scheme based on network coding. Network coding has been widely recognized as a promising information dissemination approach to improving network performance [48] by allowing and encouraging coding operations at intermediate forwarders. Primary applications of network coding include file distribution [68] and multimedia streaming [50] in peer-to-peer (P2P) overlay networks, data persistence in sensor networks [69], and information delivery in wireless networks [70]. Incorporation of network coding into these applications brings many benefits such as throughput improvement [37], energy efficiency [71], and delay minimization [72].

Network coding can be employed to solve the Cooperative Peer-to-peer Repair (CPR) problem [73], where centralized and distributed CPR algorithms are proposed based on observed heuristics. The heuristics reflect some intuitive superficies instead of the essences of network coding based information exchange. In addition, the undetermined parameters in CPR algorithms constitute another open issue: how to tune them to adapt the scheduling algorithms. Such deficiencies motivate us to explore a more insightful scheme to maximize wireless coding gain, i.e., the benefit of combining network coding and wireless broadcast [70].

The scheduling issue in the CPR problem can be reduced to a *peer scheduling* issue by making nodes (peers) send coded packets which are combinations of all packets in a node [68]. Specifically, the *peer scheduling* issue is about how to intelligently schedule the transmission opportunities among peers to maximize the wireless coding gain. Due to the shared wireless channel and the de facto half-duplex transmission feature, peer scheduling

policies have a direct impact on the overall network throughput. In many cases, the gap between the optimal and the average is huge.

Most current research focuses on block scheduling problems. Besides opportunistic snooping neighbor states, COPE [70] successfully handles the block scheduling problem by intelligently XOR-ing packets. A multi-partner scheduling scheme [74] employs the Deadline-aware Network Coding technique to adjust the coding window for meeting the time sensitive requirement of media streaming service. An energy-efficient NBgossip scheme [75] utilizes network coding for neighborhood gossip in sensor ad hoc networks. The Rarest First algorithm is advocated through real experiments from being replaced with source or network coding in the Internet [76]. The rarest first idea can be employed in wireless network coding. However, directly applying this idea to peer scheduling is not necessarily optimal.

In this chapter, we redefine a *peer scheduling* problem in network coding enabled wireless networks [73]. Based on the summarized peer scheduling principles, we propose a cooperative Peer-to-peer Information Exchange (PIE) scheme with an efficient light-weight peer scheduling algorithm. In addition to the rarest first principle on blocks, we take into consideration the freshness of peers, which is a measurement on how much innovation a peer has against other peers. PIE can not only fully exploit the broadcast nature of wireless channels, but also take advantage of cooperative peer-to-peer information exchange. Qualitative analysis and extensive simulations demonstrate its effectiveness and efficiency.

The remainder of the chapter is organized as follows. In Section 5.1, the network model is given. In Section 5.2, we present the peer scheduling principles in network coding enabled wireless networks. PIE is proposed in Section 5.3. In Section 5.4, the performance of PIE is evaluated in terms of transmission efficiency and computational overhead through extensive simulations, followed by a summary in Section 5.5.

5.1 Network Model

We consider a network model similar to that in [73]. A remote Base Station (BS) broadcasts a batch of packets (blocks)[1] to nodes. Due to the fading and dynamics of cellular channels, each peer receives some (maybe all or none) of these blocks. To mitigate the congestion of downlinks from the BS to those nodes and release the bottleneck of the BS as a network gateway, the nodes can share their received blocks with each other through local wireless networks.

A local wireless network is comprised of several nodes which are also called peers in one-hop wireless scenario. These peers can communicate with each other directly through a commonly shared wireless channel in a half-duplex mode. In other words, if two peers are transmitting at the same time, their signals will interfere with each other, and no peer can correctly receive the signal. On the other hand, due to the broadcast nature of wireless channels, every other peer can receive the signal and recover the frames correctly when one and only one is transmitting.

Without loss of generality, we assume randomly combined packets sent by a peer are linearly independent to each other since the probability of linear dependence is very low [56]. Similarly, coded packets sent out from different peers are also assumed linearly independent to each other.

Table 5.1 gives the notations used in this chapter.

[1] The terms packet and block are used interchangeably in this chapter.

Table 5.1: List of Notations

Notation	Description
TRN_i	Total Receiving Number of Peer i
DD_i	Deficiency Degree of Peer i
TSN_i	Total Sending Number of Peer i
NUB_i	Number of Unique Blocks of Peer i
BDM (BDV)	Block Distribution Matrix (Vector)
BRM	Block Rareness Matrix
PDM	Peer Difference Matrix
PFV	Peer Freshness Vector
BAP_j	Benefit of All Peers from the j-th sending operation

5.2 Investigations on Information Exchange Principles

Since a specific solution to the peer scheduling problem depends on the original status of the block distribution among the peers, we represent the status as a Block Distribution Matrix (BDM). A BDM is a (0, 1)-matrix, also known as a binary matrix, in which each element is either one or zero. Row numbers and column numbers of a BDM represent peer indexes and block indexes, respectively. In other words, BDM(i, j) = 0 means that peer i does not have block j and BDM(i, j) = 1 means that peer i has block j. Based on a BDM, we summarize the following principles. The correlations between the principles and PIE are discussed in Subsection 5.3.2.

Definition 1: The total sending number (*TSN*) is defined as the total number of sending operations performed by all peers as a whole for the completion of the information exchange.

Proposition I. From the viewpoint of peers, a lower bound of *TSN* is the maximum value among all the sums of DD_i and NUB_i, i.e.,

$$TSN \geq \max_i \{DD_i + NUB_i\}, \tag{4.25}$$

where DD_i is the number of innovative packets that peer i needs to recover the whole information, and NUB_i denotes the number of the blocks which are uniquely owned by peer i.

Proof: From the viewpoint of peer i, the TSN for all peers is equal to the sum of TRN_i and TSN_i, i.e., $TSN = TRN_i + TSN_i$, where TRN_i and TSN_i are the numbers of packets that peer i receives and sends before the completion of information exchange, respectively. Obviously, we have $TRN_i \geq DD_i$ and $TSN_i \geq NUB_i$. Thus, we have $TSN \geq DD_i + NUB_i$. Because the inequality is true for all peers, we have Eq. (4.25). ■

Proposition II. From the viewpoint of blocks, a lower bound of TSN can be given as follows:

$$TSN \geq \left\lceil \left(\sum_{i=1}^{N} DD_i \right) \Big/ (N-1) \right\rceil, \tag{4.26}$$

where N is the number of peers ($N \geq 2$).

Proof: For the j-th sending operation, the benefit of all peers (BAP_j) is defined as a cumulative value of the benefits received by all peers. Thus, we have $BAP_j \leq N - 1$. On the other hand, each peer has all blocks after the completion of information sharing. Therefore, we have:

$$\sum_{j=1}^{TSN} BAP_j = \sum_{i=1}^{N} DD_i \,. \tag{4.27}$$

Thus, we have Eq. (4.26). ■

Corollary I. As a summary of *Proposition I* and *Proposition II*, a lower bound of TSN is:

$$LB_{TSN} = \max \left\{ \left\lceil \left(\sum_{i=1}^{N} DD_i \right) \Big/ (N-1) \right\rceil, \ \max_i \{DD_i + NUB_i\} \right\}. \tag{4.28}$$

Lemma I. In the above network model, for any peer i, incoming packets have no innovation to other peers, thus peer i has no necessity to code incoming packets into its future outgoing packets.

Proof: Without loss of generality, let an incoming packet be from peer j. In the above network model, all other peers can receive this packet, which thus has no innovation to those

peers any more. In addition, it is peer *j* that codes this packet, which is a linear combination of all packets peer *j* has and thus has no innovation to peer *j*. Therefore, for any peer *i*, the incoming packet has no innovation to any other peers including peer *j* and thus peer *i* has no necessity to include the incoming packet into its future outgoing packets. ■

Proposition III. In the above network model, sending sequences are order-independent.

Proof: From ***Lemma I***, for a given peer sending sequence, switching the orders of any two peers does not change the outcome. In other words, sending sequences are order-independent. ■

5.3 The Proposed PIE Scheme

Based on the peer scheduling principles, in this section, we propose a quasi-optimal but efficient and light-weight cooperative Peer-to-peer Information Exchange (PIE) scheme.

5.3.1 The PIE Scheme

The main idea of PIE is to take the freshness of peers into consideration in addition to the rarest first principle on blocks. The basic concept of freshness is a measurement on how much innovation a peer has against all other peers, which can be represented as follows:

$$PFV_i = \sum_j PDM_{ij} = \sum_j \sum_k 1_{\{BDV_{ik} > BDV_{jk}\}}, \tag{4.29}$$

where PFV_i denotes the freshness of peer *i*, PDM_{ij} denotes the difference of peer *i* against peer *j*, BDV_i is the block distribution vector of peer *i*, which is the *i*-th row vector of block distribution matrix (*BDM*) and so does BDV_j. The indicator function is defined as follows:

$$1_{\{BDV_{ik} > BDV_{jk}\}} = \begin{cases} 1, & if \quad BDV_{ik} > BDV_{jk}; \\ 0, & Otherwirse. \end{cases} \tag{4.30}$$

where BDV_{ik} is the *k*-th element of the vector BDV_i.

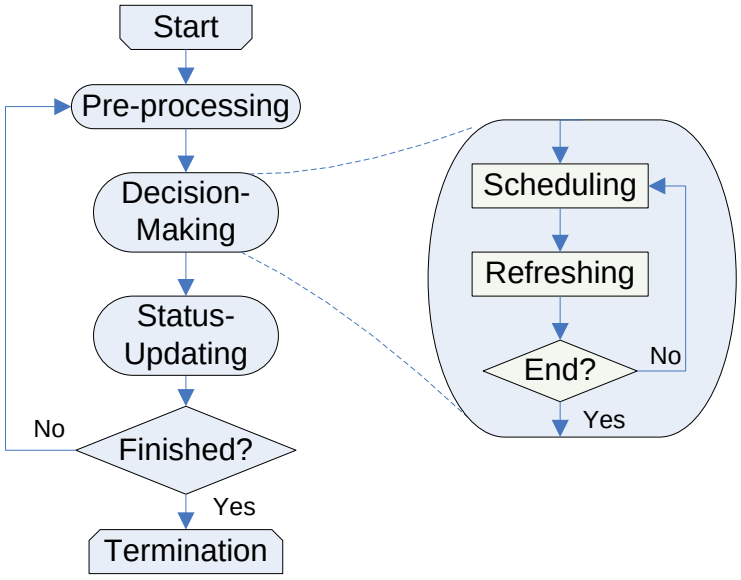

Figure 5.1: Flow Chart of PIE

As shown in Figure 5.1, PIE consists of four stages: pre-processing, decision-making, status-updating, and termination. The decision-making stage contains two modules with an algorithm in each module. The details of these stages and modules are depicted as follows.

Pre-processing: In PIE, peers first share BDVs with each other. The sharing of BDVs can be performed by each peer directly broadcasting BDVs to others through the shared side channel. Finally, each peer has the block distribution information of all other peers, which forms a BDM.

With the BDM, each peer can calculate the rareness of blocks and the freshness of peers, which are represented in a Block Rareness Matrix (BRM) and a Peer Freshness Vector (PFV), respectively. A BRM can be calculated as follows. We first calculate the rareness of each block; the rareness of a block denotes the number of peers that have this block; the less the value of the rareness of a block, the rarer the block. The block rareness information is reorganized and put into the BRM, where the row number denotes the rareness, the column number denotes the peer number, and the element value denotes the number of blocks of the rareness that a peer has. For example, BRM(i, j) = 3 means that peer j has 3 blocks of the

rareness *i*. PFV is calculated from PDM, as defined in Eq. (4.29). Another data structure is the deficiency degrees (*DD*) of all peers, which is used as the termination condition of the decision-making stage.

Algorithm 1: Peer Scheduling Algorithm

Input: *BRM*, *PFV*
Output: *next_sender*
RBPS ← *peers having the rarest blocks in BRM*
MRPS ← *peers having the most blocks in RBPS*
if |*MRPS*| = 1
  next_sender ← *the unique member of MRPS*
else
  next_sender ← *the peer in MRPS with largest freshness*
endif
return *next_sender*

Decision-Making: After pre-processing, each peer can start the decision-making stage, which consists of two modules; one is comprised of the peer scheduling algorithm, and the other the status refreshing algorithm.

The peer scheduling algorithm is described in Algorithm 1. First, we choose peers that own the rarest blocks and put them into a peer set with rarest blocks (RBPS). Then, peers with most blocks are chosen from the RBPS and put into another peer set (MRPS). The next sender is the unique peer in MRPS if it contains only one member; otherwise, the peer with the largest freshness is chosen as the next sender. The freshness values of peers are taken from PFV.

The status refreshing algorithm plays a crucial role in PIE since the refreshed status will affect the next round of peer scheduling. In Algorithm 2, *BRM*, *PDM*, *PFV*, and *DD* represent the information of system status from different aspects. *BRM* and *PFV* are for the next round of peer scheduling; *PDM* is for status refreshing; and *DD* is for the termination of the decision-making stage, where the termination condition is that *DD* equals a zero vector.

Algorithm 2: Status Refreshing Algorithm

Input: *BRM*, *PDM*, *PFV, DD, next_sender*
Output: *BRM, PDM, PFV, DD*
v_obj ← *a rarest block of the next_sender*
rare ← *the rareness of the block v_obj*
APS ← *peers having the block v_obj*
foreach *peer* **in** *APS*
  BRM(rare, peer) – –
endfor
foreach *peer* **in** *all peers*
  if *PDM(next_sender, peer)* > 0
    foreach *member* **in** *all peers*
      if *PDM(next_sender, member)* = 0
        PDM(member, peer) – –
        PFV(member) – –
      endif
    endfor
    DD(peer) – –
  endif
endfor
return *BRM, PDM, PFV, DD*

Notice that many data structures are used instead of a single BDM. The reason is that for network coding based information exchange, peers send out coded packets, which make it difficult to keep tracking the status of block distribution information using a single BDM. Finally, in the decision-making stage, PIE gives a peer scheduling sequence, which is generated through several rounds of peer scheduling and status refreshing based on the initially shared BDM.

Status-Updating: According to the peer scheduling sequence given in the decision-making stage, in this stage, peers send out one coded packet at each time without acknowledgement. Peers keep updating their own block distribution information with the

reception of new packets. If a packet is lost, a retransmission from the same peer is required to complete information exchange.

Termination: When each peer recovers all original blocks, the whole process is completed. If those peers have more information for exchange, they can repeat the above process.

5.3.2 Discussions

PIE is in line with our summarized principles. For the proof of ***Proposition I***, we have $TRN_i \geq DD_i$ and $TSN_i \geq NUB_i$. The former principle is observed by PIE, since DD_i is decreased by at most one in each round of scheduling and refreshing in Algorithm 2. The latter is also observed by PIE, since each unique block will make peer i stay in *RBPS*, resulting in that the transmission opportunities will never be scheduled to other peers with only larger-rareness blocks. In other words, from the viewpoint of blocks, before all peers which have unique blocks sends, *DD* will never equal a zero vector since the following equation holds:

$$\sum_{j=1}^{|NUB|} BAP_j = \sum_{j=1}^{|NUB|} (N-1) \leq |DD|, \tag{4.31}$$

where $|NUB|$ and $|DD|$ are the sums of all NUB_i's and all DD_i's, respectively. Thus, PIE is naturally in accordance with the ***Proposition I***. Moreover, according to Algorithm 2, we can see that the BAP_j is no larger than $N - 1$, making PIE conform to the ***Proposition II***. Finally, following *Proposition I* and *Proposition II*, the ***Corollary I*** naturally holds.

From Eq. (4.29), it can be seen that freshness is a cumulative difference of a peer against other peers. Thus, the concept of freshness represents a measurement of possible innovation a peer has against other peers. This definition captures the essence of network coding based information exchange in terms of innovative information, thus assisting to maximize the wireless coding gain.

5.4 Performance Evaluation

To verify the effectiveness and efficiency of PIE, we conduct extensive simulations for performance evaluation. In our simulation, each peer can successfully receive the original blocks from a BS with a prescribed probability. We define the probability as the sparsity degree of the original blocks. The performance of PIE is evaluated and also compared with the rarest first algorithm in terms of transmission efficiency and computational overhead.

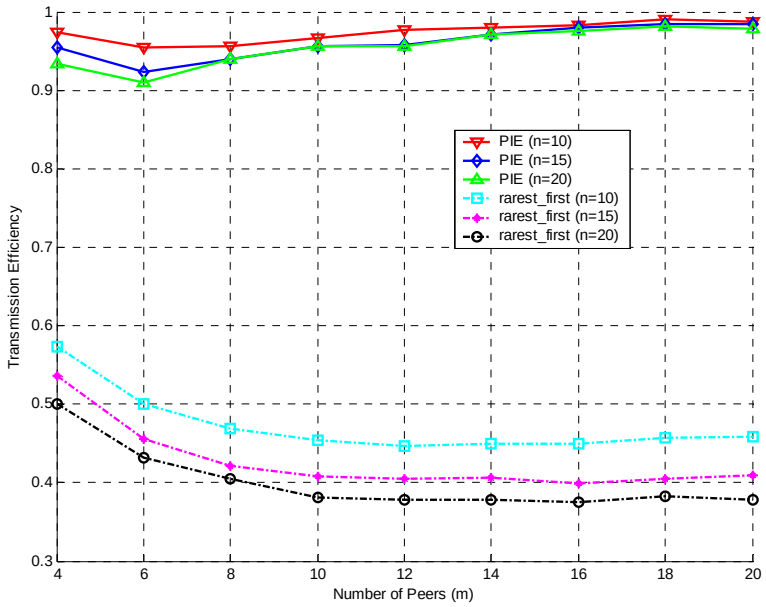

Figure 5.2: Efficiency vs. the Number of Peers (Sparsity = 0.8)

The theoretical lower bound of *TSN* in *Corollary I* is adopted as the benchmark for the evaluation of transmission efficiency, which is defined as:

$$E_t = \frac{TTA/TSN}{TTA/LB} = \frac{LB}{TSN}, \tag{4.32}$$

where E_t is transmission efficiency, TTA is the total transmission amount of information exchange, TSN is our simulation result, and LB is the theoretical lower bound of TSN, as defined in Eq. (4.28). From the definition, we know $E_t = LB/TSN \le OPT/TSN \le 1$ since

$LB \le OPT \le TSN$, where OPT denotes the optimal solution. Thus, we know PIE is near to the optimal in terms of transmission efficiency when E_t is near to 1.

5.4.1 Transmission Efficiency

Figure 5.2 shows the transmission efficiency versus the number of peers. It can be seen that the transmission efficiency of PIE is much higher, about 30% on average, than that of the rarest first algorithm. With the increase of the number of peers, the transmission efficiency of PIE increases and almost reaches its theoretical upper bound. Simulation results with different numbers of blocks (n = 10, 15, 20) are given for extensive verification.

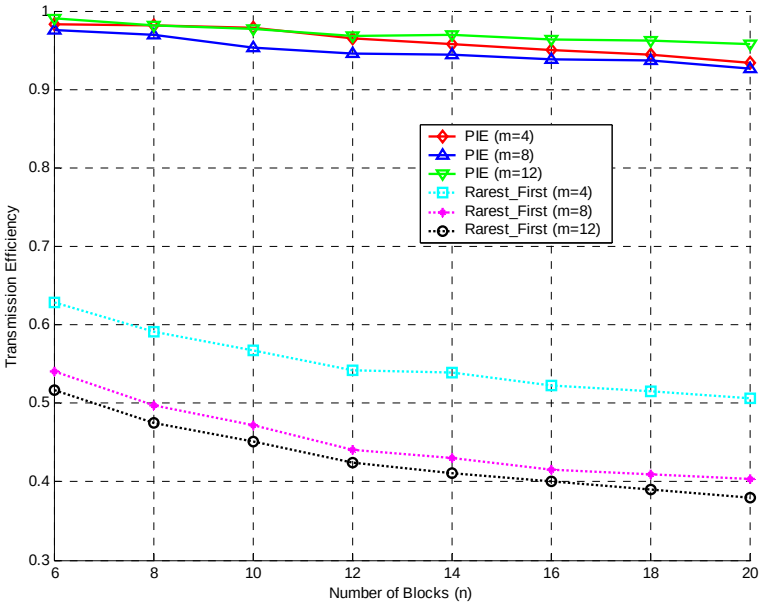

Figure 5.3: Efficiency vs. the Number of Blocks (Sparsity = 0.8)

The transmission efficiency versus the number of blocks is shown in Figure 5.3. It can be seen again that PIE outperforms the rarest first algorithm. With the increase of the number of blocks, the transmission efficiencies of both schemes decrease; while PIE still maintains more than 95% transmission efficiency in almost all scenarios. Simulation results with

different numbers of peers (m = 4, 8, and 12) are shown respectively for extensive verification. A more extensive comparison between PIE and the rarest first algorithm in terms of transmission efficiency is shown in Figure 5.4.

Figure 5.5 shows the transmission efficiency versus the sparsity degree with different numbers of peers (m = 5, 10, and 15) and different numbers of blocks (n = 5, 10, and 15). Both schemes have the almost same changing trend, while PIE outperforms the rarest first algorithm with extensively diverse sparsity degrees.

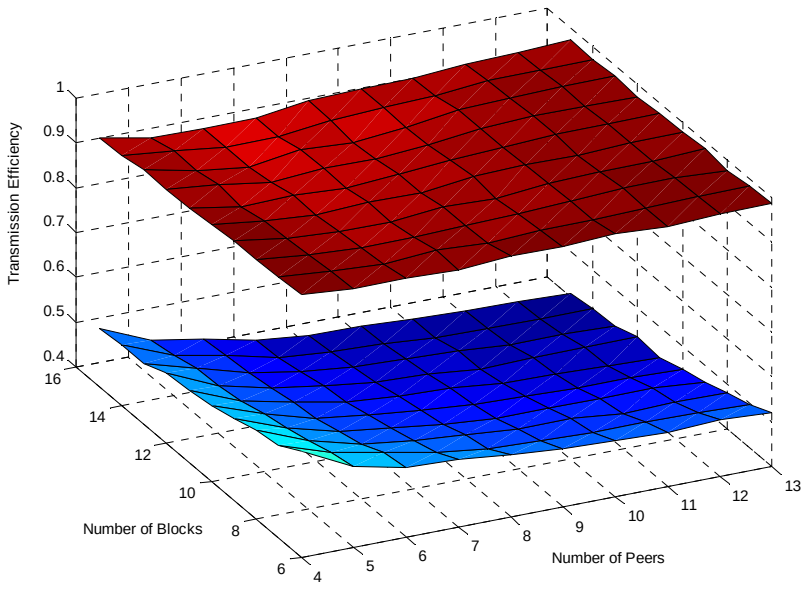

Figure 5.4: PIE vs. Rarest First (Sparsity = 0.8)

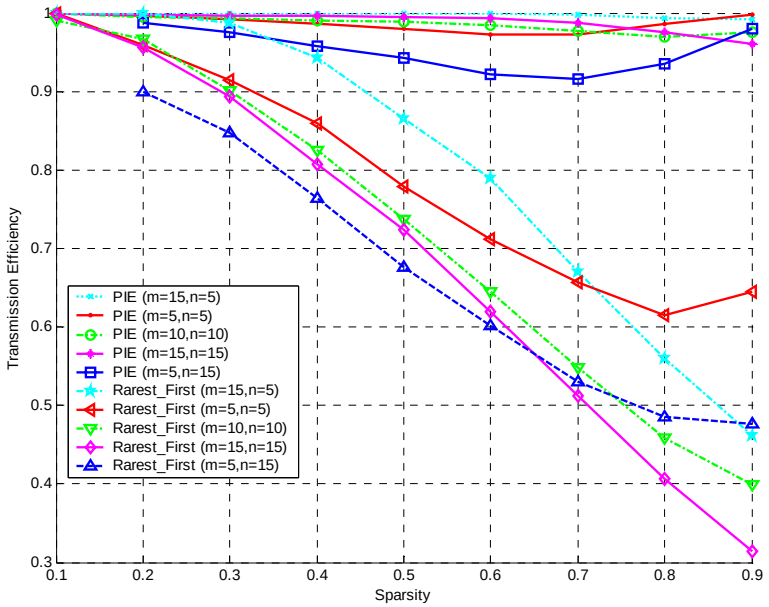

Figure 5.5: Efficiency vs. Sparsity

5.4.2 Computational Overhead

The computational overheads of PIE and the rarest first algorithm are shown in Figure 5.6, where all the computational overheads are collected from a laptop platform with a CPU of Intel Pentium M 1.8GHz and a RAM of 512MB. It can be seen that the computational overhead of the rarest first algorithm increases almost linearly with the sparsity degree, while that of PIE decreases almost linearly. Furthermore, the computational overhead of PIE still remains in the range of practical applications.

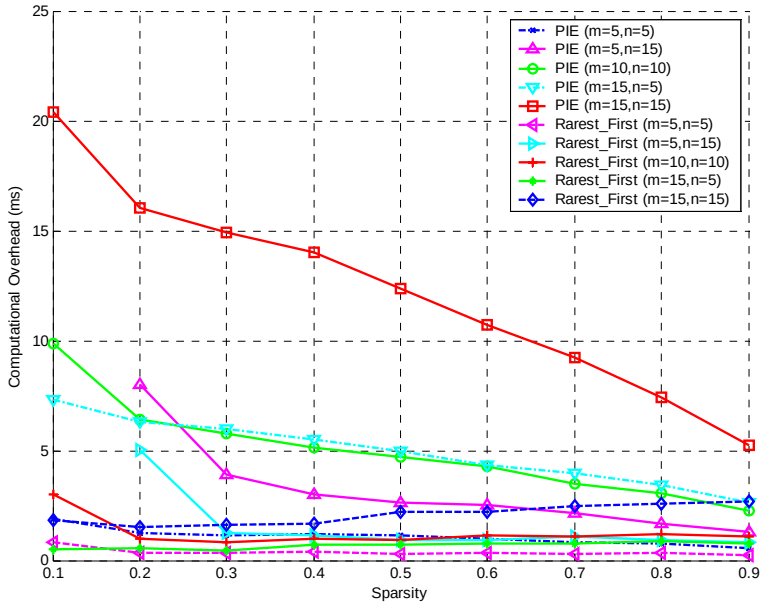

Figure 5.6: Computational Overhead vs. Sparsity (PIE and Rarest First)

5.5 Summary

In this chapter, we have proposed a cooperative Peer-to-peer Information Exchange (PIE) scheme with a compact, efficient, and light-weight peer scheduling algorithm for network coding enabled wireless networks. PIE can not only fully exploit the broadcast nature of wireless channels, but also take advantage of cooperative peer-to-peer information exchange. Qualitative analysis and extensive simulations have demonstrated the effectiveness and efficiency of PIE.

Chapter 6
Conclusions and Future Work

In this chapter, the contributions of this thesis are concluded, followed by the future work.

6.1 Conclusions

The major contributions of this thesis can be summarized as follows. We studied the characteristics of MWNs and pointed out four research challenges on the information security for MWNs. To prevent the identified security threats and protect the information security of MWNs, we proposed three research topics focusing on the above-mentioned research challenges.

In MWNs, traffic analysis and flow tracing attacks are very challenging security threats, and these attacks may be utilized by adversaries to compromise the privacy of users, such as the source anonymity. To prevent traffic analysis and flow tracing attacks in MWNs, we proposed a network coding based privacy-preserving scheme to achieve source anonymity. To the best of our knowledge, this is the first research on using network coding to achieve privacy preservation. Asissted with HEFs, the proposed scheme utilize the mixing feature of network coding to confuse the adversaries. This scheme can offer two significant privacy-preserving features, packet flow untraceability and message content confidentiality; these two privacy features can together help to achieve the privacy objective, source anonymity.

A stronger privacy objective, source unobservability, can be achieved if a scheme which can provide source anonymity is combined with dummy messages. However, dummy messages may bring the explosion of network traffic, which may further cause severe performance degradation or even service denial. Existing schemes rely on trusted proxies to filter dummy messages and thus to avoid traffic explosion. However, in MWNs, trusted proxies suffer from compromising attacks and may be compromised. It is arguable that it is impossible to thwart those internal attackers and prevent traffic explosion at the same time.

We investigated this challenging issue from a different viewpoint and utilized network coding to absorb those dummy messages instead of filtering them. To prevent traffic explosion for ensuring system availability while achieving the privacy objective of source unobservability at the same time, we proposed a privacy-preserving scheme based on network coding and dummy messages for MWNs. Dummy messages utilized in the proposed scheme are specially designed; however, adversaries cannot distinguish them from real ones since they always appears in their ciphertext format. Network coding is utilized in this scheme to automatically absorb these dummy messages, and thus traffic explosion induced DoS can be naturally avoided to ensure the system availability.

Other than the system availability, we also explored a data availability issue. We defined an information exchange issue when remote base stations are temporarily unavailable or intermittent. We referred to this issue as the *peer scheduling* issue, and the optimal peer scheduling solution is proven NP-hard. To enhance the data availability among peers when the remote base station is temporarily unavailable or intermittent, we propose a cooperative peer-to-peer information exchange scheme. Network coding is utilized in the scheme and peers are scheduled to send network-coded packets to cooperatively share the information with each other. The proposed scheme can achieve a quasi-optimal peer-to-peer information exchange in terms of throughput and transmission delay.

6.2 Future Work

Our research has made significant progress in the information security of MWNs. Yet this is still a very wide open field. There are several research directions to be explored to complement our efforts.

6.2.1 Privacy Preservation for DTNs

Delay/Disruption Tolerant Networks (DTNs) [77] are very hot research topic in recent years. DTNs can be regarded as a kind of MWNs; however, DTNs are also characterized by their

unique features such as lack of end-to-end connection, fragmentation, and bundle accumulation. These unique features pose new challenges to the information security, especially privacy preservation, of DTNs.

Privacy preservation is a new research topic in DTNs and has received little attention. The existing privacy-preserving technologies such as mix-net and onion routing are not suitable for DTNs. Considering the unique characteristics of DTNs, we will carefully examine the existing privacy-preserving schemes and design new privacy-preserving schemes for DTNs.

6.2.2 Enhancing Data Availability for DTNs

As a kind of MWNs, DTNs characterize themselves with a series of unique features such as lack of end-to-end connection, fragmentation, and bundle accumulation. These three unique features pose great challenges to the information security of DTNs. For example, the lack of end-to-end connection and fragmentation features will severely degrade the data availability in DTNs, which is one of our future research topics.

A good bundle accumulation mechanism may greatly enhance the data availability in DTNs. However, a single bundle accumulation mechanism may not able to achieve the desired data availability level. In this sense, a whole-set design of DTNs may be preferred for the enhancement of data availability. On the other hand, network coding has many desired features such as block scheduling easiness and multiple data delivery, which can be utilized for enhancing data availability in DTNs. We will consider including network coding into our future design.

Bibliography

[1] D. Howard, "It's a Wi-Fi World", *ACM NetWorker*, vol. 6, no. 3, pp. 26-30, 2002.

[2] B. Li, Y. Qin, C. P. Low, and C. L. Gwee, "A Survey on Mobile WiMAX [Wireless Broadband Access]", *IEEE Communications Magazine*, vol. 45, no. 12, pp. 70-75, 2007.

[3] M. Sukor, S. Ariffin, N. Fisal, S.K.S. Yusof, A. Abdallah, "Performance Study of Wireless Body Area Network in Medical Environment", *Proc. AICMS'08*, pp. 202-206, 2008.

[4] P. Marino, F.P. Fontan, and M.A. Dommguez, "Environmental Monitoring Based on Emerging Wireless Technologies", *Proc. ICNS'08*, pp. 30-34, 2008.

[5] D. Schwab and R. Bunt, "Characterising the use of a campus wireless network", *Proc. IEEE INFOCOM'04*, vol. 2, pp. 862-870, 2004.

[6] A.H. Lashkari, M.M.S. Danesh, and B. Samadi, "A Survey on Wireless Security Protocols (WEP, WPA and WPA2/802.11i)", *Proc. IEEE ICCSIT'09*, pp. 48-52, 2009.

[7] H.-Y. Hsieh and R. Sivakumar, "IEEE 802.11 over multi-hop wireless networks: problems and new perspectives", *Proc. IEEE VTC'02*, vol. 2, pp. 748-752, 2002.

[8] P. Gajbhiye and A. Mahajan, "A Survey of Architecture and Node Deployment in Wireless Sensor Network", *Proc. ICADIWT'08*, pp. 426–430, 2008.

[9] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing Source-Location Privacy in Sensor Network Routing", *Proc. IEEE ICDCS'05*, pp. 599-608, 2005.

[10] S. Ding, "A Survey on Integrating MANETs with the Internet: Challenges and designs", *Computer Communications*, vol. 31, no. 14, pp. 3537-3551, 2008.

[11] X. Lin, R. Lu, H. Zhu, P.-H. Ho, X. Shen and Z. Cao, "ASRPAKE: An Anonymous Secure Routing Protocol with Authenticated Key Exchange for Wireless Ad Hoc Networks", *Proc. IEEE ICC'07*, 2007.

[12] I.F. Akyildiz, X. Wang, "A Survey on Wireless Mesh Networks", *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23-S30, 2005.

[13] J.-C. Kuo, W. Liao, and T.-C. Hou, "Impact of Node Density on Throughput and Delay Scaling in Multi-hop Wireless Networks", *IEEE Transactions on Wireless Communications*, vol. 8, no. 10, pp. 5103-5111, 2009.

[14] B. Xie, A. Kumar, D. Cavalcanti, D.P. Agrawal, and S. Srinivasan, "Mobility and routing management for heterogeneous multi-hop wireless networks", *Proc. IEEE MAHSS'05*, pp. 1-7, 2005.

[15] R. Vaze and R.W. Heath, "Maximizing reliability in multi-hop wireless networks", *Proc. IEEE ISIT'08*, pp. 11-15, 2008.

[16] A.M. Eskicioglu and E.J. Delp, "A Key Transport Protocol based on Secret Sharing Applications to Information Security", *IEEE Transactions on Consumer Electronics*, vol. 48, no. 4, pp. 816-824, 2002.

[17] R.S. Fussell, "Protecting Information Security Availability via Self-adapting Intelligent Agents", *Proc. IEEE MILCOM'05*, vol. 5, pp. 2977-2982, 2005.

[18] R. Curtmola and C. Nita-Rotaru, "BSMR: Byzantine-Resilient Secure Multicast Routing in Multihop Wireless Networks", *IEEE Transactions on Mobile Computing*, vol. 8, no. 4, pp. 445-459, 2009.

[19] J.C. Kao, and R. Marculescu, "Minimizing Eavesdropping Risk by Transmission Power Control in Multihop Wireless Networks ", *IEEE Transactions on Computers*, vol. 56, no. 8, pp. 1009-1023, 2007.

[20] S. Bashar and Z. Ding, "Optimum routing protection against cumulative eavesdropping in multihop wireless networks", *IEEE MILCOM'09*, pp. 1-7, 2009.

[21] S. Islam, A. Hamid, C.S. Hong, B.-H. Chang, "Preserving Identity Privacy in Wireless Mesh Networks", *Proc. ICOIN'08*, pp. 1-5, 2008.

[22] M. Singh and A. Saxena, "Secure Computation for Data Privacy", *Proc. SECCOM'07*, pp. 58-62, 2007.

[23] M. Decker, "Location Privacy-An Overview ", *Proc. ICMB'08*, pp. 221-230, 2008.

[24] J. Wang, Y. Zhao, S. Jiang, and J. Le, "Providing Privacy Preserving in Cloud Computing", *Proc. ICTM'09*, vol. 2, pp. 213-216, 2009.

[25] Y. Wei, Y. He, and L. Hao, "An Identity Privacy Enhanced Trust Model in Fully Distributed Virtual Computing Environments", *Proc. NSWCTC '09*, vol. 1, pp. 704-708, 2009.

[26] A. Pfitzmann and M. Hansen, "Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology", v0.31, Feb. 15, 2008.

[27] A.C. Weaver, "Achieving Data Privacy and Security Using Web Services", *Proc. ICIT'05*, pp. 2-5, 2005.

[28] I. Maglogiannis, L. Kazatzopoulos, K. Delakouridis, and S. Hadjiefthymiades, "Enabling Location Privacy and Medical Data Encryption in Patient Telemonitoring Systems", *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 946-954, 2009.

[29] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: Robust Location Privacy Scheme for VANET", *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1569-1589, 2007.

[30] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures", *Proc. DASC'09*, pp. 711-716, 2009.

[31] D.M. Gregg, W.J. Blackert, D.V. Heinbuch, and D. Furnanage, "Assessing and Quantifying Denial of Service Attacks", *Proc. MILCOM'01*, vol. 1, pp. 76-80, 2001.

[32] K.S. Lakshamanan and R. Rajkumar, "RETROFIT: Reliable Exchanges through Resilient Overlays for Internet Teleoperation", *Proc. ICDCS'08*, pp. 477-482, 2008.

[33] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "Correlation-Based Traffic Analysis Attacks on Anonymity Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. pp, no. 99, pp. 1-14, 2009.

[34] A. George, A. Kumar, and A. Desoky, "An Analytical Model for the Performance Evaluation of Multi-hop Wireless Networks", *Proc. IEEE ISCC'07*, pp. 403-408, 2007.

[35] J.-Q. Jin, T. Ho, and H. Viswanathan, "Comparison of Network Coding and Non-Network Coding Schemes for Multi-hop Wireless Networks", Proc. IEEE ISIT'06, pp. 197-201, 2006.

[36] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast", *IEEE Trans. on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[37] P. A. Chou and Y. Wu. "Network Coding for the Internet and Wireless Networks", *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77–85, Sept. 2007.

[38] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," *Proc. of 51st Allerton Conf. Communication, Control and Computing*, Oct. 2003.

[39] J. Benaloh, "Dense Probabilistic Encryption", *Proc. of the Workshop on Selected Areas in Cryptography*, pp. 120–128, 1994.

[40] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuocity Classes", *Proc. of EUROCRYPT'99*, LNCS, vol. 1592, pp. 223–238, 1999.

[41] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards Statistically Strong Source Anonymity for Sensor Networks", *Proc. IEEE INFOCOM'08*, pp. 51–55, 2008.

[42] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web Transactions", *ACM Trans. on Information and System Security*, vol. 1, no. 1, pp. 66–92, Nov. 1998.

[43] C. Shields and B. N. Levine, "A Protocol for Anonymous Communication over the Internet", *Proc. of ACM CCS'00*, pp. 33–42, 2000.

[44] M. Rennhard and B. Plattner, "Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection", *Proc. of the ACM Workshop on Privacy in the Electronic Society*, pp. 91–102, 2002.

[45] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol", *Proc. of the 2003 IEEE Symposium on Security and Privacy*, pp. 2–15, May 2003.

[46] D. Goldschlag, M. Reed, and P. Syverson, "Onion Routing for Anonymous and Private Internet Connections", *Communications of the ACM*, Vol. 42, No. 2, pp. 39–41, Feb. 1999.

[47] X. Wu and N. Li, "Achieving Privacy in Mesh Networks", *Proc. of the fourth ACM workshop on Security of Ad hoc and Sensor Networks (SASN'06)*, pp. 13–22, 2006.

[48] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", *IEEE Trans. on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[49] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial Time Algorithms for Network Information Flow", *Proc. of 15th ACM symposium on Parallel Algorithms and Architectures* (SPAA'03), pp. 286–294, 2003.

[50] M. Wang and B. Li, "Network Coding in Live Peer-to-Peer Streaming", *IEEE Trans. on Multimedia*, Vol. 9, No. 8, pp. 1554–1567, 2007.

[51] E. Ayday, F. Delgosha, and F. Fekri, "Location-Aware Security Services for Wireless Sensor Networks Using Network Coding", *Proc. IEEE INFOCOM'07*, pp. 1226–1234, 2007.

[52] K. Han, T. Ho, R. Koetter, M. Medard, and F. Zhao, "On Network Coding for Security", *Proc. IEEE MILCOM'07*, pp. 1–6, 2007.

[53] Z. Li, B. Li, and L.C. Lau, "On Achieving Maximum Multicast Throughput in Undirected Networks", *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2467–2485, Jun. 2006.

[54] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-Energy Multicast in Mobile Ad Hoc Networks using Network Coding", *IEEE Trans. on Communications*, vol. 53, no. 11, pp. 1906–1918, Nov. 2005.

[55] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," *Proc. IEEE INFOCOM'09*, Rio de Janeiro, Brazil, April 19–25, 2009.

[56] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An Efficient Privacy-Preserving Scheme against Traffic Analysis Attacks in Network Coding", *Proc. IEEE INFOCOM'09*, Rio de Janeiro, Brazil, April 19–25, 2009.

[57] L. Lima, J.P. Vilela, J. Barros, and M. Medard, "An Information-Theoretic Cryptanalysis of Network Coding - Is Protecting the Code Enough?", *Proc. of ISITA'08*, pp. 1–6, 2008.

[58] P. Venkitasubramaniam and L. Tong, "Anonymous Networking with Minimum Latency in Multihop Networks", *Proc. IEEE Symposium on Security and Privacy*, pp. 18–32, 2008.

[59] Y. Challal and H. Seba, "Group Key Management Protocols: a novel taxonomy," *International Journal of Information Technology,* vol. 2, pp. 105–119, 2005.

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

[61] C. Gkantsidis and P. R. Rodriguez, "Cooperative Security for Network Coding File Distribution", *Proc. IEEE INFOCOM'06*, pp. 1–13, 2006.

[62] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, "Sage: A Strong Privacy-Preserving Scheme against Global Eavesdropping for E-health Systems", *IEEE Journal on Selected Areas in Communications*, Vol. 27, No. 4, pp. 365-378, 2009.

[63] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards Event Source Unobservability with Minimum Network Traffic in Sensor Networks", *Proceedings of ACM WiSec'08*, pp. 77-88, 2008.

[64] Y. Wu, "On Constructive Multi-Source Network Coding", *Proc. IEEE International Symposium on Information Theory*, pp. 1349-1353, 2006.

[65] C.A. Jotten, C. Sgraja, and J.J. Blanz, "On the Impact of Coarse Synchronization on the Performance of Broadcast/Multicast Single Frequency Network Operation in WCDMA", *Proc. IEEE 68th Vehicular Technology Conference*, pp. 1-6, 2008.

[66] K. Mehta, D. Liu, and M. Wright, "Location Privacy in Sensor Networks Against a Global Eavesdropper", *Proc. IEEE ICNP'07*, pp. 314-323, 2007.

[67] S. Zhu, S. Setia, and S. Jajodia, "Leap: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks", *ACM CCS*, 2003.

[68] C. Gkantsidis and P.R. Rodriguez, "Network Coding for Large Scale Content Distribution", *Proc. IEEE INFOCOM'05*, vol. 4, pp. 2235-2245, Mar. 2005.

[69] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth Codes: Maximizing Sensor Network Data Persistence", *Proc. ACM SIGCOMM'06*, pp. 255-266, 2006.

[70] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding", *IEEE/ACM Trans. Networking*, vol. 16, no. 3, pp. 497-510, 2008.

[71] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient Broadcasting Using Network Coding", *IEEE/ACM Trans. Networking*, vol. 16, no. 2, pp. 450-463, Apr. 2008.

[72] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless Broadcast Using Network Coding", *IEEE Trans. on Vehicular Technology*, vol. 58, no. 2, pp. 914-925, Feb. 2009.

[73] X. Liu, S. Raza, C.-N. Chuah, and G. Cheung, "Network Coding Based Cooperative Peer-to-Peer Repair in Wireless Ad-Hoc Networks", *Proc. IEEE ICC'08*, pp. 2153-2158, May 2008.

[74] H. Chi, Q. Zhang, J. Jia, and X. Shen, "Efficient Search and Scheduling in P2P-based Media-on-Demand Streaming Service", *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 119-130, Jan. 2007.

[75] F. Lu, L.-T. Chia, and K. L. Tay, "NBgossip - Neighborhood Gossip with Network Coding Based Message Aggregation", *Proc. IEEE MASS'07*, pp. 1-12, Oct. 2007.

[76] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest First and Choke Algorithms are Enough", *Proc. the 6th ACM SIGCOMM conference on Internet measurement, IMC'06*, pp. 203-216, 2006.

[77] Z. Zhang, "Routing in Intermittently Connected Mobile Ad hoc Networks and Delay Tolerant Networks: Overview and Challenges", *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24-37, 2006.

[78] M. Wang and B. Li, "R^2: Random Push with Random Network Coding in Live Peer-to-Peer Streaming", *IEEE Journal on Selected Areas in Communications*, Special Issue on Advances in Peer-to-Peer Streaming Systems, vol. 25, no. 9, pp. 1655-1666, December 2007.

[79] C. Wu, B. Li, and S. Zhao, "Characterizing Peer-to-Peer Streaming Flows", *IEEE Journal on Selected Areas in Communications*, Special Issue on Advances in Peer-to-Peer Streaming Systems, vol. 25, no. 9, pp. 1612-1626, December 2007.

[80] C. Wu and B. Li, "rStream: Resilient and Optimal Peer-to-Peer Streaming with Rateless Codes", *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 77-92, January 2008.