DAWN: Defending Against Wormhole Attacks in Wireless Network Coding Systems

Shiyu Ji
Oklahoma State University
Stillwater, OK 74075
shiyu@cs.okstate.edu

Tingting Chen
Oklahoma State University
Stillwater, OK 74075
tingtic@cs.okstate.edu

Sheng Zhong
Nanjing University
Nanjing, China
zhongsheng@nju.edu.cn

Subhash Kak
Oklahoma State University
Stillwater, OK 74075
subhashk@cs.okstate.edu

***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 paper, we quantify wormholes' devastating harmful impact on network coding system performance through experiments. Then 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 does not introduce any overhead by extra test messages. Extensive experimental results have verified the effectiveness and the efficiency of DAWN.**

I. 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.

Sheng Zhong was supported in part by RPGE, NSFC-61321491, and NSFC-61300235. Part of the work was done while supported in part by NSF-0845149.

978-14799-3360-0/14/$31.00 ©2014 IEEE

To further quantify the impact of wormhole attacks in wireless network coding systems, we perform extensive experiments and investigate the results in Section III.

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 paper, we propose a distributed algorithm to detect wormhole attacks in wireless intra-flow network coding systems. The main idea of our solution is that we examine the order of nodes receiving 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 algorithm does not rely on any location information, global synchronization assumption or special hardware/middleware. It is only based on the local information that can be obtained from regular network coding protocols, and thus it does not introduce any overhead of extra test messages.

The contribution of this paper is summarized 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 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 nodes 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.

The rest of this paper is organized as follows. Section II introduces related technical preliminaries. Then we demonstrate the detrimental influences of wormhole attack in Section III. In Section V, we 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 Section VI. After the Related Work Section VII we conclude this paper in Section VIII.

II. Technical Preliminaries

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

A. Random Linear Network Coding (RLNC)

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 (1).

$$s = f(r_1, r_2, \cdots, r_n) \tag{1}$$

For RLNC, f in Equation (1) is a random linear combination in the field $GF(2^k)$.

$$f(r_1, r_2, \cdots, r_n) = \sum_{i=1}^{n} \xi_i r_i \tag{2}$$

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. If one intermediate node receives a packet which is linearly independent from previous packets, this packet is called an *innovative* packet. When the destination receives m packets, whose vectors are linearly independent from each other, it can restore the source information S based on the received data R.

$$S = C^{-1}R \tag{3}$$

Here C is the matrix of the coefficients of the received packets. The capacity of RLNC converges to the optimum in probability [2], and owns an ideal performance on the compression of the transmitted data.

B. Expected transmission count (ETX)

ETX has extensive applications in network coding systems [5]. 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 deployment 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 (4).

$$ETX(v) = \frac{1}{p(u, v)} \tag{4}$$

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].

C. Attack Model of Wormholes

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 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.

III. Impact of the wormhole links on RLNC systems

As we have mentioned earlier in Section I, 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 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.

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 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.

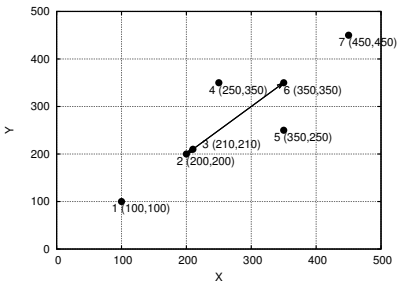

Fig. 1. We show the coordinates of each node. In our simulation, the wormhole link between node 2 and 6 is valid when the topology sensing is going on. The wormhole link will disconnect one minute after the network starts to transmit the packets, giving a huge reduction in the network performance.

Figure 2 shows the wormhole attack brings great negative influence on the network throughput. The throughputs of normal network are always greater than twice of that with wormhole link for the same setting. Similar results can be found in Figure 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.

We then investigate the wormhole links' impact on local nodes' resource consumption if launched during data transmission phase. Figure 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.

IV. 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 section 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. In stead, 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

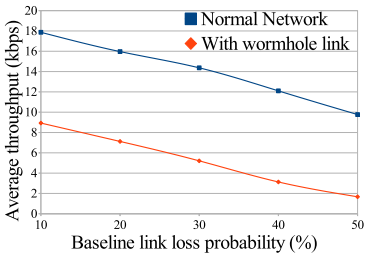

Fig. 2. The average throughput for different baseline link loss probabilities with or without wormhole link.

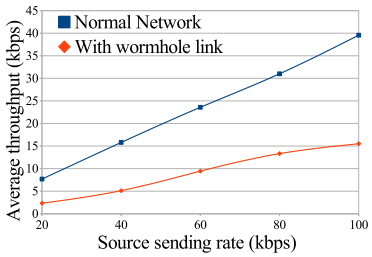

Fig. 3. The average throughput for different source sending rates with or without wormhole link.

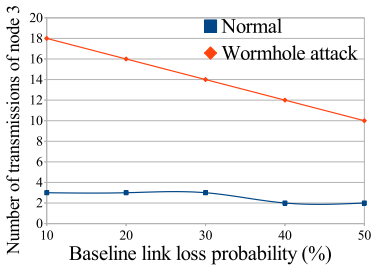

Fig. 4. The No. of local transmissions (node 3) in RLNC network with or without wormhole attack

the relationship between the innovative packet transmission direction and ETX.

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 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 (5).

$$P(X<Y) = \sum_{x=1}^{\infty}\sum_{y=x+1}^{\infty} P(X=x)P(Y=y)$$
$$= \frac{p-pq}{p+q-pq} = \frac{ETX(v)-1}{ETX(u)+ETX(v)-1} \quad (5)$$

■

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) deployment suffers a huge change as well, and then the transmissions of the novel packets are distinguishable from the expected.

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. We can show that Algorithm 1 can determine the ETXs with a unique answer. Due to space limitation, we leave the proof of this result in a technical report [18].

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 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

A target node can be intermediate node or recipient.

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: $\quad ETX(v_i) \leftarrow +\infty$
4: **end for**
5: **repeat**
6: $\quad ETX_{updated} \leftarrow$ false
7: $\quad$ **for** each node v_i in the network G, other than v_s **do**
8: $\quad\quad$ Let N be the set of the neighbors of v_i s.t. $ETX(v_k) < +\infty$ for any $v_k \in N$
9: $\quad\quad$ **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: $\quad\quad\quad ETX(v_i) \leftarrow \frac{1}{1-\prod_{v_k \in N} \frac{1}{ETX(v_k)}(1-P(v_k,v_i))}$
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: **until** $ETX_{updated}$ = false
14: **return** the ETXs for all the nodes

such as [13], [16], [15]. To defend against other cheating and malicious behavior in measuring link loss probabilities, e.g., submitting untruthful reports, both cryptographic and incentive-mechanism approaches can be used [19].

V. THE DISTRIBUTED DETECTION ALGORITHM

Based on our findings above, in this section, we design 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.

A. 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 2) and 2) *Detect* whether any attackers exist (Algorithm 3). 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 2 runs simultaneously while passing on the packets, and Algorithm 3 should be asynchronous for different nodes and run at random time slots.

Report Phase As shown in Algorithm 2, 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).

$$r = (time, A_{suspect}, A_{self}, K_{pub}, S_{novel}, sig) \quad (6)$$

Algorithm 2 `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.

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

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 sheme into our system to make it more robust against attacks. In Eq. 6 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 (7).

$$sig = Encrypt(K_{sec}, (Hash(P)) \quad (7)$$

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 3 presents the psudocode 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 5 illustrates an example that a report is forwarded twice to make sure more neighbors receive it.

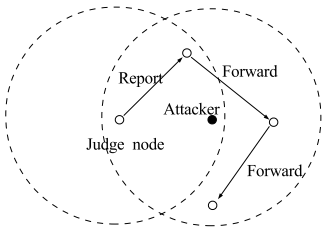

Fig. 5. An illustration of report forwarding.

Algorithm 3 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**

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.

B. Lower Bound of Detection Rate

In this subsection, 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 2.

Theorem 2. 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.

$$\forall w \in S(v), ETX(w) - ETX(v) > \delta \tag{8}$$

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

$$B = 1 - \exp\left(-\frac{2(np - \lfloor \frac{n}{2} \rfloor)^2}{n}\right) \tag{9}$$

Here p is specified as Equation (10).

$$p = \frac{ETX(v) + \delta - 1}{2ETX(v) + \delta - 1} \tag{10}$$

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 (10) by introducing the threshold δ. Thus, the success rate R satisfies

$$R \geq \sum_{k=\lceil \frac{n+1}{2} \rceil}^{n} \binom{n}{k} p^k (1-p)^{n-k} \tag{11}$$

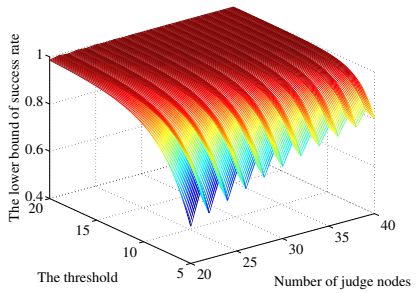

Fig. 6. The lower bound of the success probability of the proposed distributed algorithm, with variables n and δ. The ETX of the node to be detected is 5.

TABLE I
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

The lower bound B can be determined by applying Hoeffding's inequality [20].

$$R \geq 1 - \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{k} p^k (1-p)^{n-k} \tag{12}$$

$$\geq 1 - \exp\left(-\frac{2(np - \lfloor \frac{n}{2} \rfloor)^2}{n}\right) = B \tag{13}$$

■

To illustrate the lower bound more clearly, we now show some numerical results with different settings. Figure 6 demonstrates the lower bound of the detection rate of DAWN with various number of judge nodes and threshold (i.e., n and δ in Eq. (9) and (10) 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 probability near 1, as what Table I indicates. As the simulations in Section VI, the real detection rate is much higher than the lower bound.

C. 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 subsection, 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 3.

Theorem 3. 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 (14) holds

for any node v.

$$|M \cap S(v)| < \lfloor \frac{|S(v)|+1}{2} \rfloor \quad (14)$$

Here $S(v)$ is the same as in Theorem 2.

Proof: Sketch: We prove by contrapositive, i.e., if Equation (14) 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 (15).

$$|S(v) \setminus M| < \lceil \frac{|S(v)|-1}{2} \rceil \leq |M \cap S(v)| \quad (15)$$

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 (14) 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.

VI. Evaluations

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

A. Simulation Setup

We run our simulations on a Linux workstation (2.0 GHz CPU and 32 GB memory). We use the cryptography library Beecrypt [21] to implement the encryption and signature algorithms. We adopt RSA [22] and MD5 [23] algorithms with 4096-bit key size.

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 specify TPR and FPR as follows.

1) TPR, the true positives out of the positives, is defined as Equation (16).

$$\text{TPR} = \frac{\texttt{TP}}{\sum_{u \in M} |N(u)|} \quad (16)$$

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 (17).

$$\text{FPR} = \frac{\texttt{FP}}{\sum_{u \notin M} |N(u)|} \quad (17)$$

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

B. True Positive Rate v.s. False Positive Rate

To take a closer look at the effectiveness of our algorithm, 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 illustrates. Two nodes, whose addresses are 21 and 22, are involved in the wormhole link. The source node's address is 1 and the destination node's is 31. The attackers can initiate the wormhole link at any time during the simulation.

Figure 8 presents the ROC diagram of DAWN with the fixed deployment of Figure 7. The points in the ROC diagram are drawn using the pairs of TPR and FPR with different thresholds δ in Algorithm 3. Too low threshholds (i.e. $\delta < 0.5$) 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. $\delta > 2.0$) will make both TPR and FPR near 0%. That is, the system seldom releases warnings about attacks, because 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. $\delta \in (1.4, 1.6)$), the TPR is over 91.10% and the FPR is less than 12.01%. It verifies 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 9 presents the ROC diagram of DAWN on networks with different topologies. The TPRs of DAWN still remain over 89.43% for multiple topologies and the FRPs can be less than 11.10%. The performance was a little worse than that in Section VI-B as there were some scenarios where the wormhole link connected two nodes whose ETXs were close. It verifies that DAWN can detect the malicious nodes accurately for different scenarios.

C. Impact of the Amount of Judge Nodes

To investigate the influence of the number of judge nodes on the performance of DAWN, we conduct the following experiments. We vary the nodes 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 Section V-B).

Figure 10 demonstrates the TPR with different number of judge nodes in the network. 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.

D. 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

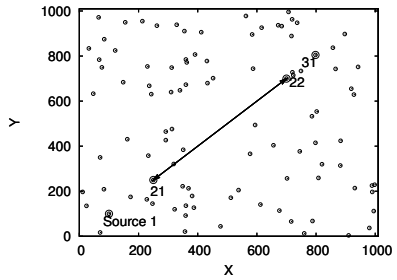

Fig. 7. Deployment of the 100 nodes. Malicious node 21 and 22 are connected by a wormhole link. The attackers can enable or disable the wormhole link at any time.

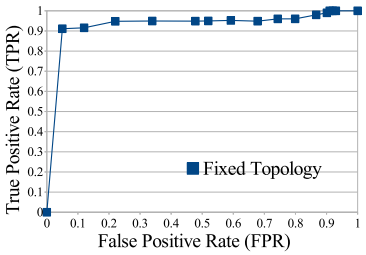

Fig. 8. The ROC diagram of DAWN based on the deployment of Figure 7.

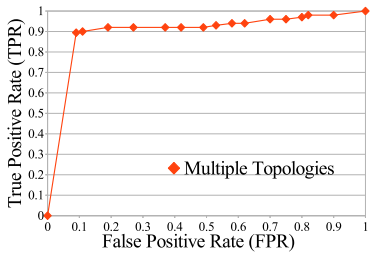

Fig. 9. The ROC diagram of DAWN on networks with various topologies.

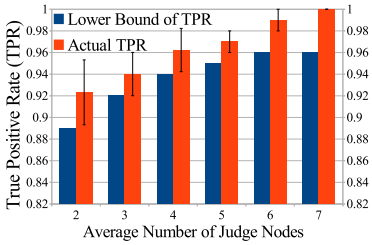

Fig. 10. The TPR increased as the number of the judge nodes surrounding the attacker increased.

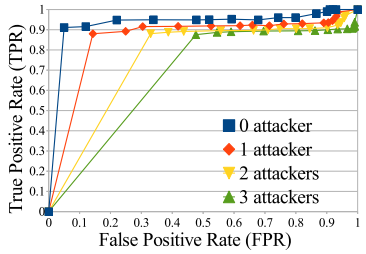

Fig. 11. The ROC diagram of colluded attacks for different scenarios. The performance reduces as the number of attackers in the judge nodes increases. There were 7 judge nodes of the attacker in total.

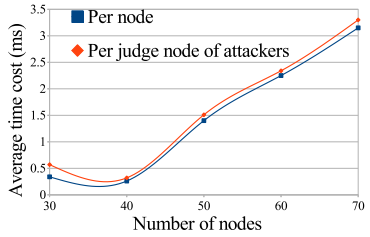

Fig. 12. The average time cost for different scenarios

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 11, 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.

E. Overhead

We investigate the cost of DAWN using two metrics: the extra computation time and communication overhead in percentage.

1) Computation Cost: We measure the computation time of our algorithm on each node in the network, for one batch of the data transmission. Table 12 shows the average computation time cost per node and per batch, when the nodes density in the network varies (with different number of nodes in the 1000x1000 sized area). 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 RLNC applications, with a few milliseconds at most.

2) Communication Overhead: For communication overhead, the metric we measure is the ratio of the extra packets generated by DAWN and the original total data packets transmitted. Table II shows the communication overhead for DAWN. It demonstrates that the communication overhead is tolerable if the nodes density in the network is not too high.

TABLE II
THE COMMUNICATION OVERHEAD STATISTICS

# nodes	DAWN overhead (%)
30	4.34
40	3.71
50	9.43
60	12.44
70	15.90

VII. RELATED WORKS

RLNC has extensive applications in wireless network field as it improves the throughput and utilization of the information capacity greatly [1], [2], such as ExOR [3], COPE [4] and MORE [5]. It is challenging to bring these solutions to realities 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 [24], Byzantine attack [25] and wormhole attack [9]. In this paper, we focus on wormhole attack at RLNC network.

For traditional networks, researchers offered several solutions to detect and avoid such attacks [9] [8] [10] [13] [14] [26] [14] [27] [15] [28] [12] and [29]. These solutions can be divided into two major groups: utilizing temporal and spatial

information, and detecting network topology change based on graph analysis. For example, Hu et al. use packet leashes to detect wormhole attacks [13], 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. introduced the guard node to help the local node detect the malicious attackers, assuming the network had a static topology [27]. There were two 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 [27]. In the second group [10], [6], [14], [8], among others, 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. As we mentioned earlier, in wireless network coding systems, the connectivity in the network is described in different ways than traditional networks. Unfortunately, there is no solutions of the wormhole attack detection for wireless network coding systems.

VIII. Conclusion

We have 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. We utilize the digital signatures to ensure every report is undeniable and cannot be forged by any attackers. The simulations have shown that the proposed algorithm 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.

References

[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. yang 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] S. Ji, T. Chen, S. Zhong, and S. Kak, "Dawn: Defending against wormhole attacks in wireless network coding systems," in *Technical Report, Oklahoma State University.*, July 2013. [Online]. Available: http://www.cs.okstate.edu/ tingtic/papers/technicalreport.pdf

[19] 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

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

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.