WORMHOLE ATTACK DETECTION ALGORITHMS IN WIRELESS NETWORK CODING SYSTEMS

By
SHIYU JI
Bachelor of Engineering in Information Security
Harbin Institute of Technology
Weihai, Shandong, China
2012

Submitted to the Faculty of the
Graduate College of the
Oklahoma State Univeristy
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2015

WORMHOLE ATTACK DETECTION ALGORITHMS IN WIRELESS NETWORK CODING SYSTEMS

Thesis Approved:

Dr. Tingting Chen

Thesis Adviser

Dr. Eric Chan-Tin

Dr. David Cline

Name: SHIYU JI

Date of Degree: July, 2015

Title of Study: WORMHOLE ATTACK DETECTION ALGORITHMS IN WIRELESS NETWORK CODING SYSTEMS

Major Field: Computer Science

Abstract:

Network coding has been shown to be an effective approach to improve the wireless system performance. However, many security issues impede its wide deployment in practice. Besides the well-studied pollution attacks, there is another severe threat, that of wormhole attacks, which undermines the performance gain of network coding. Since the underlying characteristics of network coding systems are distinctly different from traditional wireless networks, the impact of wormhole attacks and countermeasures are generally unknown. In this thesis, we quantify wormholes' devastating harmful impact on network coding system performance through experiments. Firstly, we propose a centralized algorithm to detect wormholes and show its correctness rigorously. For the distributed wireless network, we propose DAWN, a Distributed detection Algorithm against Wormhole in wireless Network coding systems, by exploring the change of the flow directions of the innovative packets caused by wormholes. We rigorously prove that DAWN guarantees a good lower bound of successful detection rate. We perform analysis on the resistance of DAWN against collusion attacks. We find that the robustness depends on the node density in the network, and prove a necessary condition to achieve collusion-resistance. DAWN does not rely on any location information, global synchronization assumptions or special hardware/middleware. It is only based on the local information that can be obtained from regular network coding protocols, and thus the overhead of our algorithms is tolerable. Extensive experimental results have verified the effectiveness and the efficiency of DAWN.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In the efforts to improve the system performance of wireless networks, network coding has been shown to be an effective and promising approach (e.g., [1, 2, 3, 4, 5]) and it constitutes a fundamentally different approach compared to traditional networks, where intermediate nodes store and forward packets as the original. In contrast, in wireless network coding systems, the forwarders are allowed to apply encoding schemes on what they receive, and thus they create and transmit new packets. The idea of mixing packets on each node takes good advantages of the opportunity diversity and broadcast nature of wireless communications, and significantly enhances system performance.

However, practical wireless network coding systems face new challenges and attacks, whose impact and countermeasures are still not well understood because their underlying characteristics are different from well-studied traditional wireless networks. The wormhole attack is one of these attacks. In a wormhole attack, the attacker can forward each packet using wormhole links and without modifies the packet transmission by routing it to an unauthorized remote node. Hence, receiving the rebroadcast packets by the attackers, some nodes will have the illusion that they are close to the attacker. With the ability of changing network topologies and bypassing packets for further manipulation, wormhole attackers pose a severe threat to many functions in the network, such as routing and localization [6, 7, 8, 9, 10]. To investigate wormhole attacks in wireless network coding systems, we focus on their impact and countermeasures in a class of popular network coding scheme - the random linear network coding (RLNC) system [2]. In this system, in order to best utilize resources, before data transmissions, routing decisions (i.e., how many times of transmissions a forwarder should make for each novel packet) are made based on local link conditions by some test transmissions.

Since in wireless network coding systems the routing and packet forwarding procedures are different from those in traditional wireless networks, the first question that we need to answer is: Will wormhole attacks cause serious interruptions to network functions and downgrade system performance? Actually no matter what procedures are used, wormhole attacks severely imperil network coding protocols. In particular, if wormhole attacks are launched in routing, the nodes close to attackers will receive more packets than they should and be considered as having a good capability in help forwarding packets. Thus they will be assigned with more responsibility in packet forwarding than what they can actually provide. Furthermore, other nodes will be correspondingly contributing less. This unfair distribution of workload will result in an inefficient resource utilization and reduce system performance. Wormhole attacks launched during the data transmission phase can also be very harmful. First, wormhole attacks can be used as the first step towards more sophisticated attacks, such as man-in-the-middle attacks and entropy attacks [11]. For example, by retransmitting the packets from the wormhole links, some victim nodes will have to process much more non-innovative packets that will waste their resources; these constitute entropy attacks. Secondly, the attackers can periodically turn on and off the wormhole links in data transmissions, confusing the system with fake link condition changes and making it unnecessarily rerun the routing process. To further quantify the impact of wormhole attacks in wireless network coding systems, we perform extensive experiments and investigate the results in Chapter 3.

The main objective of this paper is to detect and localize wormhole attacks in wireless network coding systems. The major differences in routing and packet forwarding rule out using existing countermeasures in traditional networks [12, 13, 9, 10, 14, 6, 8, 7, 10, 15, 16]. In network coding systems like MORE[5], the connectivity in the network is described using the link loss probability value between each pair of nodes, while traditional networks use connectivity graphs with a binary relation (i.e., connected or not) on the set of nodes. For this reason, prior works based on graph analysis [10, 6, 14, 8] cannot be applied. Some other existing works rely on the packet round trip time difference introduced by wormhole attacks to detect them [13, 15, 16]. Unfortunately, this type of solutions cannot work with network coding either. They require either to use an established route that does not exist with network coding, or to calculate the delay between every two neighboring nodes which will introduce a huge amount of error in network coding systems.

In this thesis, we first propose a centralized algorithm to detect wormholes leveraging a central node in the network. For the distributed scenarios, we propose a distributed algorithm, DAWN, to detect wormhole attacks in wireless intra-flow network coding systems. The main idea of our solutions is that we examine the order of the nodes to receive the innovative packets in the network, and explore its relation with a widely used metric, Expected Transmission Count (ETX), associated with each node [17, 5]. Our algorithms do not rely on any location information, global synchronization assumption or special hardware/middleware. Our solutions only depend on the local information that can be obtained from regular network coding protocols, and thus the overhead that our algorithms introduce is acceptable for most applications.

Different wireless networks have different characteristics and requirements. Some wireless networks have central controller, while others are highly distributed without any centralized authority. It is desirable to apply different solutions based on the network types. Our centralized algorithm is inspired by the fact that the wormhole link can significantly change the network topology, which can be measured by ETX. This idea is also heuristic to our distributed solution DAWN, which emphasizes on the scenario where no central administration node exists. Thus, our algorithms can address different scenarios. We first present the centralized solution and then discuss the distributed one, for a clear logic flow. On the other hand, compared with our distributed algorithm DAWN, our centralized algorithm also owns several advantages. The centralized algorithm concentrates the computation workload to the central node, and thus each normal node will suffer much less workload than DAWN. Since the transmissions between each node and the central node are unicast, the caused communication overheads of the centralized algorithm are lower than DAWN, which broadcasts the reports. The centralized algorithm leverages the global information of the flows, and thus it can detect the wormhole link efficiently, and the resulted warnings can be delivered to each node more quickly than DAWN.

1.1 Contribution Summarization

We summarize the contributions of this thesis as follows.

- We are the first to study the impact and countermeasures of wormhole attacks in wireless network coding systems.

- We investigate the harmful impact of wormholes on system performance and regional nodes' resource utilization. We demonstrate the results via simulations on various scenarios.

- We propose a centralized algorithm to detect wormholes. In this algorithm, a central node collects the information from all the nodes in the network and analyzes whether there exists a wormhole link. The algorithm leverages the order of the nodes to receive the innovative packet, and utilizes machine learning techniques to distinguish the wormhole cases. We also give rigorous analysis of the centralized algorithm and find the condition of its effectiveness.

- For distributed network without centralized authority, we propose DAWN, a Distributed detection Algorithm against Wormhole in wireless Network coding systems. In DAWN, during regular data transmissions, each node records the abnormal arrival of innovative packets and share this information with its neighbors. This algorithm is efficient and practical without strong assumptions. Furthermore, we theoretically prove that DAWN guarantees a good lower bound of successful detection rate.

- We perform analysis on the resistance of DAWN against collusion attacks. We find that the robustness depends on the node density in the network, and prove a necessary condition to achieve collusion-resistance.

- We use extensive experiments in various network settings, to verify that DAWN is effective (with over 89.43% detection rate), and efficient.

1.2 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 will introduce related technical preliminaries. Then we will demonstrate the detrimental influences of wormhole attack in Chapter 3. Chapter 4 will explain how to determine the ETX of each node, and Chapter 5 will propose a centralized algorithm to detect the wormhole attack. In Chapter 6, we will describe our wormhole attack detection algorithm, and we will show the effectiveness and robustness of our solutions. Our experiments and the related analysis will be discussed in Chapter 7. After the Related Work Chapter 8 we will conclude this paper in Chapter 9.

Chapter 2

Technical Preliminaries

In this chapter, we describe the technical preliminaries needed in this paper.

2.1 Random Linear Network Coding (RLNC)

Linear Network Coding (LNC), especially Random Linear Network Coding (RLNC), owns numerous applications [18, 19, 20]. Linear network coding permits each node in the network to pass on the combinations of the received data, in order to optimize the information capacity. Let $r_1, r_2, \cdots, r_n$ denote the received data, and s be the encoded data to be passed to another node. We can obtain the combination f based on the received data based on Equation (2.1).

$$s = f(r_1, r_2, \cdots, r_n) \tag{2.1}$$

For RLNC, f in Equation (2.1) is a random linear combination in the field $GF(2^k)$.

$$f(r_1, r_2, \cdots, r_n) = \sum_{i=1}^{n} \xi_i r_i \tag{2.2}$$

Here, ξ_i is a randomly generated coefficient.

In network coding, every node except the recipient applies a random linear mapping from the inputs to outputs over the field $GF(2^k)$. Each packet contains a vector in the m-dimensional code vector space V. Particularly, each packet sent by the source node contains a basis of the code vector space V. If one intermediate node receives a packet which is linearly independent from previous packets, this packet is called an *innovative* packet. Essentially, an innovative packet must contain at least one basis that the node has not received, and the arrival of an innovative packet will increase the *rank* of the received packets by one. When the destination receives m innovative packets, whose vectors are linearly independent from each other, it can restore the source information S based on the received data R.

$$S = C^{-1}R \tag{2.3}$$

Here C is the matrix of the coefficients of the received packets. Since each received packet is essentially a linear combination of the original packets from the source, we can perfectly restore the original messages by multiplying the inverse of C. The capacity of RLNC converges to the optimum in probability [2], and owns an ideal performance on the compression of the transmitted data. However, since the packet can derive various forms during the transmissions in network coding, when the wormhole attack is initiated, it is difficult to apply some traditional solutions (i.e. tracing the timestamps of a particular packet) to defend. Thus, the wide applications of network coding push us to find another way to defend against wormhole attack.

2.2 Expected transmission count (ETX)

ETX has extensive applications in network coding systems [5, 3, 4, 21]. In this paper, the ETX of a node u in the network coding system denotes the expected total number of transmissions (including retransmissions) that the source node should make, in order to make the node u receive one innovative packet successfully. A node of high ETX means it is difficult to make it heard from the source, usually because the node is far from the source and the links between them are very lossy. Thus, the metric of the ETXs is a good representation of the network structure.

In existing works (e.g., [5, 17]), the ETXs are calculated based on the probabilities of packet loss between each pair of the nodes in the network. Let u and v be two nodes, and $p(u, v)$ be the probability of successful transmission between nodes u and v. For the simplest case, if the network only has a sender u and a recipient v, then the ETX of the sender u is 1.0, and the ETX of v is shown as Equation (2.4).

$$ETX(v) = \frac{1}{p(u, v)} \tag{2.4}$$

The probability $p(u, v)$ can be estimated based on the previous transmission record, using some statistical models like weighted means and window-based observation[5]. Based on (2.4), if the link between the nodes is very lossy, the ETX of v can be very high, indicating that it is difficult to deliver messages through the link. Usually too many hops might contribute to the high link loss probability. As we will talk about, the wormhole link connects two distant nodes with one hop of very low loss probability, and thus reduces the ETXs significantly. This fact is heuristic to our algorithms.

2.3 RLNC with ETX

In this paper, we consider a wireless network with a set of homogeneous nodes running network coding protocols (including routing protocols like [5] to calculate the number of per-packet transmissions for each node, and data transmission protocols). Nodes are connected via lossy wireless links. For any two nodes u and v in the network such that the successful transmission rate between u and v, $p(u, v) > 0$, then we say u and v are neighbors. We assume that ETXs are calculated to describe the network topology, and are measured periodically to support routing functions. Each node knows its own ETXs and its neighbors' ETXs.

In the wireless network systems, we consider that Public Key Infrastructure (PKI) is in place to implement the public key cryptographic techniques. For the wireless network, we regard each node[1] as a user who has a pair of public and private keys. The identity and the public key of each user are managed by the Certificate Authority (CA), which is a trusted entity. If any node A wants to safely communicate with node B, A has to request B's public key from the CA firstly. After the transmission, node B has to request A's public key from the CA in order to verify the message from A. CA is also responsible to predistribute and revoke the key pairs of the nodes. The nodes and the CA together form the PKI, which can guarantee that no node can forge reports from other nodes.

2.4 Wormhole Attack Model

In wormhole attacks, the attackers between distant locations transmit packets using a out-of-band tunnel. The transmission tunnel is called a wormhole link. The packet loss rate on the wormhole link is negligible. The kinds of the wormhole links can be various, such as an Ethernet cable, an optical link, or a secured long-range wireless transmission [8]. When the wormhole attack is initiated, the attackers can capture data packets on either side, forward them through the wormhole link and rebroadcast them on the other node.

Figure 2-1 gives the basic scenario of wormhole attack. Two malicious nodes A and B are connected by a wormhole link. Hence B can have access to A's neighbors a_1, a_2, a_3 with only one hop, even though B is supposed to be remote from A's neighborhood. Clearly the wormhole link AB significantly changes the network's topological structure.

[1] Here each node includes the normal nodes in the wireless network, and the central administrator, which presents in our centralized algorithm.

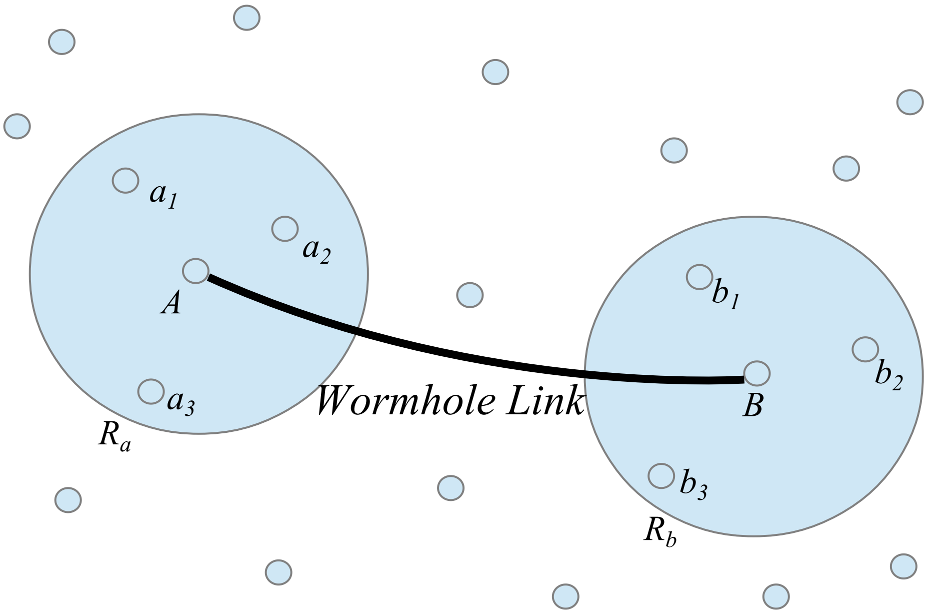

Figure 2-1: Wormhole attack model.

Chapter 3

Negative Impacts of Wormhole Links on RLNC Systems

As we have mentioned earlier in Chapter 1, wormhole attacks have severe impact on wireless network coding systems. Depending on different launching time, wormhole attacks can seriously downgrade the system performance (by forging link states and thus generating inefficient routing assignment), and cause individual nodes to deal with many non-innovative packets and waste their resources. We now examine these negative impact via simulations.

We configure a RLNC network with seven nodes randomly distributed as Figure 3-1. In the network coding system, MORE [5] is running (including the routing phase). The link loss probability $p'(u, v)$ between two nodes u and v is calculated as $p(u, v) = P_B \cdot f(d(u, v))$, where P_B is the baseline loss probability, and $f(\cdot)$ is a coefficient function based on the transmission distance $d(u, v)$. A data flow is established between node 1 (the source), and node 7 (the destination). The default data sending rate is 40 kbps.

3.1 Impacts on Throughput

We first examine wormholes' impact on network throughput if launched in the routing phase. In particular, we let the wormhole link be established between node 2 and 6, in

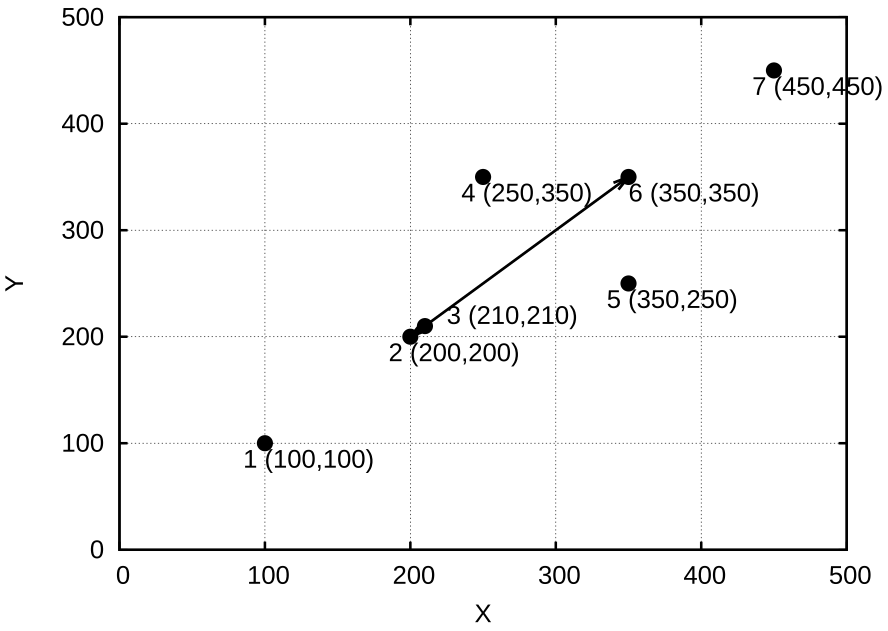

Figure 3-1: We show the coordinates of each node. In our simulation, the wormhole link between node 2 and 6 is valid when the topology sensing is going on. The wormhole link will disconnect one minute after the network starts to transmit the packets, giving a huge reduction in the network performance.

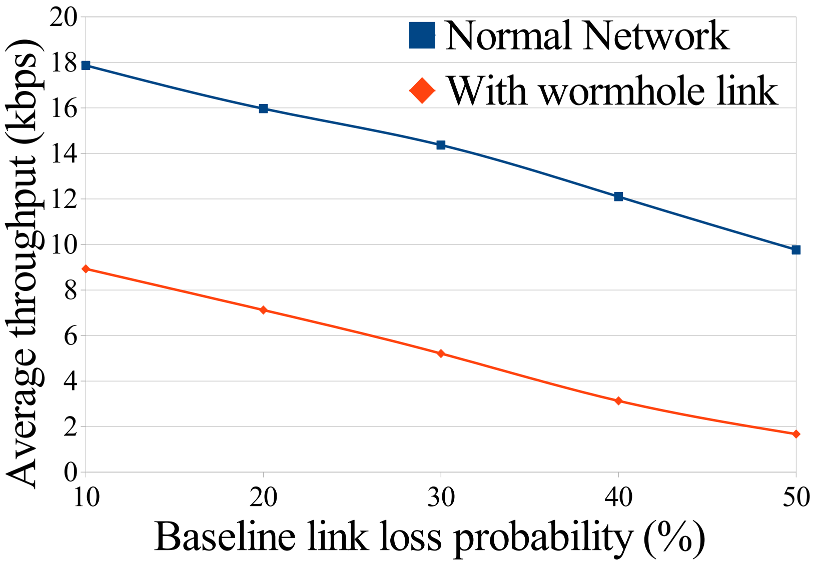

Figure 3-2: The average throughput for different baseline link loss probabilities with or without wormhole link.

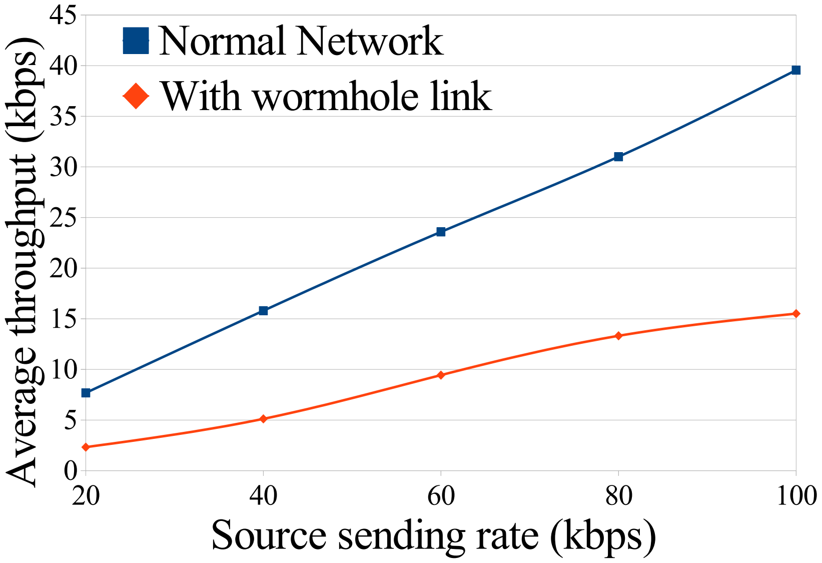

Figure 3-3: The average throughput for different source sending rates with or without wormhole link.

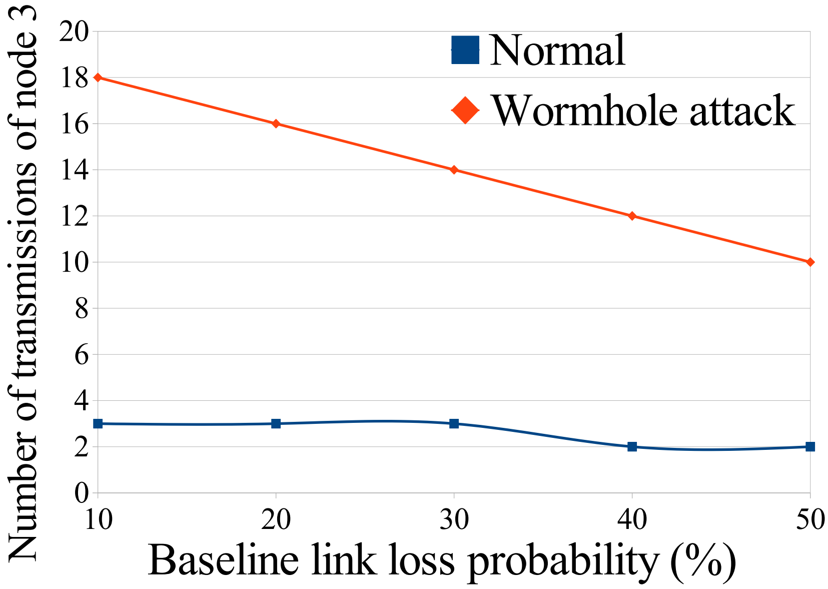

Figure 3-4: The No. of local transmissions (node 3) in RLNC network with or without wormhole attack

the routing procedure. The wormhole link disconnects one minute after the network starts to transmit the packets from the source to the destination. The simulation runs for 10 minutes, and we measure the average network throughput and compare it with the case without the wormhole attack.

Figure 3-2 shows the wormhole attack brings great negative influence on the network throughput. The throughputs of normal network are always greater than twice of those with wormhole link for the same setting. Similar results can be found in Figure 3-3 when we test the network with different data sending rates and $P_B = 30\%$. The reason is that the existence of wormhole link cheats each node in the topology sensing, making the ETXs of the surrounding nodes lower than the actual values. Thus, in the packet forwarding phase, when the wormhole link disconnects, the throughputs will decrease due to the insufficient times of packet forwarding.

3.2 Impacts on Local Workload

We then investigate the wormhole links' impact on local nodes' resource consumption if launched during data transmission phase. Figure 3-4 shows the number of transmissions of node 3 in different scenarios, with and without wormhole link respectively. The result demonstrates that Node 3 suffers a significant increment of transmissions and thus energy consumption for redundancy due to the wormhole link.

Chapter 4

Characterizing Wormhole Attacks in Wireless Network Coding Systems

As we have mentioned, detecting wormhole attacks in wireless network coding systems is difficult compared with traditional networks, due to the different nature of topology description and different principle of packet transmission. In order to facilitate the design of countermeasures, in this chapter we investigate the unique characteristics of network coding system behavior with wormhole links.

Unlike traditional networks, packet round trip time is not a valid metric for wireless network coding to distinguish the system under attack and the normal case. The fundamental reason is that with network coding, the packets being transmitted on each hop are different, and thus it is difficult to track down packets and record their trip time. Therefore, this packet-centric idea does not work for network coding. Instead, in this paper, our method is node-centric, i.e., we focus on the metrics that can be naturally obtained by nodes in the existing network coding protocols. In particular, we explore the relationship between the innovative packet transmission direction and ETX.

4.1 General Result

In wireless network coding systems, packets are transmitted from source to destination not in their original form. Actually, given fixed source and destination nodes, for a pair of intermediate neighbor nodes, it is difficult to tell whether the information flow directions are always the same. To figure out this question, we leverage one widely used metric, ETX. In wireless network coding systems, where no fixed routes exist, ETX, the expected number of the packets for the source node to transmit so that the target node (intermediate node or recipient) receives the packet, provides a way to portray the topological structure of the network and the relations among nodes. On the other hand, to describe the information flow direction, one important concept to explore is innovative packet, i.e., the packet received by a node containing new information that cannot be derived from already received packets.

In particular, for systems without wormhole links, we quantify the probability of the packet transmission directions between a pair of neighbor nodes, based on the concept of innovative packet and ETX, as shown in Theorem 1.

Theorem 1. For any two neighbor nodes u and v in the network satisfying $ETX(u) < ETX(v)$, the probability that v will receive an innovative packet from u is $\frac{ETX(v)-1}{ETX(u)+ETX(v)-1}$.

Proof. ETX is the *expected* number of the sent packets to make sure the forwarding node or recipient receives the innovative packet. Let $p = 1/ETX(u)$ and $q = 1/ETX(v)$, and then p and q are the probabilities to deliver the novel packet from the source node to u and v, respectively. Since $ETX(u) < ETX(v)$, $p > q$. We set up two random variables X and Y, that $X = x$ is the event it takes x packets to make u hear the innovative packet, and $Y = y$ is the event y packets make the novel packet arrive at v successfully. In fact, X and Y apply to geometric distribution and they are independent from each other. We calculate the probability of the event $X < Y$

as Equation (4.1).

$$
\begin{aligned}
P(X < Y) = \sum_{x=1}^{\infty} \sum_{y=x+1}^{\infty} P(X = x) P(Y = y) \\
= \frac{p - pq}{p + q - pq} = \frac{ETX(v) - 1}{ETX(u) + ETX(v) - 1} \qquad (4.1)
\end{aligned}
$$

□

The basic idea of this theorem is that in general information flows are more likely to be transmitted from the nodes of low ETXs to those of high ETXs.

When the network contains wormhole links, they will change the overall topological structure of the network. The actual ETX (with wormhole links) metric suffers a huge change as well, and then the transmissions of the novel packets are distinguishable from the expected.

4.2 Algorithm to Determine ETX

Since ETX is an important metric to characterize wormhole attacks, below we describe how to determine the ETX of each node based on the probability of successful transmission $P(i, j)$ between every two nodes v_i and v_j. Each $P(i, j)$ between two nodes can be measured by sending and receiving small packets and getting the statistical result. All the probabilities of successful transmission $P(\cdot, \cdot)$ together form the network adjacency matrix $\mathcal{P}$. We assume the matrix $\mathcal{P}$ is known for the network. We propose Algorithm 1 EDA to calculate the ETX for each node.

In Algorithm 1, we make the ETXs depict the difficulty of delivering the innovative packet to each node. For each transmission of innovative packet, any node can receive it as long as at least one of its neighbors has received the packet and successfully transmit it to the node. That gives rise to lines 9 to 11 in Algorithm 1. Another important revelation is any node can derive its own ETX given the ETXs of its neighbors and the relevant loss probabilities. That is, it is not necessary for each node to know the global ETXs or any location information of other nodes. Thus, our

Algorithm 1 ETX-Determining Algorithm (EDA)

```
Input: the entire network $G$ with nodes $V$ and their locations $L$, and the source node $v_s$
Output: the ETXs for all the nodes in the network $G$
 1: $ETX(v_s) \leftarrow 1.0$
 2: for each node $v_i$ in $V$, except $v_s$ do
 3:     $ETX(v_i) \leftarrow +\infty$
 4: end for
 5: repeat
 6:     $ETX_{updated} \leftarrow$ false
 7:     for each node $v_i$ in the network $G$, other than $v_s$ do
 8:         Let $N$ be the set of the neighbors of $v_i$ s.t. $ETX(v_k) < +\infty$ for any $v_k \in N$
 9:         if $ETX(v_i) > \frac{1}{1-\prod_{v_k \in N} \frac{1}{ETX(v_k)}(1-P(v_k,v_i))}$ then
10:             $ETX(v_i) \leftarrow \frac{1}{1-\prod_{v_k \in N} \frac{1}{ETX(v_k)}(1-P(v_k,v_i))}$
11:             $ETX_{updated} \leftarrow$ true
12:         end if
13:     end for
14: until $ETX_{updated} =$ false
15: return the ETXs for all the nodes
```

algorithms are independent on location information. In Theorem 2, we can show that Algorithm 1 can determine the ETXs with a unique answer.

Theorem 2. The returned solution of Algorithm 1 is unique.

Proof. We first study the scenario of connected network. The outer loop from line 5 to 14 keeps decreasing the ETXs based on the successful transmission probabilities between each pair of nodes. Note that if the ETX of any node decreases, the ETXs of all its neighbor nodes will remain the same according to the conditional branch at line 9. That is, if $ETX(v_i)$ decreases, and let v_j be any one of v_i's neighbors and let N be the set of v_j's neighbors, we have the value $\mathcal{T} = \frac{1}{1-\prod_{v_k \in N} \frac{1}{ETX(v_k)}(1-P(v_k,v_i))}$ increases and thus $ETX(v_j) < \mathcal{T}$. Then the value of $ETX(v_j)$ cannot change. Thus, there is no pair of nodes whose ETX values are directly dependent on each other. Algorithm 1 can halt in definite steps and output a unique solution. If the network is not connected, all the ETXs of the nodes that are not connected with the source node will be infinite, and the solution is also unique. Proof completes. □

Since our wormhole detection algorithm will rely on the values of ETXs, it is important to ensure that the system has appropriate defense against possible attacks on

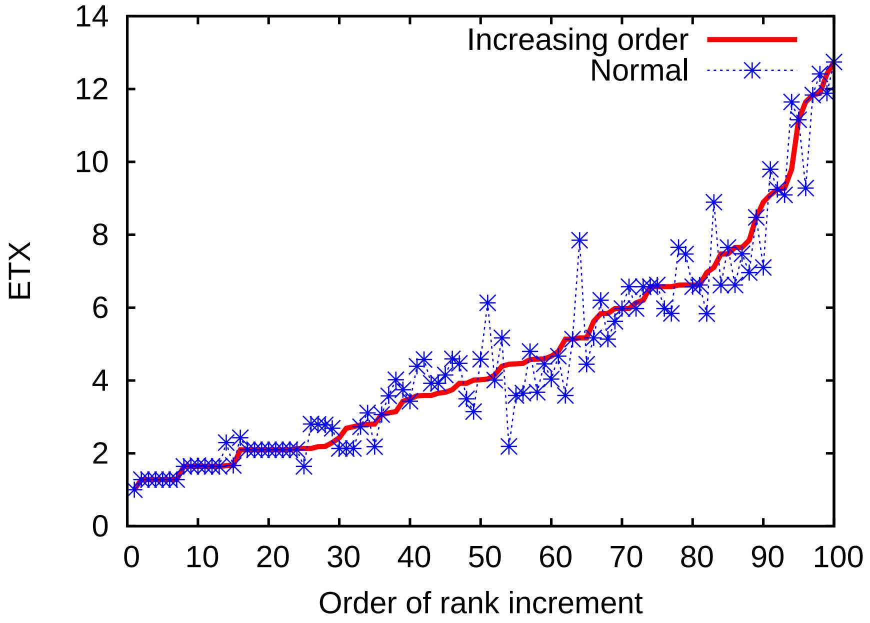

Figure 4-1: Node rank increment order of normal RLNC network.

ETXs. In practice link loss probabilities used in ETXs calculation are measured and reported using small control packets sent among nodes and these packets are transmitted under conventional protocols instead of network coding. To protect these protocols from wormhole attacks, existing countermeasures of wormholes in conventional wireless networks can be leveraged such as [13, 16, 15]. To defend against other cheating and malicious behavior in measuring link loss probabilities, i.e. submitting untruthful reports, both cryptographic and incentive-mechanism approaches can be used [22].

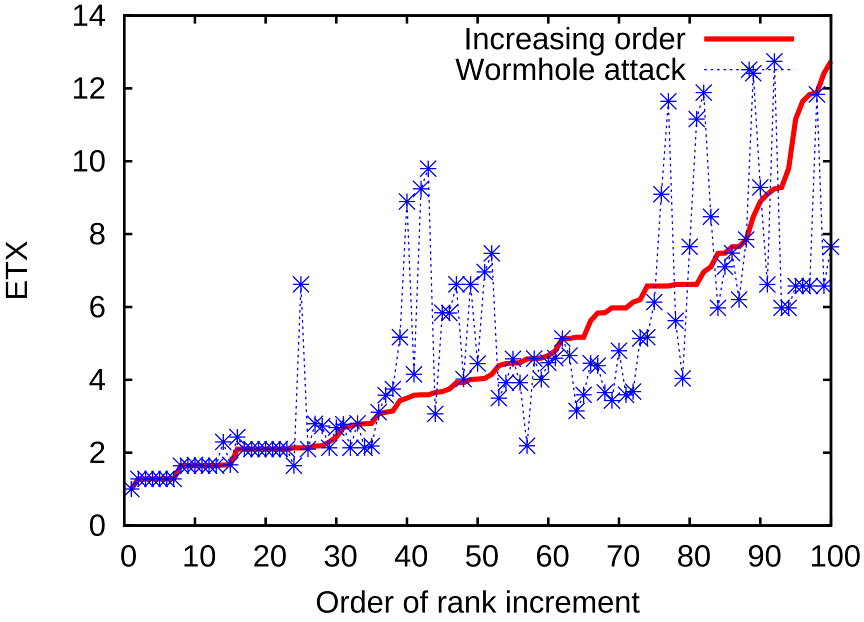

Figure 4-2: Node rank increment order of network under wormhole attack.

Chapter 5

The Centralized Algorithm

In this chapter, we propose the centralized algorithm, which utilizes the ETX metric and the order of rank increment to detect wormhole attacks. In order to protect the validity of our method, we also introduce the public cryptographic scheme for the network. For the proposed algorithm, we not only perform the analysis of its correctness, but also discuss its technical details in this chapter.

5.1 Algorithm Design

As what we have presented in Chapter 2.1, for each forwarding node in RLNC network, receiving the innovative packet will cause the rank of the previously received packets increases by one. We also find that the nodes with lower ETXs will be more likely to receive innovative packets (i.e., increase the rank) earlier than other nodes. On the other hand, wormhole links will make some nodes receive innovative packets (i.e., increase the rank) much earlier that they should. Thus, in the proposed centralized algorithm, we explore the order of rank increments in order to detect the wormhole links.

Basically, in RLNC, when an innovative packet is sent from the source node, the nodes near the source node are more likely to receive the innovative packets earlier than the nodes that are far from the source node. In Chapter 4, we have demonstrated ETX is a proper metric to measure the distances between each node and the source node. Thus, the nodes with low ETXs can probably receive the innovative packets earlier. However, the existence of wormhole link intuitively changes the normal network topology since the innovative packets can be transmitted through the wormhole link directly and safely, and thus the nodes around the remote side of the wormhole link can receive the novel packets earlier than expected. With a wormhole link, the order of the rank increments among the nodes will be significantly changed. To illustrate the significant changes, we have a RLNC simulation and Figures 4-1 and 4-2 demonstrate the orders of rank increments with and without wormhole link. Here we have 100 nodes in the network, and we run Algorithm 1 to calculate the ETXs. In the figures, the red curve denotes the ascending ETXs of the nodes. Then we start the network coding transmission. The source node sends out an innovative packet, and for each node, receiving the innovative packet will result in rank increment from 0 to 1. We collect the time stamps of rank increments on the nodes during the whole transmission, and find out the time order of rank increments. That is the blue line, which denotes the ETXs of the nodes based on the ascending time order of rank increments. We find that the blue line deviates from the red line when the wormhole link exists. That is, the wormhole link truly changes the network topology as well as the transmission flows. Therefore, we can observe the time order of rank increments, and release alerts when the deviation of the order exceeds the bound, which is set by the administrator. We can even determine the range of the nodes who may be involved in wormhole attack. For example, in Figure 4-2, the nodes whose ETXs are from 6.0 to 10.0 may be involved with wormhole attack, since they contribute majorly to the deviation of the blue curve.

For the centralized algorithm, we set up a central node, which owns the authority to gather information from all the nodes in the network, and we run a wormhole detection algorithm based on the rank increasing information on the central node. Each node is responsible to record the time when the rank of the received packets increases and then generates a report, which includes the details such as the time, the node address, and the rank. Each node delivers the reports to the central node via common unicast.

Based on the intuitions above, we propose Algorithm 2, the centralized algorithm to detect wormhole attacks on the central node. In Algorithm 2, the central node chooses an event of rank change, i.e., the rank increment from i to $i+1$, and then searches the received reports to find all the related ones. Then we compare the time order of ETXs with the ascending ETX sequence and calculate the distance between them. If the distance exceeds the threshold, we decide there exists wormhole attack, and release the warning. At last, we update the bound of the distance for the next detection, in order to make our algorithm adaptive. We apply *k-means* [23] to determine the bound, since k-means is powerful to learn the bound distinguishing two opposite samples[1].

In Algorithm 2, each report t is a tuple as Equation (5.1).

$$t = (time, addr, ETX, rank, K_{pub}, sig) \tag{5.1}$$

Here, $time$ denotes the time stamp of the rank increment; $addr$ denotes the address of the node who sends the report; ETX is the ETX of the reporting node; the value $rank$ means the rank increased from $rank - 1$ to $rank$. K_{pub} is the public key of the reporting node. sig is the digital signature of the report. The signature can be calculated by Equation (6.2). In Equation (6.2), we adopt a hashing function to obtain the abstract of the plain data $P = (time, addr, ETX, rank, K_{pub})$, and then encrypt the abstract using secret key K_{sec} of the local node. The result is the signature sig. In Algorithm 2, T_r denotes the set of the reports of rank increment from $r-1$ to r.

5.2 Analysis

We now perform the analysis of the correctness of Algorithm 2. That is, the wormhole link can remarkably aggravate the *Distance* in Algorithm 2 so that we can leverage some learning mechanisms to distinguish whether wormhole links exist with high

[1] Here the scenario is unsupervised learning since we do not know whether there exists wormhole link temporarily after each transmission. If we have the knowledge of some historic wormhole attacks, we can utilize the supervised learning algorithms, such as Support Vector Machine [24].

Algorithm 2 THE CENTRALIZED ALGORITHM

```
Input: $T$: the reports from all the nodes $V$ in the network $G$; $D$: the number of dimensions of the code vector space; $Normal$: the normal distance; $Threshold$: the threshold of alert
Output: whether there exists a wormhole attack in the network $G$; the updated $Normal$
Randomly select a rank $r$ s.t. $r \geq 1$ and $r$ should be small enough, i.e., $1 \leq r \leq 5$.
Let $T_r$ be the set of the reports whose rank increments are from $r - 1$ to $r$.
Sort $T_r$ into a sequence $T_r^e$ s.t. the values of ETX in $T_r^e$ are ascending.
Let $L_e$ be the sequence of ascending ETXs in $T_r^e$.
Sort $T_r$ into a sequence $T_r^t$ s.t. the values of time in $T_r^t$ are ascending.
Let $L_t$ be the sequence of ETXs in $T_r^t$ while preserving the order.
$Distance \leftarrow$ CALCULATE-DISTANCE($L_e$,$L_t$, $|V|$)
if $Distance - Normal > Threshold$ then
    Find out the addresses of the nodes with the most aberrant ETXs.
    Release a warning of wormhole attack.
end if
Update the value of $Normal$ using k-means.
```

Algorithm 3 CALCULATE-DISTANCE

```
Input: $L_1, L_2$: two lists; $n$: the number of nodes
Output: the distance between $L_1$ and $L_2$
Set up two $n$-dimensional vectors $X$ and $Y$.
$d \leftarrow 0$
for $i$ from 1 to $n$ do
    $d \leftarrow d + (L_1[i] - L_2[i])^2$
end for
return $\sqrt{d}$
```

accuracy. The *Distance* essentially illustrates the Euclidean distance between the two ETX orders. One order owns ascending ETXs, and the other applies the order of rank increments. We now analyze the *Distance* in quantitative way. We denote by $\delta(l)$ the set of the ETX differences between two nodes, and the location distance between the two nodes is l. For instance, in the network there is a node a with ETX 2.0, and there is a node b with ETX 5.0. The location distance between a and b is 10 units. Thus, the ETX difference $5.0 - 2.0 = 3.0$ is in the set $\delta(10)$. We denote by $\max_{l \le r} \delta(l)$ the maximum ETX difference when the location distance is no longer than r, and $\min_{l \ge r} \delta(l)$ denotes the minimum ETX difference when the location distance is no shorter than r. Let R be the radius of neighborhood, and let L be the length of wormhole link, if any. Then we have the lower bound of *Distance*.

Theorem 3. If there exists a wormhole link with length L in the connected RLNC network, we have

$$Distance \ge \min_{l \ge L} \delta(l) - \max_{l \le R} \delta(l). \tag{5.2}$$

Proof. Let the wormhole link connect the nodes a and b. Without loss of generality, we assume ETX$(a) <$ ETX(b). The order of the nodes with ascending ETXs should be

$$\cdots, a, c_1, c_2, \cdots, c_k, b, \cdots \tag{5.3}$$

The ETX difference between a and c_1 has to be no larger than $\max_{l \le R} \delta(l)$. Otherwise, in (5.3) no node before a (including a) can be neighbor of any node after a. Thus the RLNC network is not connected, with at least two separated components. The wormhole link can directly deliver the innovative packet from a to b. Thus, with the wormhole link, the order will be

$$\cdots, a, b, \cdots \tag{5.4}$$

We next estimate the lower bound of *Distance* based on (5.3) and (5.4). That is

$$\begin{aligned}
&Distance \\
\geq\, &\mathrm{ETX}(b) - \mathrm{ETX}(c_1) \\
=\, &\mathrm{ETX}(b) - \mathrm{ETX}(a) - (\mathrm{ETX}(c_1) - \mathrm{ETX}(a)) \\
\geq\, &\mathrm{ETX}(b) - \mathrm{ETX}(a) - \max_{l \leq R} \delta(l) \\
\geq\, &\min_{l \geq L} \delta(l) - \max_{l \leq R} \delta(l).
\end{aligned} \tag{5.5}$$

Here to obtain the first inequality, we consider the minimal difference between (5.3) and (5.4). That is, the change from (5.3) to (5.4) is only the swap between b and c_1. The above inequality finishes the proof. □

Since the length of wormhole link L is much longer than the neighborhood radius R, for most cases the ETX difference for L is much larger than that for R, and thus $\min_{l \geq L} \delta(l)$ is much larger than $\max_{l \leq R} \delta(l)$.[2] That is, we can guarantee a sufficiently large lower bound for the *Distance* given the long enough wormhole link. Since the estimation of the lower bound in the proof of Theorem 3 is quite rough, the actual *Distance* is much larger than the lower bound here. Thus, the wormhole link can guarantee a large enough *Distance* to make it recognizable.

5.3 Discussions

We now discuss the technical details in Algorithm 2. We first explain why we have the bounds (*Normal* and *Threshold*) on *Distance*, a value describing the deviation of ETX order, at line 8 of Algorithm 2. The main reason is that the *Distance* owns uncertainty in nature. In order to study the uncertainty in *Distance*, we need to analyze the transmissions of each innovative packet. Let p be the successful transmission probability over one hop. From the source node to the destined node, the number

[2] There are some cases that the ETX difference is very small even if L is much longer than R. To circumvent such scenarios, we may choose another pair of source and destination, and recalculate the ETXs to make the ETXs distinguishable.

of the sent packets until the transmission succeeds is a random variable X which conforms to geometric distribution. Denoted by T_X the ETX value of the destined node, and then T_X is the expected value of X[3]. Thus, we have

$$T_X = \mathrm{E}[X] = \sum_{k=1}^{\infty} kp(1-p)^{k-1} = \frac{1}{p}. \tag{5.6}$$

In RLNC, since the transmissions over the hops are independent from each other, the successful transmission probability p between two nodes is the product of all the probabilities of the hops that together connect both nodes.

$$p = \prod_{h_k \in \mathfrak{H}} p_k \tag{5.7}$$

where $\mathfrak{H}$ denotes the set of the hops that connect the source and destination, and p_k denotes the successful transmission probability over the hop h_k. In RLNC, each innovative packet can be transmitted through different sequences of hops $\mathfrak{H}$. Among the hop-sequences, there exists one that can transmit the packets in the shortest time. To make sure that the packets through the fastest hop-sequence arrive earlier than others in a high probability, the variance of X over the fastest hop-sequence should be as low as possible. We calculate the variance as follows:

$$\mathrm{Var}(X) = \mathrm{E}[X^2] - \mathrm{E}^2[X] = \frac{1-p}{p^2}. \tag{5.8}$$

Applying Equation (5.6), we obtain Equation (5.9).

$$\mathrm{Var}(X) = \mathrm{E}[X](\mathrm{E}[X]-1) = T_X(T_X - 1). \tag{5.9}$$

As the ETX value T_X increases, the variance $\mathrm{Var}(X)$ becomes even larger and it increases much faster than ETX. That means if the ETX of the target node is high (that is, the target node is far from the source node), it is difficult to predict the

[3] According to [17], ETX denotes the expected number of the packets to send until one packet successfully arrives at the destination.

time when the innovative packet arrives at the target node. In Figure 4-1, there are obvious deviations on the nodes of high ETXs (around 8.0 to 12.0), even though there is no wormhole link. Thus we have to give tolerance to such deviations. We define a deviation $Normal$ for the normal case at line 8 of Algorithm 2, and we leverage some unsupervised learning techniques to obtain the bounds $Normal$ and $Threshold$. Since wormhole links contribute more deviation by redirecting innovative packet flows, the centralized algorithm can correctly detect the wormhole links with proper learning algorithms.

We next explain why we choose a relatively small rank at line 1 of Algorithm 2. The essential reason is about the correspondence between the order of rank increments and the innovative packet sent by the source node. In RLNC, each innovative packet sent by the source contains a basis of the code vector space. For each forwarding node, an innovative packet must have at least one basis that the node has not received. Thus, the order of rank increments is essentially the order of receiving the basis from the source. However, if there are pervading undelivered innovative packets in the network, it is difficult to guarantee that the rank increment of each node is due to the latest basis sent by the source. Thus, it is desirable to observe the rank increments when there are relatively a small number of innovative packets pervading the whole network. In other words, the rank of the sent innovative packets should be small enough. Thus, we choose a relatively small rank at line 1 of Algorithm 2. A small enough rank can guarantee the correspondence between the order of rank increments and the basis, and thus the centralized algorithm can effectively distinguish whether there exists wormhole link based on the order of rank increments.

Chapter 6

The Distributed Detection Algorithm

In this chapter, we consider a practical scenario where centralized authority cannot be found. We propose DAWN, a distributed algorithm to detect wormhole attacks in wireless network coding systems. We will perform rigorous analysis on the detection rate of our algorithm and its resistance against collusions.

6.1 Algorithm Design

The basic idea of DAWN is based on the result of Theorem 1. For any two nodes in the neighborhood, the one with lower ETX is supposed to receive novel packets earlier than the other one with high probabilities. In other words, innovative packets are transmitted from low ETX nodes to high ETX nodes with high probabilities. In order to monitor the innovative packets transmission direction, nodes will work collaboratively. In particular, DAWN has two phases on each node: 1) *Report* packets direction observation results to its neighbors (Algorithm 4) and 2) *Detect* whether any attackers exist (Algorithm 5). The *Detect* phase is based on the received results from neighbors during the *Report* phase. Both of the algorithms are running on every node in the network. Algorithm 4 runs simultaneously while passing on the packets, and Algorithm 5 should be asynchronous for different nodes and run at random time slots.

Report Phase As shown in Algorithm 4, for each node, it will suspect that one neighbor is an attacker if it receives novel packets from the neighbor but the ETX of this neighbor is much higher than that of itself (i.e. , the distance between the ETXs is greater than the threshold δ). It sends its judgment as a report to its neighbors (Line 3 - 5). A node is called a *judge node* of a neighbor if the distance between their ETXs is greater than the threshold. Each report r is a tuple as Equation (6.1).

$$r = (time, A_{suspect}, A_{self}, K_{pub}, S_{novel}, sig) \tag{6.1}$$

Here, $time$ is when the reporting node discovers the abnormal transmission. $A_{suspect}$ is the address of the suspected node, which sends out a novel packet and owns a higher ETX than the recipient's. A_{self} is the address of the reporting local node. Since any node can modify the report when forwarding it, we need to apply cryptographic techniques to protect the integrity of the reports. We use digital signatures of the reports to defend against malicious modification, and abstract of the novel packet for administrative verification. Thus, we introduce symmetric cryptographic scheme into our system to make it more robust against attacks. In Equation 6.1 K_{pub} is the public key of the reporting node. S_{novel} is the set of the signatures of the received novel packets. sig is the signature of the report. The signatures are produced as Equation (6.2).

$$sig = Encrypt(K_{sec}, (Hash(P)) \tag{6.2}$$

Here K_{sec} is the secret key of the reporting node. P is the novel packet that was received from the target.

Detect Phase Algorithm 5 presents the pseudocode of the *Detect* phase of DAWN. For each node in the *Detect* phase, it receives reports from the judge nodes of any potential attackers. It first examines whether a report is from a valid judge node. If so, it will forward the report unless it has already been forwarded twice. Three-hops of the reports make sure that more (reachable) neighbors of the potential attacker will hear this report (Line 8). Figure 6-1 illustrates an example that a report is forwarded

Algorithm 4 `ReportFunction`

Input: $N(u)$: the set of u's neighbors; the number of the novel packets u received from each neighbor in the last batch; δ: the threshold on ETX difference.

Output: s_v: the local observation result for each neighbor $v \in N(u)$; Report messages if any.

```
for $v \in N(u)$ do
    Denote $p_v$ the number of novel packets that $u$ received from $v$ during the last batch
    if ETX($v$) − ETX($u$) > $\delta$ AND $p_v > 0$ then
        $u$ broadcasts the report $r(u, v, 0)$;
        Note: $r(u, v, 0)$ represents the report sent from $u$ about suspicious wormhole behavior of $v$, with hop count 0.
        $s_v = 1$;
    else
        $s_v = 0$;
    end if
end for
```

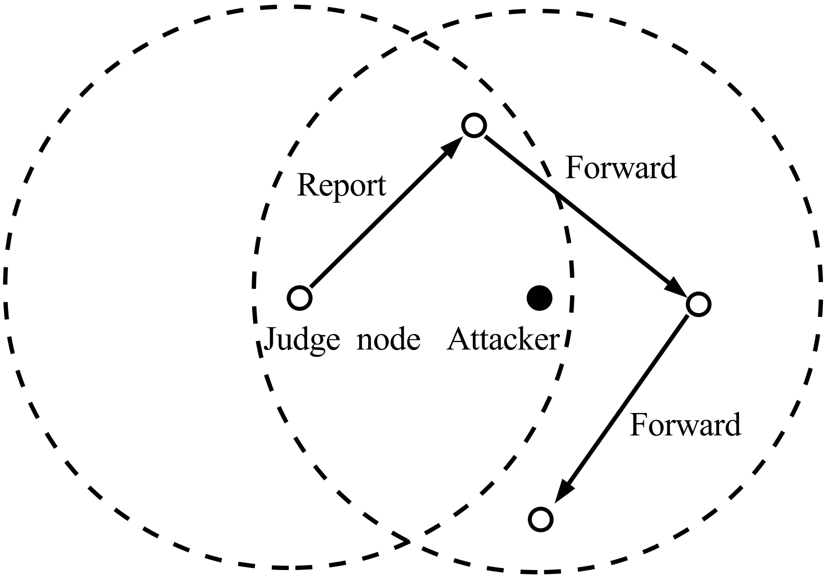

Figure 6-1: An illustration of report forwarding.

twice to make sure more neighbors receive it.

The detection algorithm on each node accumulates and calculates the number of its judge nodes who send report about the reported potential attacker in the current batch. If the number of judge nodes compose the majority (Line 15), the node will make the decision that the attacker is involved in a wormhole attack and block it from future communications.

Algorithm 5 The Distributed Detection Algorithm for Wormholes in Wireless Network Coding Systems(DAWN) on Node u

Input: R: the set of reports received in the last batch; $N(u)$: the set of u's neighbors; s_j: the local observation result of each neighbor $j \in N(u)$; δ: the threshold.

Output: Detected wormhole attackers in $N(u)$, if any.

```
 1: for Each report $r(i,j,k) \in R$ do
 2:     if ETX($j$) $-$ ETX($i$) $\leq \delta$ OR $i \notin N(j)$ then
 3:         Discard this report;
 4:     else
 5:         if $j \in N(u)$ then
 6:             $s_j \leftarrow s_j + 1$;
 7:         end if
 8:         if $k < 2$ then
 9:             Forward this report $r(i,j,k+1)$;
10:         end if
11:     end if
12: end for
13: for each $v \in N(u)$ do
14:     Let $C(v)$=$\{i \mid i \in N(v)$ s.t. ETX($v$) $-$ ETX($i$) $> \delta\}$
15:     if $s_v \geq \lceil \frac{|C(v)|+1}{2} \rceil$ then
16:         Mark $v$ as a detected wormhole attacker, and block any traffic from or to node
        $v$ in future batches.
17:     end if
18: end for
```

6.2 Lower Bound of Detection Rate

In this section, we will show our proposed distributed algorithm DAWN can perform well with a high lower bound on detection rate. In particular, we have obtain the result in Theorem 4.

Theorem 4. For an individual node v to be detected, let $N(v)$ denote the set of the neighbors of v, and $S(v)$ is the subset of $N(v)$ s.t.

$$\forall w \in S(v), ETX(w) - ETX(v) > \delta \tag{6.3}$$

Here δ is the threshold. Let $n = |S(v)|$, then the lower bound of the success rate of the algorithm is

$$B = 1 - \exp\left(-\frac{2(np - \lfloor\frac{n}{2}\rfloor)^2}{n}\right) \tag{6.4}$$

Here p is specified as Equation (6.5).

$$p = \frac{ETX(v) + \delta - 1}{2ETX(v) + \delta - 1} \tag{6.5}$$

Proof. Based on Theorem 1, one lower bound of the probabilities that one node in $S(v)$ will receive the novel packet earlier than v equals to p in Equation (6.5) by introducing the threshold δ. Thus, the success rate R satisfies

$$R \geq \sum_{k=\lceil\frac{n+1}{2}\rceil}^{n} \binom{n}{k} p^k (1-p)^{n-k} \tag{6.6}$$

The lower bound B can be determined by applying Hoeffding's inequality [25].

$$R \geq 1 - \sum_{k=0}^{\lfloor\frac{n}{2}\rfloor} \binom{n}{k} p^k (1-p)^{n-k} \tag{6.7}$$

$$\geq 1 - \exp\left(-\frac{2(np - \lfloor\frac{n}{2}\rfloor)^2}{n}\right) = B \tag{6.8}$$

□

To illustrate the lower bound more clearly, we now show some numerical results with different settings. Figure 6-2 demonstrates the lower bound of the detection rate of DAWN with various number of judge nodes and threshold (i.e., n and δ in Equation (6.4) and (6.5) respectively). We may set proper n and δ for each node (i.e. $n = 41$, $\delta = 10.0$, $ETX = 5.0$) in order to address the attackers successfully with a high

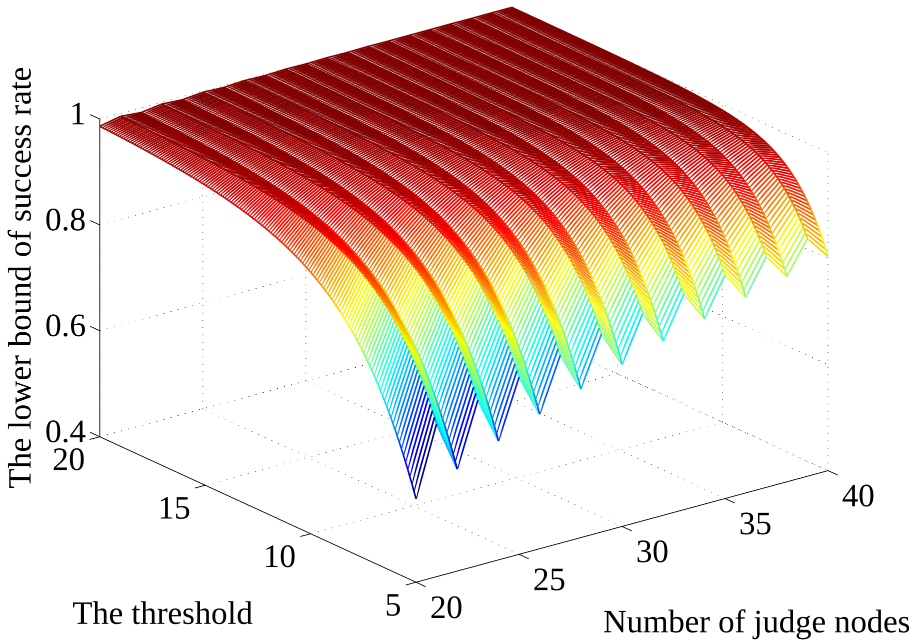

Figure 6-2: The lower bound of the success probability of the proposed distributed algorithm, with variables n and δ. The ETX of the node to be detected is 5.

Table 6.1: Lower bounds B for different scenarios

$ETX(v)$	δ	n	B
5.0	9.0	39	98.66
5.0	8.0	49	98.97
5.0	10.0	41	99.38

probability near 1, as what Table 6.1 indicates. As the simulations in Chapter 7, the real detection rate is much higher than the lower bound.

6.3 Collusion Resistance of DAWN

The distributed detection algorithm DAWN requires the collaboration of the wormhole attackers' neighbor nodes, i.e., monitoring attackers' behavior, sending, forwarding and analyzing reports. It is possible that although these nodes do not participate in wormhole links, they collude with wormhole attackers by making false reports against honest nodes or other misbehavior in the report procedure to make the detection algorithm malfunction.

In this section, we analyze the resistance of DAWN against collusions in the report procedure. In particular, we obtain a condition on the number of colluding nodes, under which DAWN is resistant against colluding attacks, as stated in Theorem 5.

Theorem 5. Let M be the set of the colluding nodes in the whole network. Then a necessary condition for DAWN to be resistant against colluding attacks is that Equation (6.9) holds for any node v.

$$|M \cap S(v)| < \lfloor \frac{|S(v)| + 1}{2} \rfloor \tag{6.9}$$

Here $S(v)$ is the same as in Theorem 4.

Proof. Sketch: We prove by contrapositive, i.e., if Equation (6.9) does not hold, the decision error rate is not bounded. Suppose that DAWN is making a decision whether any node v is a wormhole attacker. If v is innocent, all the malicious nodes in $S(v)$ can send false reports claiming v is involved in the wormhole attack. However, the number of the good nodes in $S(v)$ who can send reports indicating v is innocent is specified as Equation (6.10).

$$|S(v) \setminus M| < \lceil \frac{|S(v)| - 1}{2} \rceil \leq |M \cap S(v)| \tag{6.10}$$

Because it is the same with the scenario that most nodes of $S(v)$ is honest while v is malicious, it is impossible to judge whether v is malicious. For the case where v is a wormhole attacker and Equation (6.9) does not hold, similar conclusion can be drawn. □

For other scenarios where the colluding nodes dominate the neighborhood of wormholes attackers, since it falls out of the main scope of this paper, we omit the detailed solutions here and leave it to future work.

6.4 Attackers Can Be Smarter

Above discussions have covered the ordinary wormhole attack and the collusion attack. However, the attacker can manipulate the wormhole link more intellectually.

For example, the attacker may cheat its neighbors about its ETX by misreporting the link loss probabilities. The attacker can also initiate the wormhole link opportunistically. The ultimate objective of the attacker is to avoid being detected by the judge nodes. In this section, we discuss the defending solutions against these intellectual strategies.

The attacker can successfully forge its ETX only by misreporting the distances between its neighbors and itself. Otherwise, its neighbors, which are good nodes and can share correct information with each other, can find the attacker's claimed ETX is incorrect. Moreover, we can eliminate such misreporting behaviors by letting at least three of its neighbors work collaboratively to discover that the attacker's claimed position (based on its claimed distances) does not exist. Thus, we can make it impractical for the attacker to forge its ETX given the threshold and others' ETXs.

It is still possible for the attacker to apply a opportunistic policy on initiating the wormhole link, in order to avoid being detected. That is, the attacker only initiates the wormhole link when there is no judge node in its neighborhood, assuming the wireless network is dynamic (i.e. some mesh networks or sensor networks). For this scenario, we can assign some trusted nodes along the network boundary[1] to ensure that there are always enough judge nodes for each node. Even though this solution is a little expensive, we believe it is efficient when the network is deployed within a not too large area.

[1] There are very few nodes near the network boundary, indicating some nodes may have insufficient judge nodes.

Chapter 7

Evaluations

To evaluate the effectiveness and efficiency of our Centralized Algorithm and DAWN, we have developed a C based discrete event simulator for network coding systems and implemented our algorithms in the simulator.

7.1 Simulation Setup

We run our simulations on a Linux workstation (2.0 GHz CPU and 32 GB memory). We use the cryptography library Beecrypt [26] to implement the encryption and signature algorithms. We adopt RSA [27] and MD5 [28] algorithms with 4096-bit key size. We also simulate a certificate authority (CA), which manages the public keys and identities of the nodes. That is, a public key infrastructure (PKI) is applied in our simulations. When we calculate the time cost and communication overhead, we also include the contribution of PKI.

Performance Metrics: The main performance metrics in our evaluations include True Positive Rate (TPR), False Positive Rate (FPR), extra computation time and the ratio of extra communication over the total data transmissions. We define TPR and FPR as follows.

1. TPR, the true positives out of the positives, is defined as Equation (7.1).

$$\text{TPR} = \frac{\texttt{TP}}{\sum_{u \in M} |N(u)|} \tag{7.1}$$

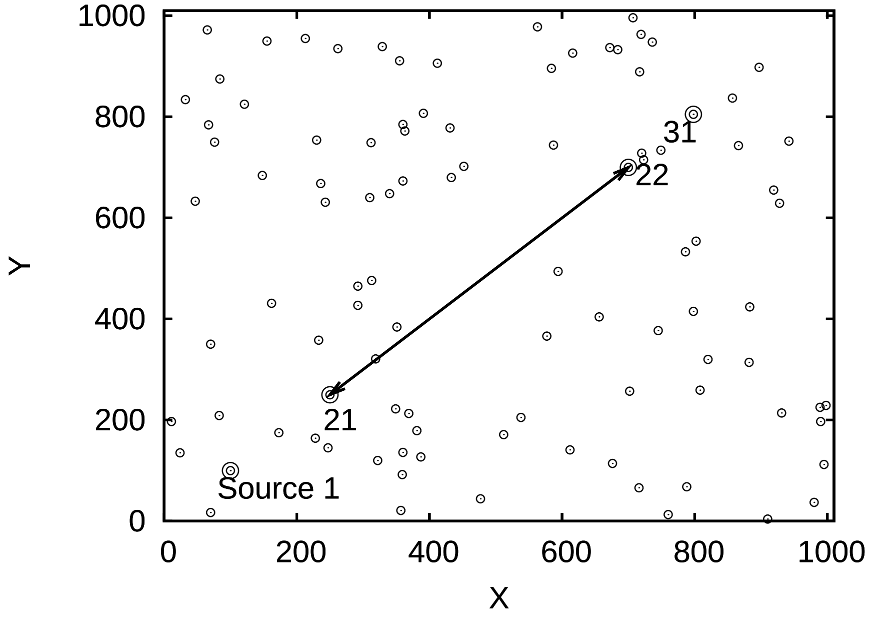

Figure 7-1: Deployment of the 100 nodes. Malicious node 21 and 22 are connected by a wormhole link. The attackers can enable or disable the wormhole link at any time.

Here TP denotes the number of the attackers' neighbors, who correctly detect the attack. $\sum_{u \in M} |N(u)|$ is the total number of attackers' neighbors.

2. FPR is false positives out of the negatives, as Equation (7.2).

$$\text{FPR} = \frac{\texttt{FP}}{\sum_{u \notin M} |N(u)|} \tag{7.2}$$

Here FP denotes the total number of the false detection alarms initiated by any node.

7.2 True Positive Rate v.s. False Positive Rate

To take a closer look at the effectiveness of our algorithms, our first simulation is on the network with a fixed topology. 100 nodes are distributed uniformly within the area of 1000x1000 length units, as Figure 7-1 illustrates. Two nodes, whose addresses are 21 and 22, are involved in the wormhole link. The attackers can initiate the wormhole link at any time during the simulation. We consider the unicast and broadcast. For the unicast case, the source node's address is 1 and the destination node's is 31. When node 31 receives each innovative packet, it sends an ACK message to node 1 in unicast. For broadcast, the destination node is not specified. We will test random topologies later, when we will choose the source and destined nodes randomly as well. For unicast, if the source and destination are not connected by the network, we eliminate this trial when we calculate the measurements.

Figure 7-2 presents the ROC diagram of Centralized Algorithm and DAWN with the fixed deployment of Figure 7-1. The points in the ROC diagram are drawn using the pairs of TPR and FPR with different thresholds, i.e., $Threshold$ in Algorithm 2 and δ in Algorithm 5. Too low threshold (i.e. $Threshold < 10$ or $\delta < 0.5$) will make both TPR and FPR near 100%. That is, the system is over sensitive and always gives false alarms. Reversely, too high threshold (i.e. $Threshold > 100$ or $\delta > 2.0$) will make both TPR and FPR near 0%. That is, the system seldom releases warnings about attacks, because the Centralized Algorithm is too tolerant, and for DAWN there will be few judge nodes of the target due to the strict requirement brought by high threshold. If we choose proper threshold (i.e. $Threshold$ around 50 and $\delta \in (1.4, 1.6)$), for Centralized Algorithm the TPR is over 92.00% and the FPR is less than 11.50%, and for DAWN the TPR is over 91.10% and the FPR is less than 12.01%. It verifies both Centralized Algorithm and DAWN can detect the attackers accurately.

The second set of simulations is on multiple networks with various topologies. We deploy 100 different topologies, and calculate the average TPR and FPR. For each topology, we run 100 instances. The TPR and FPR for each topology are averaged over the 100 instances. Figure 7-3 presents the ROC diagram of Centralized Algorithm and DAWN on networks with different topologies. The TPRs of both the algorithms still remain over 89.43% for multiple topologies and the FRPs can be less than 11.10%. The performance is a little worse than that in Chapter 7.2 as there are some scenarios where the wormhole link connected two nodes whose ETXs are close. It verifies that Centralized Algorithm and DAWN can detect the malicious nodes accurately for

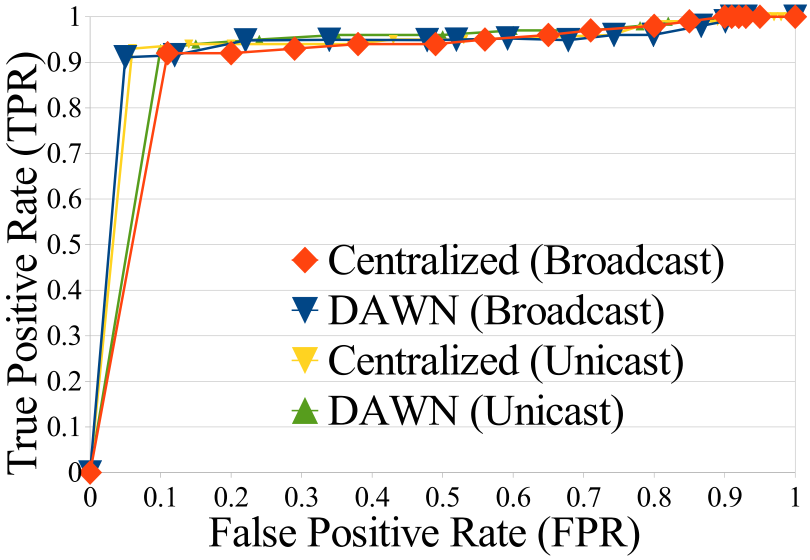

Figure 7-2: The ROC diagram of Centralized Algorithm and DAWN based on the deployment of Figure 7-1.

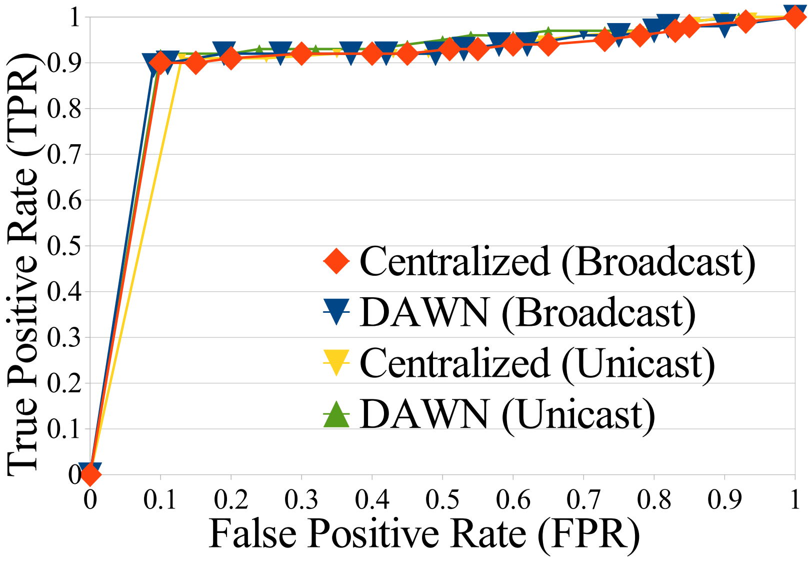

Figure 7-3: The ROC diagram of Centralized Algorithm and DAWN on networks with various topologies.

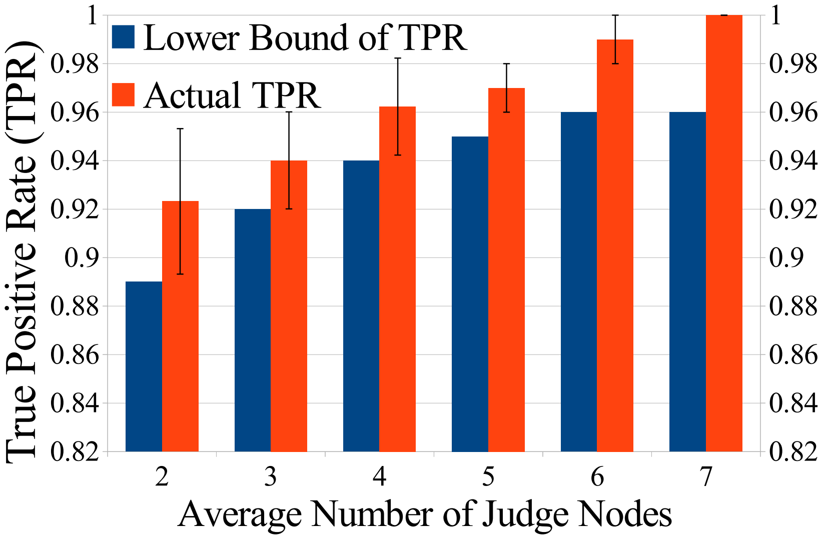

Figure 7-4: The TPR increases as the number of the judge nodes surrounding the attacker increases.

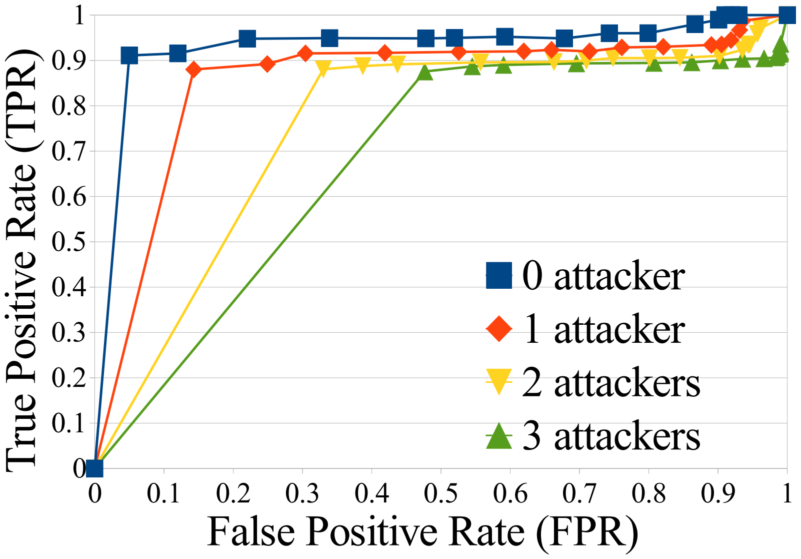

Figure 7-5: The ROC diagram of colluded attacks for different scenarios. The performance reduces as the number of attackers in the judge nodes increases. There were 7 judge nodes of the attacker in total.

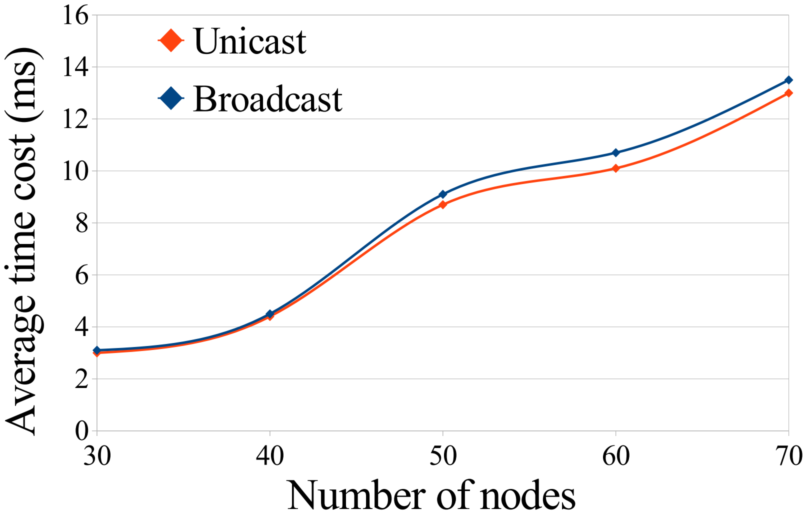

Figure 7-6: Centralized Algorithm: the average time cost of the central node in different scenarios

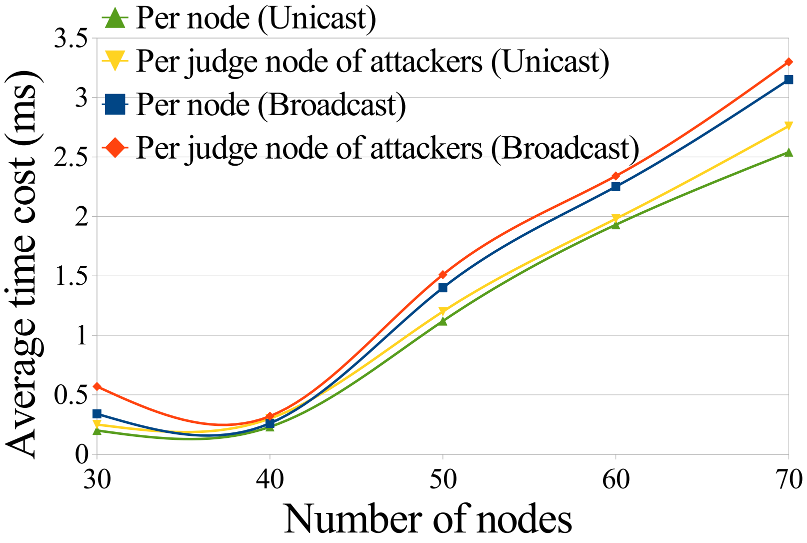

Figure 7-7: DAWN: the average time cost per each node in different scenarios

different scenarios.

7.3 Impact of the Amount of Judge Nodes on DAWN

To investigate the influence of the number of judge nodes on the performance of DAWN, we conduct the following experiments. We vary the node density in the network to change the number of judge node around the wormhole attackers. For different scenarios with different judge nodes, we calculate the actual TPR in the network as well as the theoretical lower bound (as described in Chapter 6.2).

Figure 7-4 demonstrates the TPR with different number of judge nodes in the unicast networks. Basically, we can see that both the actual TPR and the theoretical lower bound increase when the number of the judge nodes increases from 2 to 7. Even in the scenario where there are only 2 judge nodes around the wormhole attackers, the TRP can still be over 92.32%. Moreover, the actual TPR is always greater than the theoretical lower bound. It verifies the TPR can be sufficiently high if the number of the judge nodes is big enough.

7.4 Evaluation on Collusion Resistance of DAWN

In order to examine the capability of DAWN in resisting collusions among judge nodes. We test our algorithm in the scenarios with different numbers of colluding judge nodes. We perform experiments in the setting where there are 7 judge nodes in total. We observe the TPRs with different number of colluding nodes.

In Figure 7-5, it shows that the TPR decreases as the number of the colluding judge nodes increases. There is an abrupt reduction of the TPR when the number of colluding nodes changes from 3 to 4. In the cases with 1, 2 and 3 colluding nodes, all the TPRs are over 87.41%. It verifies that DAWN has strong resistance against the colluding attacks.

7.5 Overhead

We investigate the overheads of Centralized Algorithm and DAWN using two metrics: the computation time and communication overhead in percentage.

7.5.1 Computation Cost

We measure the average computation time of the central node in the Centralized Algorithm, as well as the average time cost per each node in the network for DAWN. The simulation takes one batch of the data transmission.

For Centralized Algorithm, Figure 7-6 shows the average computation time cost of the central node, when we set different node densities of the network (with different number of nodes in the 1000x1000 sized area). The computation time grows linearly with the total number of the nodes, since more nodes can generate more reports for the central node to process. For both unicast and broadcast, the total time cost is several milliseconds, which is tolerable for most RLNC systems.

For DAWN, Figure 7-7 shows the average computation time cost per node and per batch with various node densities. We can observe that our algorithm costs more time, when there are more nodes in the network and correspondingly more events to monitor and report. Overall it shows the computation time cost of DAWN is tolerable for most RLNC applications, with a few milliseconds at most.

7.5.2 Communication Overhead

For communication overhead, the metric we use is the ratio of the number of the extra packets generated by the wormhole attack defending algorithm, and the number of the original total data packets transmitted. Table 7.1 and Table 7.2 show the communication overheads for Centralized Algorithm and DAWN in unicast and broadcast respectively. It demonstrates that both the communication overheads are tolerable if the node density in the network is not too high.

Table 7.1: The communication overhead statistics for unicast

# nodes	Centralized Alg. overhead (%)	DAWN overhead (%)
30	1.32	3.54
40	1.44	3.64
50	3.01	8.22
60	3.53	10.49
70	4.11	12.72

Table 7.2: The communication overhead statistics for broadcast

# nodes	Centralized Alg. overhead (%)	DAWN overhead (%)
30	1.45	4.34
40	1.65	3.71
50	3.41	9.43
60	3.94	12.44
70	4.32	15.90

Chapter 8

Related Works

Random Linear Network Coding (RLNC) has extensive applications in wireless networking community as it can significantly boost the throughput and utilization of the information capacity [1, 2]. Many schemes that can achieve RLNC have been proposed, such as ExOR [3], COPE [4] and MORE [5]. It is challenging to bring these solutions to the real world due to the complexity of implementation or lacking research of the related security problems. The naive RLNC is vulnerable to several types of attack, such as pollution attack [29], Byzantine attack [30] and wormhole attack [9]. We will focus on wormhole attack at RLNC network.

For classically modeled networks, researchers have offered several solutions to detect and avoid such attacks [9, 8, 10, 13, 14, 31, 32, 15, 33, 16, 12, 34, 6, 35]. We can divide these existing solutions into two major categories: utilizing temporal and spatial information, and detecting network topology change based on topology and graph analysis.

- The first category [9, 13, 31, 15, 16, 32] uses information on time or space within the network. For example, Hu et al [13] use packet leashes to detect wormhole attacks, by appending in each packet the location information of the senders and they accordingly detect the physically impossible transmissions. Both [16] and [15] are based on the round-trip travel time of packet to detect wormhole links. Khalil et al [32] introduced the guard node to help the local node detect the malicious attackers, assuming the network had a static topology. Li et al [31] proposed forced collisions to defend against wormhole attacks. There are two major limitations for the methods dependent on time and space: the nodes in the network have to be tightly synchronous and the node location information is available [32].

- In the second category [10, 6, 14, 8, 35], topological structures and graph models are used. Among them, Wang et al use visualization methods to detect wormhole links in sensor networks, revealing the intrinsic change of network topological structure under attacks [14]. In [6], Dong et al detect and locate various wormholes and relies on observing inevitable topology deviations introduced in the network by wormholes. Maheshwari et al [8] leverage the connectivity information to detect wormhole links within wireless multi-hop networks. They assume the network can always be modeled as a graph, and the connectivity can always be represented by edge between vertices. Nevertheless, in wireless network coding systems, the connectivity in the network is described in different ways than traditional networks. Xu et al [35] use the abnormalities in hop counts brought by wormhole link. The significant change in hop counts reveals the essential change in network topological structure. The authors can detect such a change by sending probe packets. However, in RLNC, each innovative packet may spread by a series of multicasting and thus a hop counting scheme cannot be established. Thus one cannot use the ideas in the second category.

Before my work, there is no solution of the wormhole attack detection for wireless network coding systems.

Chapter 9

Conclusion

In this paper, we have investigated the negative impacts of wormhole attacks on wireless network coding systems. We have proposed two algorithms that utilize the metric ETX to defend against wormhole attacks. We have proposed a Centralized Algorithm that assigns a central node to collect and analyze the forwarding behaviors of each node in the network, in order to react timely when wormhole attack is initiated. We have proven the correctness of the Centralized Algorithm by deriving a lower bound of the deviation in the algorithm. We have also proposed a Distributed detection Algorithm against Wormhole in wireless Network coding systems, DAWN. DAWN is totally distributed for the nodes in the network, eliminating the limitation of tightly synchronized clock. DAWN is efficient and thus it fits for wireless sensor network. For both centralized and distributed algorithms, we have utilized the digital signatures to ensure every report is undeniable and cannot be forged by any attackers. The simulations have shown that the proposed algorithms can detect the malicious nodes participating in wormhole attack with high successful rate and the algorithm is efficient in terms of computation and communication overhead.

Bibliography

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

[2] 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 Transactions on Information Theory*, vol. 52, no. 10, 2006.

[3] S. Biswas and R. Morris, "Opportunistic routing in multihop wireless networks," in *ACM SIGCOMM*, September 2004.

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM*, September 2006.

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

[6] D. Dong, Y. Liu, X. Li, and X. Liao, "Topological detection on wormholes in wireless ad hoc and sensor networks," *IEEE Transactions on Networking*, vol. 19, 2011.

[7] J. Kim, D. Sterne, R. Hardy, R. K. Thomas, and L. Tong, "Timing-based localization of in-band wormhole tunnels in manets," in *ACM WiSec*, 2010.

[8] S. R. D. R. Maheshwari, J. Gao, "Detecting wormhole attacks in wireless networks using connectivity information," in *IEEE INFOCOMM*, 2007.

[9] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole attacks in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, 2006.

[10] R. Poovendran and L. Lazos, "A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks," *Wireless Network*, vol. 13, no. 1, 2007.

[11] A. J. Newell, R. Curtmola, and C. Nita-Rotaru, "Entropy attacks and countermeasures in wireless network coding," in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, ser.

WISEC '12. New York, NY, USA: ACM, 2012, pp. 185–196. [Online]. Available: http://doi.acm.org/10.1145/2185448.2185473

[12] W. Wang, B. Bhargava, Y. Lu, and X. Wu, "Defending against wormhole attacks in mobile ad hoc networks: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 6, no. 4, pp. 483–503, Jun. 2006. [Online]. Available: http://dx.doi.org/10.1002/wcm.v6:4

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *IEEE INFOCOMM*, March 2003.

[14] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe)*, October 2004.

[15] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, "Truelink: A practical countermeasure to the wormhole attack in wireless networks," in *Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, ser. ICNP '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 75–84. [Online]. Available: http://dx.doi.org/10.1109/ICNP.2006.320200

[16] S. Čapkun, L. Buttyán, and J.-P. Hubaux, "Sector: secure tracking of node encounters in multi-hop wireless networks," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, ser. SASN '03. New York, NY, USA: ACM, 2003, pp. 21–32. [Online]. Available: http://doi.acm.org/10.1145/986858.986862

[17] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, 2005.

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

[19] A. S. Avestimehr, S. N. Diggavi, and D. N. Tse, "Wireless network information flow: A deterministic approach," *Information Theory, IEEE Transactions on*, vol. 57, no. 4, pp. 1872–1905, 2011.

[20] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6463–6486, 2011.

[21] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys (CSUR)*, vol. 37, no. 2, pp. 164–194, 2005.

[22] F. Wu, T. Chen, S. Zhong, L. E. Li, and Y. R. Yang, "Incentive-compatible opportunistic routing for wireless networks," in *Proceedings of the 14th ACM*

international conference on Mobile computing and networking, ser. MobiCom '08. New York, NY, USA: ACM, 2008, pp. 303–314. [Online]. Available: http://doi.acm.org/10.1145/1409944.1409979

[23] S. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[24] C. Cortes and V. Vapnik, "Support vector machine," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[25] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association*, vol. 58, no. 301, 1963.

[26] Beecrypt. [Online]. Available: http://sourceforge.net/projects/beecrypt/

[27] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[28] R. Rivest, "The md5 message-digest algorithm," *RFC 1321*, 1992.

[29] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, March 2009.

[30] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Proceedings of the 2004 IEEE International Symposium on Information Theory (ISIT)*, January 2004.

[31] Z. Li, D. Pu, W. Wang, and A. Wyglinski, "Forced collision: detecting wormhole attacks with physical layer network coding," *Tsinghua Science and Technology*, vol. 16, no. 5, 2011.

[32] I. Khalil, S. Bagchi, and N. B. Shroff, "Liteworp: a lightweight countermeasure for the wormhole attack in multihop wireless networks," in *Dependable Systems and Networks (DSN)*, July 2005.

[33] L. Qian, N. Song, and X. Li, "Detection of wormhole attacks in multi-path routed wireless ad hoc network: a statistical analysis approach," *Journal of Network and Computer Applications*, vol. 30, no. 1, 2007.

[34] L. Buttyan, L. Dora, and I. Vajda, "Statistical wormhole detection in sensor networks," in *European Workshop on Security and Privacy in ad-hoc and sensor networks*, July 2005.

[35] F. Makedon, G. Chen, J. Ford, and Y. Xu, "Detecting wormhole attacks in wireless sensor networks," *International Federation for Information Processing Digital Library*, vol. 253, no. 1, 2010.

VITA

Shiyu Ji

Candidate for the Degree of

Master of Science

Thesis: WORMHOLE ATTACK DETECTION ALGORITHMS IN WIRELESS NETWORK CODING SYSTEMS

Major Field: Computer Science

Biographical:

Education: B.E. Harbin Institute of Technology, 2012.

Experience: Teaching Assistant, Oklahoma State University, 2012-2015.