Toward Secure Network Coding in Wireless Networks: Threats and Challenges

Jing Dong Reza Curtmola Ruben Sethi Cristina Nita-Rotaru
Department of Computer Science, Purdue University
{dongj,crix,rsethi,crisn}@cs.purdue.edu

***Abstract*—In recent years, network coding has emerged as a new communication paradigm that can significantly improve the efficiency of network protocols by requiring intermediate nodes to mix packets before forwarding them. Recently, several real-world systems have been proposed to leverage network coding in wireless networks. Although the theoretical foundations of network coding are well understood, a real-world system needs to solve a plethora of practical aspects before network coding can meet its promised potential. These practical design choices expose network coding systems to a wide range of attacks.**

In this paper, we identify two general frameworks that encompass several network coding-based systems proposed for unicast in wireless networks. Our systematic analysis of the components of these frameworks reveals vulnerabilities to a wide range of attacks, which may severely degrade system performance. Adequate understanding of these threats is essential to effectively design secure practical network coding systems.

I. Introduction

Network coding is a promising paradigm that has been shown to improve throughput and provide elegant solutions to problems that were traditionally considered difficult, such as congestion control and reliability. The core principle of network coding is that intermediate nodes actively mix (or code) input packets and forward the resulting coded packets.

Several practical systems were proposed to bridge theory with practice [1]–[6], in the context of unicast for wireless mesh networks. Although the theoretical foundations of network coding are well understood, real-world systems need to solve a plethora of practical aspects before network coding meets its promised potential. As such, network coding systems make numerous practical design choices and optimizations that are essential to leverage network coding and achieve good performance. Unfortunately, in the quest for performance, security aspects are disregarded: Many of these design choices result in protocols that have numerous security vulnerabilities. Thus, there is a need for network coding based systems which are more resilient in adversarial environments.

Previous work relevant to the security of network coding has focused exclusively on the *packet pollution* attack, in which attacker nodes inject corrupted packets in the network. Several solutions were proposed to combat this attack [7]–[9]. However, in the aforementioned practical systems, packet pollution is only one of many potential attacks.

In this paper, following a well-known security principle that states a system is as secure as its weakest link, we focus on the security of a network coding system in its entirety and examine all of its components. We describe two general frameworks that encompass several network coding-based systems proposed for unicast in wireless networks. Depending on how they leverage the benefits of network coding, we classify these systems into *intra-flow* network coding systems [1]–[3], which mix packets within the same individual flows, and *inter-flow* network coding systems [4]–[6], which mix packets across multiple different flows.

We systematically analyze the components of these frameworks and identify potential security vulnerabilities that may severely degrade system performance. To the best of our knowledge, this is the first paper to systematically analyze the security of each component in practical network coding-based wireless systems. As a proof of concept, we experimentally demonstrate the severity of attacks in network coding systems. Our experiments show that even a single attacker whose only action is dropping packets can cause over 50% of throughput degradation for over 80% of flows.

We identify the challenges in addressing the security vulnerabilities and propose a general direction for defense. The multitude of attack avenues presented by current network coding-based systems leads us to conclude there is a tension between the performance of such systems and their security. Mostly due to protocol complexity, it becomes extremely difficult to secure such systems in their entirety. Our paper should be viewed as a cautionary note pointing out the frailty of current network coding-based wireless systems.

II. Related Work

Previous work related to security of network coding focuses exclusively on the packet pollution attack. Solutions to packet pollution attacks can be categorized into cryptographic approaches and information theoretic approaches. Cryptographic approaches rely on specialized homomorphic hash functions [10], [11] or homomorphic digital signatures [7]–[9] that allow intermediate nodes to filter out polluted packets. Information theoretic approaches do not to filter out polluted packets at intermediate nodes; they either encode enough redundant information into packets which allows receivers to detect pollution [12], or use a distributed protocol which allows receivers to tolerate pollution [13].

III. Overview of Network Coding-based Wireless Systems

In this section, we present the general frameworks for network coding systems for wireless mesh networks. There are two general approaches for applying network coding to wireless mesh networks, intra-flow network coding and inter-flow network coding. Both approaches exploit the *broadcast advantage* and *opportunistic listening* in wireless networks to reduce transmissions and improve performance. However, these benefits are realized differently: Intra-flow network coding systems mix packets within individual flows, while inter-flow network coding systems mix packets across multiple flows.

We focus on the system aspects of the protocols, such as control message exchanges, data delivery process, and node state maintenance.

A. Intra-flow Network Coding

In intra-flow network coding systems, packets are delivered in batches. Each node forwards linear combinations of the packets in a batch, which are referred to as *coded packets*. The source node continuously broadcasts linear combinations of packets from the current batch until an acknowledgment (ACK) for this batch is received from the destination, at which point the source begins transmitting the next batch of packets.

The coded packets are relayed to the destination via a set of intermediate nodes, referred to as *forwarder nodes*. Each forwarder node stores linearly independent packets it overhears and forwards new coded packets by combining packets stored in its buffer. When the destination node receives enough linearly independent coded packets, it decodes the packets by solving a system of linear equations and unicasts an ACK packet to the source node, allowing the source to start sending the next batch.

An intra-flow coding system consists of the following components: Forwarding node selection and rate assignment, data packet forwarding, and acknowledgment delivery.

1) Forwarding node selection and rate assignment: This process determines the forwarder node set and the rate at which each forwarder node forwards coded packets. The optimal selection of forwarder nodes and rate assignment needs to take into consideration several factors, including the relative distance of each node to the source and destination, interference among nodes, and fairness among different flows. Due to the global nature of the input required, existing intra-flow coding protocols [1]–[3] use a centralized approach, where the computation is based on a link state graph maintained at each node as in a link state routing protocol. Typically, the source performs the centralized computation of the forwarding node set and rate assignment, and disseminates this information to the other nodes piggybacked on data packets.

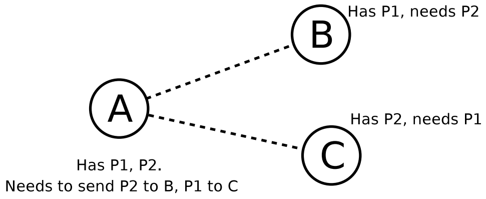

Fig. 1. Node A can broadcast $p_1 \oplus p_2$ to allow B and C to decode the packet and retrieve p_2 and p_1, respectively. COPE [4] applies network coding for immediate neighboring nodes. In general, node B and C can be several hops away from A. As long as they have one of the packets, they can still decode the combined packet to retrieve the other packet. DCAR [5] considers this more general case.

2) Data packet forwarding: The forwarder nodes and the destination node maintain a buffer of linearly independent packets that they overhear. Each forwarder node broadcasts coded packets generated from random linear combination of coded packets stored in its buffer at a pre-assigned rate as discussed above. When the destination node overhears enough linearly independent coded packets, it decodes the packets by solving a system of linear equations, and initiates the acknowledgment process as described below.

3) Acknowledgment delivery: The ACK packet is delivered from the destination to the source using the traditional single path routing process via the best quality path. The timely and reliable delivery of ACK is critical to ensure that the source moves to the next batch quickly. Thus, each intermediate nodes delivers ACK packets with high priority and ensures reliability by mandating an explicit acknowledgment from the next hop.

B. Inter-flow Network Coding

Inter-flow network coding exploits opportunistic listening and wireless broadcast with *opportunistic coding* at intermediate nodes. The key idea is that when a node has a set of packets for different flows to be delivered to different next hop nodes, instead of unicasting each packet individually to its corresponding next hop node, the node combines the packets together and broadcasts the combined packet once for all the next hop nodes. Therefore, inter-flow coding reduces multiple individual unicast transmissions to only one broadcast transmission. The condition for this to work is that the later hops of the packets have overheard the necessary packets to decode the coded packet. Below we refer to this condition as the *decodability condition*. The general setting of inter-flow coding is illustrated in Fig. 1. Since node B and C have overheard packet p_1 and p_2, respectively, node A can broadcast $p_1 \oplus p_2$, which will allow both node B and C to obtain their desired packet.

An inter-flow coding system generally consists of the following three components: Discovery of coding opportunities, transmission of coded packets, and routing integration for increased coding opportunities.

1) Discovery of coding opportunities: Coding opportunities at a node refer to the packets that can be coded together for transmission such that the decodability condition is satisfied. Based on the scope considered for coding opportunities, we can classify inter-flow coding protocols into *localized coding protocols* (only one hop neighbors of a node are considered for potential coding opportunities) and *global coding protocols* (all the nodes in the network are considered). In both cases, the discovery of coding opportunities requires a node to collect information about packet reception at other nodes.

In localized coding protocols (*e.g.*, COPE [4] and [14]), each node periodically reports its packet reception to its neighbors via local broadcasts. To deal with loss of reception reports, link qualities are also used to guess whether a neighboring node receives a packet. For example, if a neighbor of a node has very good link quality to the previous hop of a packet, then the node can infer that this neighbor also receives the packet with high confidence.

In global coding protocols (*e.g.*, DCAR [5]), each node keeps track of all other nodes in the network that can overhear a packet by maintaining the neighboring node set of all the nodes on the path of the packet. To achieve this, the protocol follows a common flood-based on-demand route discovery process, except that each node includes in the route request message its neighboring node set in addition to its own identifier.

2) Transmission of coded packets: To achieve a reliability guarantee similar to the 802.11 link layer reliability, a node needs to ensure that a coded packet is received by all the intended next hops. However, 802.11 broadcast lacks the reliability of unicast communication. To address this problem, a *pseudo-broadcast* technique ([4], [5]) has been commonly adopted. With pseudo-broadcast, the sender node sends coded packets with 802.11 unicast using one of the intended next hops as the MAC receiver. The packet will be retransmitted multiple times until the designated MAC receiver receives the packet and acknowledges it. The multiple retransmissions also allow other next hop nodes more opportunities to receive the packet. To guarantee full reliability, the other next hop nodes are also required to acknowledge the packet, which is achieved by piggybacking the ACK on other packets broadcast by the node. If the ACKs of some next hop nodes are not received after a timeout period, the sender node retransmits the packet for such nodes by either sending them individually or coding them with other packets.

3) Routing integration: An inter-flow coding protocol can be designed independently of routing protocols, where coding opportunities arise from incidental path intersections. A natural extension to further improve the performance is to design coding-aware routing protocols, so that paths are selected to maximize the benefit of coding. Such coding-aware routing protocols are usually realized with new coding-aware metrics, which discount the cost of links that allow coding. The optimal path selection based on such metrics can be performed in either distributed or centralized fashion. For example, in DCAR [5], the metric aggregation and path selection follows the same steps of traditional source routing protocols, except that each node also considers coding opportunities when computing path metrics. In contrast, [6] adopts a centralized link state routing-like approach, where each node floods its own coding-aware link metrics and local flow information in the network. The source node then computes the optimal paths based on the complete network information.

IV. Threats, Challenges, and Safe Design Principles

In this section, we present potential security threats in current network coding systems and the challenges in addressing the security threats. We focus solely on attacks on the network coding system that aim to disrupt the data delivery process.

A. Intra-flow Network Coding

We analyze security vulnerabilities in each component of intra-flow network coding systems.

1) Forwarding node selection and rate assignment: A key input to this process is the link state graph, which is maintained as in a link state routing protocol as follows. Each node monitors its local link qualities, and periodically floods the information in the entire network. Albeit being simple, there are numerous security vulnerabilities that can result in incorrect link state graphs at nodes, and consequently incorrect selection of forwarding nodes and rate assignment.

• **Link Quality Falsification or Modification.** The attacker node can claim false metrics for its own adjacent links. Such attacks are difficult to prevent, as this information is local to the attacker node. A reactive approach, *e.g.* in [15], that detects the attack and then reactively identifies and isolates the attacker, may be a more viable solution. Alternatively, the attacker node may modify link qualities reported by other nodes as they are flooded in the network. Such attacks can be prevented using message authentication, such as digital signatures.

• **Wormholes.** Wormhole attacks can introduce fictitious links between honest nodes, and distort their perception of network topology. Although existing wormhole solutions, such as packet leashes [16] and TrueLink [17] can be applied, they typically incur substantial overhead, which can potentially nullify the performance gain of network coding.

Designing a secure and efficient link state protocol is a challenging task. To our knowledge, there is no existing solution that ensures correct link state propagation in wireless networks in the presence of colluding attackers.

2) Data forwarding process: The data forwarding process is subject to packet pollution and packet dropping attacks.

• **Packet Pollution.** Packet pollution attacks are the most well-known and most studied attacks against network coding systems. In packet pollution attacks, the attacker node injects corrupted packets into the network. Since each forwarder node combines received packets to form new coded packets, such attacks can cause an epidemic effect, where the corrupted packets from one affected honest node further affect other honest nodes. As a result, by injecting even a few corrupted packets, the attacker can degrade the performance significantly. Existing defense techniques are generally heavy-weighted (see Section II), leading to a significant negative impact on the protocol performance. Designing a lightweight pollution defense for intra-flow coding systems in wireless networks is still an open problem.

• **Packet Dropping.** Intuitively, network coding systems should be resilient to packet dropping attacks due to the inherent redundancy in multi-path routing and opportunistic listening. However, in current protocols, the selection of forwarding nodes and rate assignment are carefully optimized to reduce interference and the total number of transmissions. As a side effect, this results in fragile systems that are sensitive to node misbehaviors, even as simple as packet dropping. As demonstrated by our experiments (Section V), even a single packet dropping attacker can result in over 40% of degradation in performance for most flows. Addressing packet dropping attacks in network coding systems is more challenging than in traditional routing protocols, as the number of packets a node transmits and the time of transmissions are contingent on the opportunistic receptions at the node. As a result, traditional approaches, *e.g.*, watchdog [18], no longer apply.

3) Acknowledgment delivery process: The timely and reliable delivery of ACK messages is critical to the performance of the protocol. This process is vulnerable to the following attacks.

• **ACK Injection or Modification.** The attacker forges a bogus ACK or modifies an ACK packet causing the source to move onto the next batch prematurely. As a result, the destination may receive only partial batches, and consequently is not able to decode any data packets. Such attacks can be prevented with message authentication, such as digital signatures.

• **ACK Dropping.** If the attacker node lies on the ACK delivery path, it can drop all the ACK packets. This can prevent the source node from advancing through batches and cause it to keep transmitting one batch of packets forever. An attacker can enhance their chance of being selected on the ACK delivery path by manipulating path metrics or using wormhole attacks.

• **ACK Delay.** The attacker node delays the delivery of ACK packets, instead of dropping ACK packets completely. This attack is more stealthy than ACK dropping attacks and can also cause a significant throughput degradation, as it prolongs the time required for sending a batch of packets.

B. Inter-flow Network Coding

We also analyze each of the components of inter-flow coding systems for potential security vulnerabilities.

1) Discovery of coding opportunities: The correct discovery of coding opportunities at a node relies on the correct packet reception information that the node collects from other nodes. The process of collecting the packet reception information of other nodes is subject to various types of attacks for both localized and global coding protocols as follows.

• **Packet Reception Information Mis-reporting.** In localized coding protocols (*e.g.*, COPE [4]), an attacker can impersonate honest nodes and report incorrect packet reception information to their neighbors. Such an attack can cause a node to send coded packets that cannot be decoded by the intended next hop nodes. Since such non-decodable packets cannot be acknowledged, it will cause the sender node to continuously transmit useless packets. This attack can be addressed with message authentication schemes. However, given the high frequency of packet reception reports, the authentication scheme needs to be extremely lightweight.

• **Link State Pollution.** Localized coding protocols also rely on the link quality between nodes to infer packet reception status at other nodes. Therefore, attacks on link state routing protocols which cause incorrect link state graph at nodes can also cause a node to infer incorrect packet reception information and consequently send non-decodable packets.

• **Neighbor Set Pollution.** In global coding systems (*e.g.*, DCAR [5]), a node determines coding opportunities based on the neighboring node set information collected during the route discovery process. An attacker can cause the collection of incorrect neighboring node set information either by direct modifications of route request packets or by using wormholes to introduce fictitious links. The resulting incorrect neighboring node set can cause a node either to miss coding opportunities, or worse yet, to send coded packets that cannot be decoded by downstream nodes, which can potentially degrade the throughput of the targeted flow to zero. Existing approaches for secure source routing protocols, such as Ariadne [19], can be used to authenticate the neighboring node information, and prevent malicious modifications. However, techniques for defending wormhole attacks are also necessary for securing the neighbor set information.

2) Transmission of coded packets: The packet transmission process in inter-flow coding systems is also subject to various types of attacks as follows.

- **ACK Injection or Modification.** By injecting bogus ACKs or modifying ACKs, the attacker node can cause premature ending of necessary packet retransmissions in the pseudo-broadcast technique, resulting in the failure of packet reception at next hop nodes. Again, message authentication schemes can be used as a countermeasure; however, due to the high frequency of ACK messages, it is crucial that the scheme selected is efficient in terms of both computation and bandwidth.

- **Packet Pollution.** As in intra-packet coding schemes, by injecting only a few corrupted packets an attacker can cause epidemic corruption of packets. Existing pollution defense schemes proposed for intra-flow coding, such as homomorphic signatures, cannot be applied in this context, as coded packets are formed from packets generated by different sources. Defending against pollution attacks in inter-flow coding systems is still an open problem.

- **Packet Dropping.** Compared to traditional routing protocols, inter-flow coding systems encourage path sharing in an effort to increase coding opportunities. By exploiting such a tendency in the path selection, an attacker can manage to be selected by many paths, and hence can disrupt the communication of many flows via packet dropping. Existing defense techniques, such as watchdog [18], no longer apply in inter-flow coding schemes, because in such systems, the packets forwarded by each node are coded packets, which in most cases are not decodable by its upstream node; hence, the forwarding of a packet cannot be verified via simple overhearing.

3) Routing integration: In order to select an optimal route that considers coding opportunities, a coding-aware routing protocol not only requires the correctness of link and path metrics as in traditional routing protocols, but it also requires the correctness of coding benefits reported by each node. Thus, in addition to manipulating link and path metrics, an attacker node can disrupt the protocol by manipulating coding opportunities. For example, by reporting very high coding opportunities, the attacker can improve its chances to be selected on the path and gain the control over the flow. Since coding opportunities not only depend on the network topology, but also depend on the current flow structure, it is more challenging to ensure the correctness of coding opportunities reported by a node than ensuring the correctness of pure topological metrics (*e.g.*, link or path qualities).

C. Summary and Defense Directions

Given the complexity of existing network coding systems and the numerous security vulnerabilities, it is a formidable task to design a defense scheme that proactively prevents all the aforementioned threats and is sufficiently lightweight such that the performance gain of network coding is preserved. A more promising direction is to design reactive defense schemes, where nodes monitor the network performance and take corrective actions only when abnormal conditions occur. However, in general, it is hard to distinguish the exact types of attacks, as the same attack effect may be caused by many different malicious actions. Another important challenge lies in the identification and isolation of attackers, because the epidemic nature of many attacks causes honest nodes affected by the attack to exhibit attacker-like behavior.

Instead of patching existing systems, a more fundamental approach is to design new network coding protocols with security consideration in the first place. Such protocols may be less optimized in performance compared to existing protocols, however, the security guarantees provided can make them attractive choices for adversarial environments.

V. Experimental Evaluations

As a proof of concept, we demonstrate the severity of security threats in network coding by evaluating the impact of packet dropping attacks.

Methodology. Our experiments are based on a representative intra-flow coding protocol, MORE [1], under the Glomosim [20] simulator. We enhance Glomosim to use a trace-driven physical layer based on traces from the real-world link quality measurements in MIT Roofnet [21], an experimental 802.11b/g mesh network of 38 nodes widely used in research papers [22], [23].

We use 802.11 MAC with 5.5Mbps raw bandwidth and 250 meter nominal range. We use the setup for MORE that gives optimal performance based on [1]: $\mathbb{F}_{2^8}$ as the finite field for network coding, the batch size is 32 packets, and the packet size is 1500 bytes.

For each experiment, we select a source node and a destination node at random. The throughput CDF graph we plot is generated from 100 random pairs of source and destination nodes.

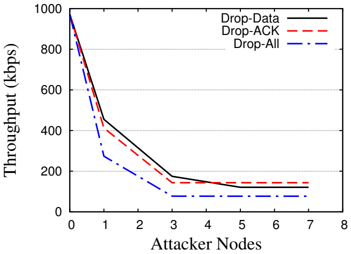

Fig. 2. Average throughput under multiple packet dropping attackers.

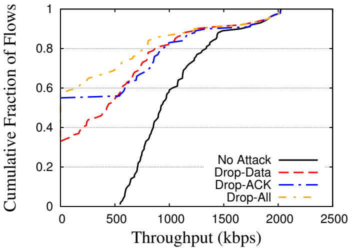

Fig. 3. Throughput CDF under single packet dropping attacker.

Attack Scenarios. We examine three different types packet dropping attacks.

- Drop-Data: only data packets are dropped
- Drop-ACK: only ACK packets are dropped
- Drop-All: both data and ACK packets are dropped

In all three types of attacks, the attacker nodes are selected at random among all forwarding nodes.

Results. Fig. 2 shows the average throughput as we vary the number of packet dropping attackers from 1 to 7. We see that the throughput drops rapidly as the number of attackers increases for all three attack types.

To further demonstrate the danger of the attack, Fig. 3 shows the throughput CDF for the case of a *single* attacker. Even with only a single attacker, a large percentage of flows experience zero throughput, around 35% for Drop-Data attack and over 50% for Drop-ACK and Drop-All attacks. Such cases occur when the single attacker node happens to be the critical forwarder node that every packet has to pass through. Therefore, contrary to the common belief of robustness of network coding systems, practical systems are actually quite fragile, a side effect of performance optimization algorithms that aim to reduce interference by minimizing the forwarder node set size and forwarding rates.

VI. Conclusion

Through detailed and systematic analysis of current network coding systems, we reveal that both intra-flow and inter-flow network coding systems are vulnerable to a wide range of attacks at various stages of the protocol. The use of coding techniques not only introduces new attacks, but also makes existing attacks more damaging and more challenging to defend against. Except for packet pollution, the security threats for network coding are largely unexplored, hence, provide exciting opportunities for security research.

Given the complexity of the existing systems, instead of attempting proactive solutions, designing reactive defense schemes is a more promising direction toward securing network coding systems. A more fundamental approach is to design new network coding systems with security in mind. With appropriate trade-offs between security and performance, such systems can be the most attractive choices in adversarial environments.

References

[1] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, “Trading structure for randomness in wireless opportunistic routing,” in *Proc. of ACM SIGCOMM ’07*, 2007.

[2] X. Zhang and B. Li, “Optimized multipath network coding in lossy wireless networks,” in *Proc. of ICDCS ’08*, 2008.

[3] X. Zhang and B. Li, “DICE: a game theoretic framework for wireless multipath network coding,” in *Proc. of Mobihoc 2008.*

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “Xors in the air: practical wireless network coding,” in *Proc. of ACM SIGCOMM ’06*, 2006.

[5] J. Le, J. C. S. Lui, and D. M. Chiu, “DCAR: Distributed coding-aware routing in wireless networks,” in *Proc. of ICDCS ’08*, 2008.

[6] S. Das, Y. Wu, R. Chandra, and Y. C. Hu, “Context-based routing: Technique, applications, and experience,” in *Proc. of NSDI ’08.*

[7] D. Charles, K. Jain, and K. Lauter, “Signatures for network coding,” in *Proc. of CISS ’06*, 2006.

[8] F. Zhao, T. Kalker, M. Medard, and K. Han, “Signatures for content distribution with network coding,” in *Proc. of ISIT ’07.*

[9] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, “An efficient signature-based scheme for securing network coding against pollution attacks,” in *Proceedings of INFOCOM 08*, 2008.

[10] M. Krohn, M. Freedman, and D. Mazieres, “On-the-fly verification of rateless erasure codes for efficient content distribution,” in *Proc. of Symposium on Security and Privacy*, 2004.

[11] C. Gkantsidis and P. Rodriguez Rodriguez, “Cooperative security for network coding file distribution,” *Proc. of INFOCOM 2006.*

[12] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, “Byzantine modification detection in multicast networks using randomized network coding,” in *Proc. of ISIT ’04.*

[13] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, “Resilient network coding in the presence of byzantine adversaries,” in *Proc. of INFOCOM ’07*, 2007.

[14] Q. Dong, J. Wu, W. Hu, and J. Crowcroft, “Practical network coding in wireless networks,” in *Proc. of MobiCom ’07*, 2007.

[15] J. Dong, R. Curtmola, and C. Nita-Rotaru, “On the pitfalls of using high-throughput multicast metrics in adversarial wireless mesh networks,” in *Proc. of SECON ’08*, June 2008.

[16] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Packet leashes: A defense against wormhole attacks in wireless ad hoc networks,” in *INFOCOM*, 2003.

[17] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, “Truelink: A practical countermeasure to the wormhole attack in wireless networks,” in *Proc. of ICNP ’06*, 2006.

[18] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *Proc. of MOBICOM*, August 2000.

[19] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: a secure on-demand routing protocol for ad hoc networks,” *Wirel. Netw.*, vol. 11, no. 1-2, pp. 21–38, 2005.

[20] “Global mobile information systems simulation library - glomosim,” http://pcl.cs.ucla.edu/projects/glomosim/.

[21] “MIT roofnet - publications and trace data.” http://pdos.csail.mit.edu/roofnet/doku.php?id=publications.

[22] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, P. Rodriguez, and S. Gheorghiu, “Multipath code casting for wireless mesh networks,” in *CoNEXT ’07*, 2007.

[23] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *Proc. of ACM MobiCom 2003.*