An algebraic watchdog for wireless network coding

An Algebraic Watchdog for Wireless Network Coding

MinJi Kim*, Muriel Médard*, João Barros†, and Ralf Kötter‡

*Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
Email: {minjikim, medard}@mit.edu

†Instituto de Telecommunicações
Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Engenharia da Universidade do Porto, Portugal
Email: jbarros@fe.up.pt

‡Institute for Communications Engineering
Technische Universität München, Munich, Germany

***Abstract*— In this paper, we propose a scheme, called the *algebraic watchdog* for wireless network coding, in which nodes can detect malicious behaviors probabilistically, police their downstream neighbors locally using overheard messages, and, thus, provide a secure global *self-checking network*. Unlike traditional Byzantine detection protocols which are *receiver-based*, this protocol gives the senders an active role in checking the node downstream. This work is inspired by Marti *et al.*'s *watchdog-pathrater*, which attempts to detect and mitigate the effects of routing misbehavior.**

As the first building block of a such system, we focus on a two-hop network. We present a graphical model to understand the inference process nodes execute to police their downstream neighbors; as well as to compute, analyze, and approximate the probabilities of misdetection and false detection. In addition, we present an algebraic analysis of the performance using an hypothesis testing framework, that provides exact formulae for probabilities of false detection and misdetection.

I. Introduction

There have been numerous contributions to secure wireless networks, including key management, secure routing, Byzantine detection, and various protocol designs (for a general survey on this topic, see [1]). We focus on Byzantine detection. The traditional approach is *receiver-based – i.e.* the receiver of the corrupted data detects the presence of an upstream adversary. However, this detection may come too late as the adversary is partially successful in disrupting the network (even if it is detected). It has wasted network bandwidth, while the source is still unaware of the need for retransmission.

Reference [2] introduces a protocol for routing wireless networks, called the *watchdog and pathrater*, in which upstream nodes police their downstream neighbors using *promiscuous monitoring*. Promiscuous monitoring means that if a node A is within range of a node B, it can overhear communication to and from B even if those communication do not directly involve A. This scheme successfully detects adversaries and removes misbehaving nodes from the network by dynamically adjusting the routing paths. However, the protocol requires a significant overhead (12% to 24%) owing to increased control traffic and numerous cryptographic messages.

Our goal is to design/analyze a watchdog-inspired protocol for wireless networks using network coding. Network coding [3][4] is advantageous as it not only increases throughput and robustness against failures and erasures but also it is resilient in dynamic/unstable networks where state information may change rapidly or may be hard to obtain. Taking advantage of the wireless setting, we propose a scheme for coded networks, in which nodes can verify probabilistically, and police their neighbors locally using promiscuous monitoring. Our ultimate goal is a robust *self-checking network*. In this paper, we present the first building block of a such system, and analyze the algebraic watchdog protocol for a two-hop network.

The paper is organized as follows. In Section II, we present the background and related material. In Section III, we introduce our problem statement and network model. In Section IV, we analyze the protocol for a simple two-hop network, first algebraically in Section IV-B and then graphically in Section IV-A. In Section V, we summarize our contribution and discuss some future work.

II. Background and Definitions

A. Secure Network Coding

Network coding, first introduced in [3], allows algebraic mixing of information in the intermediate nodes. This mixing has been shown to have numerous performance benefits. It is known that network coding maximizes throughput [3], as well as robustness against failures [4] and erasures [5]. However, a major concern for network coding system is its vulnerability to Byzantine adversaries. A single corrupted packet generated by a Byzantine adversary can contaminate all the information to a destination, and propagate to other destinations quickly. For

This material is based upon work under a subcontract #069145 issued by BAE Systems National Security Solutions, Inc. and supported by the DARPA and the Space and Naval Warfare System Center, San Diego under Contract No. N66001-08-C-2013.

‡Ralf Kötter passed away earlier this year.

978-1-4244-4313-0/09/$25.00 ©2009 IEEE

example, in random linear network coding [5], one corrupted packet in a generation (*i.e.* a fixed set of packets) can prevent a receiver from decoding any data from that generation even if all the other packets it has received are valid.

There are several papers that attempt to address this problem. One approach is to correct the errors injected by the Byzantine adversaries using *network error correction* [6]. They bound the maximum achievable rate in an adversarial setting, and generalizes the Hamming, Gilbert-Varshamov, and Singleton bounds. Jaggi *et al.*[7] propose a distributed, rate-optimal, network coding scheme for multicast network that is resilient in the presence of Byzantine adversaries for sufficiently large field and packet size. Reference [8] generalizes [7] to provide correction guarantees against adversarial errors for any given field and packet size. In [9], Kim *et al.*compare the cost and benefit associated with these Byzantine detection schemes in terms of transmitted bits by allowing nodes to employ the detection schemes to drop polluted data.

B. Secure Routing Protocol: Watchdog and Pathrater

The problem of securing networks in the presence of Byzantine adversaries has been studied extensively, e.g. [10],[11],[12]. The *watchdog and pathrater* [2] are two extensions to the Dynamic Source Routing [13] protocol that attempt to detect and mitigate the effects of routing misbehavior. The watchdog detects misbehavior based on promiscuous monitoring of the transmissions of the downstream node to confirm if this relay correctly forwards the packets it receives. If a node bound to forward a packet fails to do so after a certain period of time, the watchdog increments a failure rating for that node and a node is deemed to be misbehaving when this failure rating exceeds a certain threshold. The pathrater then uses the gathered information to determine the best possible routes by avoiding misbehaving nodes. This mechanism, which does not punish these nodes (it actually relieves them from forwarding operations), provides an increase in the throughput of networks with misbehaving nodes.

C. Hypothesis Testing

Hypothesis testing is a method of deciding which of the two hypotheses, denoted H_0 and H_1, is true, given an observation denoted as U. In this paper, H_0 is the hypothesis that R is well-behaving, H_1 is that R is malicious, and U is the information gathered from overhearing. The observation U is distributed differently depending whether H_0 or H_1 is true, and these distributions are denoted as $P_{U|H_0}$ and $P_{U|H_1}$ respectively.

An algorithm is used to choose between the hypotheses given the observation U. There are two types of error associated with the decision process:

- *Type 1 error, False detection*: Accepting H_1 when H_0 is true (*i.e.* considering a well-behaving R to be malicious), and the probability of this event is denoted γ.
- *Type 2 error, Misdetection*: Accepting H_0 when H_1 is true (*i.e.* considering a malicious R to be well-behaving), and the probability of this event is denoted β.

The Neyman-Pearson theorem gives the optimal decision rule that given the maximal tolerable β, we can minimize γ by accepting hypothesis H_0 if and only if $\log \frac{P_{U|H_0}}{P_{U|H_1}} \geq t$ for some threshold t dependant on γ. For more thorough survey on hypothesis testing in the context of authentication, see [14].

D. Notations and definitions

We shall use elements from a field, and their bit-representation. To avoid confusion, we use the same character in italic font (*i.e.* x) for the field element, and in bold font (*i.e.* $\mathbf{x}$) for the bit-representation. We use underscore bold font (*i.e.* $\underline{\mathbf{x}}$) for vectors. For arithmetic operations in the field, we shall use the conventional notation (*i.e.* $+, -, \cdot$). For bit-operation, we shall use $\oplus$ for addition, and $\otimes$ for multiplication.

We also require polynomial hash functions defined as follows (for a more detailed discussion on this topic, see [15]).

*Definition 1 (**Polynomial hash functions**):* For a finite field $\mathbf{F}$ and $d \geq 1$, the class of polynomial hash functions on $\mathbf{F}$ is defined as follows:

$$\mathcal{H}^d(\mathbf{F}) = \{h_a | a = \langle a_0, ..., a_d \rangle \in \mathbf{F}^{d+1}\},$$

where $h_a(x) = \sum_{i=0}^{d} a_i x^i$ for $x \in \mathbf{F}$.

III. Problem Statement

We model a wireless network with a hypergraph $G = (V, E_1, E_2)$, where V is the set of the nodes in the network, E_1 is the set of hyperedges representing the connectivity (wireless links), and E_2 is the set of hyperedges representing the interference. We use the hypergraph to capture the broadcast nature of the wireless medium. If $(v_1, v_2) \in E_1$ and $(v_1, v_3) \in E_2$ where $v_1, v_2, v_3 \in V$, then there is an intended transmission from v_1 to v_2, and v_3 can overhear this transmission (possibly incorrectly). There is a certain transition probability associated with the interference channels known to the nodes, and we model them with binary channels.

A node $v_i \in V$ transmits coded information x_i by transmitting a packet $\underline{\mathbf{p_i}}$, where $\underline{\mathbf{p_i}} = [\mathbf{a_i}, \mathbf{h_{I_i}}, \mathbf{h_{x_i}}, \mathbf{x_i}]$ is a $\{0,1\}$-vector. A valid packet $\underline{\mathbf{p_i}}$ is defined as below:

- $\mathbf{a_i}$ corresponds to the coding coefficients α_j, $j \in I_i$, where $I_i \subseteq V$ is the set of nodes adjacent to v_i in E_1,
- $\mathbf{h_{I_i}}$ corresponds to the hash $h(x_j)$, $v_j \in I_i$ where $h(\cdot)$ is a h-bit polynomial hash function,
- $\mathbf{h_{x_i}}$ corresponds to the polynomial hash $h(x_i)$,
- $\mathbf{x_i}$ is the n-bit representation of $x_i = \sum_{j \in I} \alpha_j x_j$.

We assume that the hash function used, $h(\cdot)$, is known to all nodes, including the adversary. In addition, we assume that $\mathbf{a_i}$, $\mathbf{h_{I_i}}$ and $\mathbf{h_{x_i}}$ are part of the header information, and are sufficiently coded to allow the nodes to correctly receive them even under noisy channel conditions. Therefore, if a node overhears the transmission of $\mathbf{p_i}$, it may not be able to correctly receive x_i, but it receives α_j and $h(x_j)$ for $v_j \in I_i$, and $h(x_i)$. Protecting the header sufficiently will of course induce some overhead, but the assumption remains a reasonable one to make. First, the header is smaller than the message itself. Second, even in the routing case, the header and the state information must to be coded sufficiently. Third, the hashes

$\mathbf{h_{I_i}}$ and $\mathbf{h_{x_i}}$ are contained within one hop – *i.e.* a node that receives $\underline{\mathbf{p_i}} = [\mathbf{a_i}, \mathbf{h_{I_i}}, \mathbf{h_{x_i}}, \mathbf{x_i}]$ does not need to repeat $\mathbf{h_{I_i}}$, thus sending only $\mathbf{h_{x_i}}$. Therefore, the overhead associated with the hashes is proportional to the in-degree of a node, and does not accumulate with the routing path length.

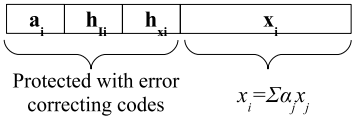

Fig. 1. A valid packet $\underline{\mathbf{p_i}}$ sent by well-behaving R

Assume that v_i transmits $\underline{\mathbf{p_i}} = [\mathbf{a_i}, \mathbf{h_{I_i}}, \mathbf{h_{x_i}}, \hat{\mathbf{x}}_\mathbf{i}]$, where $\hat{\mathbf{x}}_\mathbf{i} = \mathbf{x_i} \oplus \mathbf{e}$, $\mathbf{e} \in \{0,1\}^n$. If v_i is misbehaving, then $\mathbf{e} \neq 0$. It is important to note that the adversary can choose any $\mathbf{e}$; thus, the adversary can choose the message $\hat{\mathbf{x}}_\mathbf{i}$. Our goal is to detect with high probability when $\mathbf{e} \neq 0$. Note that even if $|\mathbf{e}|$ is small (*i.e.* the hamming distance between $\hat{\mathbf{x}}_\mathbf{i}$ and $\mathbf{x_i}$ is small), the algebraic interpretation of $\hat{\mathbf{x}}_\mathbf{i}$ and $\mathbf{x_i}$ may differ significantly. For example, consider $n = 4$, $\hat{\mathbf{x}}_\mathbf{i} = [0000]$, and $\mathbf{x_i} = [1000]$. Then, $\mathbf{e} = [1000]$ and $|\mathbf{e}| = 1$. However, the algebraic interpretation of $\hat{\mathbf{x}}_\mathbf{i}$ and $\mathbf{x_i}$ are 0 and 8. Thus, even a single bit flip can alter the message very significantly.

Our goal is to explore an approach to detect and prevent malicious behaviors in wireless networks using network coding. The scheme takes advantage of the wireless setting, where neighbors can overhear others' transmissions albeit with some noise, to verify probabilistically that the next node in the path is behaving given the overheard transmissions.

IV. Two-hop network

Consider a network (or a small neighborhood of nodes in a larger network) with nodes $v_1, v_2, ...v_m$, v_{m+1}, v_{m+2}. Nodes v_i, $i \in [1, m]$, want to transmit x_i to v_{m+2} via v_{m+1}. A single node v_i, $i \in [1, m]$, cannot check whether v_{m+1} is misbehaving or not even if v_i overhears $\mathbf{x_{m+1}}$, since without any information about x_j for $j \in [1, m]$, x_{m+1} is completely random to v_i. On the other hand, if v_i knows x_{m+1} and x_j for all $j \in [1, m]$, then v_i can verify that v_{m+1} is behaving with certainty; however, this requires at least $m-1$ additional reliable transmissions to v_i.

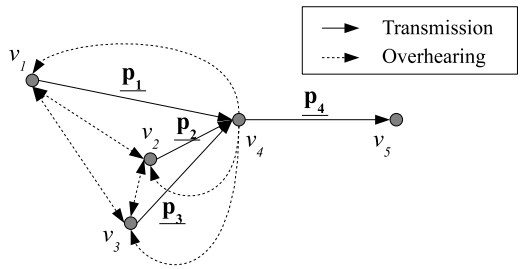

Fig. 2. A wireless network with $m = 3$.

Therefore, we take advantage of the wireless setting, in which nodes can overhear their neighbors' transmissions. In Figure 2, we use the solid lines to represent the intended channels E_1, and dotted lines for the interference channels E_2

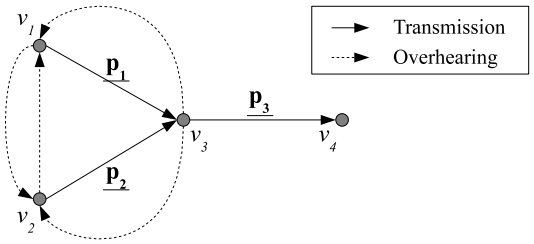

Fig. 3. A wireless network with $m = 2$.

which we model with binary channels as mentioned in Section III. Each node checks whether its neighbors are transmitting values that are consistent with the gathered information. If a node detects that its neighbor is misbehaving, then it can alert other nodes in the network and isolate the misbehaving node.

As outlined in Section II-C, we denote the hypothesis that R is well-behaving by H_0, and H_1 corresponds to that of a malicious R. In the next subsections, we shall use an example with $m = 2$, as shown Figure 3, to introduce the graphical model which explains how a node v_i checks its neighbor's behavior. Then, we use an algebraic approach to analyze/compute γ and β for this example network.

A. Graphical model approach

In this section, we present a graphical approach to model this problem systematically, and to explain how a node may check its neighbors. This approach may be advantageous as it lends easily to already existing graphical model algorithms as well as some approximation algorithms.

We shall consider the problem from v_1's perspective. As shown in Figure 4, the graphical model has four layers: Layer 1 contains 2^{n+h} vertices, each representing a bit-representation of $[\tilde{\mathbf{x}}_\mathbf{2}, \mathbf{h}(\mathbf{x_2})]$; Layer 2 contains 2^n vertices, each representing a bit-representation of $\mathbf{x_2}$; Layer 3 contains 2^n vertices corresponding to $\mathbf{x_3}$; and Layer 4 contains 2^{n+h} vertices corresponding to $[\tilde{\mathbf{x}}_\mathbf{3}, \mathbf{h}(\mathbf{x_3})]$. Edges exist between adjacent layers as follows:

- *Layer 1 to Layer 2:* An edge exists between a vertex $[\mathbf{v}, \mathbf{u}]$ in Layer 1 and a vertex $\mathbf{w}$ in Layer 2 if and only if $\mathbf{h}(\mathbf{w}) = \mathbf{u}$. The edge weight is normalized such that the total weight of edges leaving $[\mathbf{v}, \mathbf{u}]$ is 1, and the weight is proportional to:

  $$\mathbf{P}(\mathbf{v}|\text{ Channel statistics and } \mathbf{w} \text{ is the original message}),$$

  which is the probability that the inference channel outputs message $\mathbf{v}$ given an input message $\mathbf{w}$.
- *Layer 2 to Layer 3:* The edges represent a permutation. A vertex $\mathbf{v}$ in Layer 2 is adjacent to a vertex $\mathbf{w}$ in Layer 3 if and only if $w = c + \alpha_2 v$, where $c = \alpha_1 x_1$ is a constant, $\mathbf{v}$ and $\mathbf{w}$ are the bit-representation of v and w, respectively. The edge weights are all 1.
- *Layer 3 to Layer 4:* An edge exists between a vertex $\mathbf{v}$ in Layer 3 and a vertex $[\mathbf{w}, \mathbf{u}]$ in Layer 4 if and only if $\mathbf{h}(\mathbf{v}) = \mathbf{u}$. The edge weight is normalized such that the total weight leaving $\mathbf{v}$ is 1, and is proportional to:

  $$\mathbf{P}(\mathbf{w}|\text{ Channel statistics and } \mathbf{v} \text{ is the original message}).$$

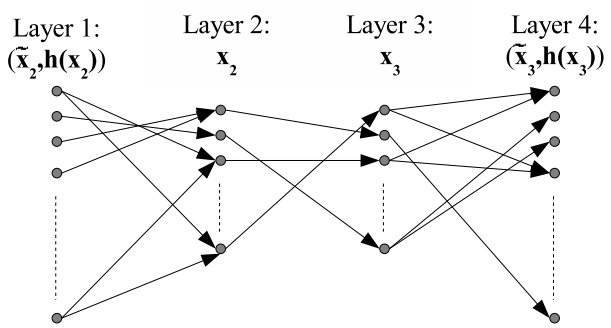

Fig. 4. A graphical model from v_1's perspective

Node v_1 overhears the transmissions from v_2 to v_3 and from v_3 to v_4; therefore, it receives $[\tilde{\mathbf{x}}_\mathbf{2}, \mathbf{h}(\mathbf{x_2})]$ and $[\tilde{\mathbf{x}}_\mathbf{3}, \mathbf{h}(\mathbf{x_3})]$, corresponding to the *starting point* in Layer 1 and the *destination point* in Layer 4 respectively. By computing the sum of the product of the weights of all possible paths between the starting and the destination points, v_1 computes the probability that v_3 is consistent with the information gathered.

This graphical model illustrates sequentially and visually the inference process v_1 executes. In addition, the graphical approach may be extend to larger networks. Cascading multiple copies of the graphical model may allow us to systematically model larger networks with multiple hops as well as $m \geq 3$. (Note that when m increases, the graphical model changes into a family of graphs; while when n increases, the size of each Layer increases.) Furthermore, by using approximation algorithms and pruning algorithms, we may be able to simplify the computation as well as the structure of the graph.

B. Algebraic approach

Consider v_1. By assumption, v_1 correctly receives $\mathbf{a_2}$, $\mathbf{a_3}$, $\mathbf{h_{I_2}}$, $\mathbf{h_{I_3}}$, $\mathbf{h_{x_2}}$, and $\mathbf{h_{x_3}}$. In addition, v_1 receives $\tilde{\mathbf{x}}_\mathbf{2} = \mathbf{x_2} + \mathbf{e}'$ and $\tilde{\mathbf{x}}_\mathbf{3} = \mathbf{x_3} + \mathbf{e}''$, where $\mathbf{e}'$ and $\mathbf{e}''$ are outcomes of the interference channels. Given $\tilde{\mathbf{x}}_\mathbf{j}$ for $j = \{2, 3\}$ and the transition probabilities, v_1 computes $r_{j\to 1}$ such that the sum of the probability that the interference channel from v_j and v_1 outputs $\tilde{\mathbf{x}}_\mathbf{j}$ given $\mathbf{x} \in B(\tilde{\mathbf{x}}_\mathbf{j}, r_{j\to 1})$ is greater or equal to $1-\epsilon$ where ϵ is a constant, and $B(\mathbf{x}, r)$ is a n-dimensional ball of radius r centered at $\mathbf{x}$. Now, v_1 computes $\tilde{X}_j = \{\mathbf{x} \mid h(x) = h(x_j)\} \cap B(\tilde{\mathbf{x}}_\mathbf{j}, r_{j\to 1})$ for $j = \{2, 3\}$. Then, v_1 computes $\alpha_1 x_1 + \alpha_2 \hat{x}$ for all $\hat{\mathbf{x}} \in \tilde{X}_2$. Then, v_1 intersects $\tilde{X}_3$ and the computed $\alpha_1 x_1 + \alpha_2 \hat{x}$'s. If the intersection is empty, then v_1 claims that R is misbehaving.

We explain the inference process described above using the graphical model introduced in Section IV-A. The set $\{\mathbf{x} \mid h(x) = h(x_2)\}$ represents the Layer 2 vertices reachable from the starting point ($[\tilde{\mathbf{x}}_\mathbf{2}, \mathbf{h}(\mathbf{x_2})]$ in Layer 1), and $\tilde{X}_2$ is a subset of the reachable Layer 2 vertices such that the total edge weight (which corresponds to the transition probability) from the starting point is greater than $1-\epsilon$. Then, computing $\alpha_1 x_1 + \alpha_2 \hat{x}$ represents the permutation from Layers 2 to 3. Finally, the intersection with $\tilde{X}_3$ represents finding a set of Layer 3 vertices such that they are adjacent to the destination point ($[\tilde{\mathbf{x}}_\mathbf{3}, \mathbf{h}(\mathbf{x_3})]$ in Layer 4) and their total transition probability to the destination point is greater than $1-\epsilon$.

Note that a malicious v_3 would not inject errors in $\mathbf{h_{x_3}}$ only, because the destination v_4 can easily verify if $\mathbf{h_{x_3}}$ is equal to $h(\mathbf{x_3})$. Therefore, $\mathbf{h_{x_3}}$ and $\mathbf{x_3}$ are consistent. In addition, v_3 would not inject errors in $\mathbf{h_{x_j}}$, $j \in I_3$, as each node v_j can verify the hash of its message. On the other hand, a malicious v_3 can inject errors in $\mathbf{a_3}$, forcing v_4 to receive incorrect coefficients $\tilde{\alpha}_j$'s instead of α_j's. However, any error introduced in $\mathbf{a_3}$ can be translated to errors in $\mathbf{x_3}$ by assuming that $\tilde{\alpha}_j$'s are the correct coding coefficients. Therefore, we are concerned only with the case in which v_3 introduces errors in $\mathbf{x_3}$ (and therefore, in $\mathbf{h_{x_3}}$ such that $\mathbf{h_{x_3}} = h(\mathbf{x_3})$).

Lemma 4.1: For n sufficiently large, the probability of false detection, $\gamma \leq \epsilon$ for any arbitrary small constant ϵ.

Proof: Assume that v_3 is not malicious, and transmits $\mathbf{x_3}$ and $\mathbf{h_{x_3}}$ consistent with v_4's check. Then, for n sufficiently large, v_1 can choose $r_{2\to 1}$ and $r_{3\to 1}$ such that the probability that the bit representation of $x_3 = \alpha_1 x_1 + \alpha_2 x_2$ is in $\tilde{X}_3$ and the probability that $\mathbf{x_2} \in \tilde{X}_2$ are greater than $1-\epsilon$. Therefore, $\tilde{X}_3 \cap \{\alpha_1 x_1 + \alpha_2 \hat{x} \mid \forall \hat{\mathbf{x}} \in \tilde{X}_2\} \neq \emptyset$ with probability arbitrary close to 1. Therefore, a well-behaving v_3 passes v_1's check with probability at least $1-\epsilon$. Thus, $\gamma \leq \epsilon$. ■

Lemma 4.2: $\mathbf{P}$(A malicious v_3 is undetected from v_1's perspective) is:

$$\min\left\{1, \frac{\sum_{k=0}^{r_{1\to 2}} \binom{n}{k}}{2^{(h+n)}} \cdot \frac{\sum_{k=0}^{r_{2\to 1}} \binom{n}{k}}{2^{(h+n)}} \cdot \frac{\sum_{k=0}^{r_{3\to 1}} \binom{n}{k}}{2^h}\right\}.$$

Proof: Assume that v_3 is malicious and injects errors into $\mathbf{x_3}$. Consider an element $\mathbf{z} \in \tilde{X}_3$, where $z = \alpha_1 x_1 + \alpha_2 x_2 + e = \alpha_1 x_1 + \alpha_2 (x_2 + e_2)$ for some e and e_2. Note that, since we are using a field of size 2^n, multiplying an element from the field by a randomly chosen constant has the effect of randomizing the product. Here, we consider two cases:

- *Case 1:* If $x_2 + e_2 \notin \tilde{X}_2$, then v_3 fails v_1's check.
- *Case 2:* If $x_2 + e_2 \in \tilde{X}_2$, then v_3 passes v_1's check; however, v_3 is unlikely to pass v_2's check. This is because $\alpha_1 x_1 + \alpha_2 (x_2 + e_2) = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_2 e_2 = \alpha_1 (x_1 + e_1) + \alpha_2 x_2$ for some e_1. Here, for uniformly random α_1 and α_2, e_1 is also uniformly random. Therefore, the probability that v_3 will pass is the probability that the uniformly random vector $x_1 + e_1$ belongs to $\tilde{X}_1 = \{x \mid h(x) = h(x_1)\} \cap B(\tilde{\mathbf{x}}_\mathbf{1}, r_{1\to 2})$ where v_2 overhears $\tilde{\mathbf{x}}_\mathbf{1}$ from v_1, and the probability that the interference channel from v_1 to v_2 outputs $\tilde{\mathbf{x}}_\mathbf{1}$ given $\mathbf{x} \in B(\tilde{\mathbf{x}}_\mathbf{1}, r_{1\to 2})$ is greater than $1-\epsilon$.

$$\mathbf{P}(\text{A malicious } v_3 \text{ passes } v_2\text{'s check}) = \mathbf{P}(x_1 + e_1 \in \tilde{X}_1) = \frac{Vol(\tilde{X}_1)}{2^n},$$

where $Vol(\cdot)$ is equal to the number of $\{0,1\}$-vectors in the given set. Since $Vol(B(x,r)) = \sum_{k=0}^{r} \binom{n}{k} \leq 2^n$, and the probability that $h(x)$ is equal to a given value is $\frac{1}{2^h}$, $Vol(\tilde{X}_1)$ is given as follows:

$$Vol(\tilde{X}_1) = \frac{Vol(B(\tilde{x}_1, r_{1\to 2}))}{2^h} = \frac{\sum_{k=0}^{r_{1\to 2}} \binom{n}{k}}{2^h}.$$

From v_1's perspective, the probability that a $\mathbf{z} \in \tilde{X}_3$ passes the checks, $\mathbf{P}(\mathbf{z}$ passes check), is:

$$0 \cdot \mathbf{P}(x_2 + e_2 \notin \tilde{X}_2) + \frac{\sum_{k=0}^{r_{1\to 2}} \binom{n}{k}}{2^{(h+n)}} \cdot \mathbf{P}(x_2 + e_2 \in \tilde{X}_2).$$

Similarly, $\mathbf{P}(x_2 + e_2 \in \tilde{X}_2) = \frac{\sum_{k=0}^{r_{2\to 1}} \binom{n}{k}}{2^{(h+n)}}$, and $Vol(\tilde{X}_3) = \frac{\sum_{k=0}^{r_{3\to 1}} \binom{n}{k}}{2^h}$. Then, the probability that v_3 is undetected from v_1's perspective is the probability that *at least one* $\mathbf{z} \in \tilde{X}_3$ passes the check:

$$\begin{aligned}&\mathbf{P}(\text{A malicious } v_3 \text{ is undetected from } v_1\text{'s perspective})\\ &\quad = \min\{1, \mathbf{P}(\mathbf{z} \text{ passes check}) \cdot Vol(\tilde{X}_3)\}\end{aligned}$$

Note that $\mathbf{P}(\mathbf{z}$ passes check$) \cdot Vol(\tilde{X}_3)$ is the expected number of $\mathbf{z} \in \tilde{X}_3$ that passes the check; thus, given a high enough $\mathbf{P}(\mathbf{z}$ passes check$)$, would exceed 1. Therefore, we take $\min\{1, \mathbf{P}(\mathbf{z}$ passes check$) \cdot Vol(\tilde{X}_3)\}$ to get a valid probability. This proves the statement. ■

Lemma 4.3: $\mathbf{P}$(A malicious v_3 is undetected from v_2's perspective) is:

$$\min\left\{1, \frac{\sum_{k=0}^{r_{1\to 2}} \binom{n}{k}}{2^{(h+n)}} \cdot \frac{\sum_{k=0}^{r_{2\to 1}} \binom{n}{k}}{2^{(h+n)}} \cdot \frac{\sum_{k=0}^{r_{3\to 2}} \binom{n}{k}}{2^h}\right\},$$

where v_2 overhears $\tilde{\mathbf{x}}_\mathbf{3}$ from v_3, and the probability that the interference channel from v_3 to v_2 outputs $\tilde{\mathbf{x}}_\mathbf{3}$ given $\mathbf{x} \in B(\tilde{\mathbf{x}}_\mathbf{3}, r_{3\to 2})$ is greater than $1 - \epsilon$.

Proof: By similar analysis as in proof of Lemma 4.2. ■

Theorem 4.4: The probability of misdetection, β, is:

$$\beta = \min\left\{1, \frac{\sum_{k=0}^{r_{1\to 2}} \binom{n}{k}}{2^{(h+n)}} \cdot \frac{\sum_{k=0}^{r_{2\to 1}} \binom{n}{k}}{2^{(h+n)}} \cdot \frac{1}{2^h} \sum_{k=0}^{r} \binom{n}{k}\right\},$$

where $r = \min\{r_{3\to 1}, r_{3\to 2}\}$.

Proof: The probability of misdetection is the minimum of the probability that v_1 and v_2 misdetecting malicious v_3. Therefore, by Lemma 4.2 and 4.3, the statement is true. ■

Theorem 4.4 shows that the probability of misdetection β decreases with the hash size, as the hashes restrict the space of consistent codewords. In addition, since $r_{1\to 2}$, $r_{2\to 1}$, $r_{3\to 1}$, and $r_{3\to 2}$ represent the uncertainty introduced by the interference channels, β increases with them. Lastly and the most interestingly, β decreases with n, since $\sum_{k=0}^{r} \binom{n}{k} < 2^n$ for $r < n$. This is because network coding randomizes the messages over a field whose size is increasing exponentially with n, and this makes it difficult for an adversary to introduce errors without introducing inconsistencies.

Note that we can apply Theorem 4.4 even when v_1 and v_2 cannot overhear each other. In this case, both $r_{1\to 2}$ and $r_{2\to 1}$ equal to n, giving the probability of misdetection, $\beta = \min\{1, \sum_{k=0}^{r} \binom{n}{k}/8^h\}$ where $r = \min\{r_{3\to 1}, r_{3\to 2}\}$. Here, β highly depends on h, the size of the hash, as v_1 and v_2 are only using their own message and the overheard hashes.

The algebraic approach results in a nice analysis with exact formulae for γ and β. In addition, these formulae are conditional probabilities; as a result, they hold regardless of a priori knowledge of whether v_3 is malicious or not. However, this approach is not very extensible as the number of "reasonable" messages grows exponentially with m.

V. Conclusion and Future Work

We proposed a scheme, the *algebraic watchdog* for coded networks, in which nodes can verify their neighbors probabilistically and police them locally by means of overheard messages. We presented a graphical model for two-hop networks to explain how a node checks its neighbors; as well as compute, analyze, and potentially approximate the probabilities of misdetection/false detection. We also provided an algebraic analysis of the performance using an hypothesis testing framework, which gives exact formulae for the probabilities.

Our ultimate goal is to design a network in which the participants check their neighborhood locally to enable a secure global network - *i.e.* a self-checking network. There are several avenues for future work, of which we shall list only a few. First, there is a need to develop models and frameworks for the algebraic watchdog in general topology as well as multi-hop networks. In addition, possible future work includes developing inference methods and approximation algorithms for nodes to decide efficiently whether they believe their neighbor is malicious or not.

References

[1] J.-P. Hubaux, L. Buttyán, and S. Capkun, "The quest for security in mobile ad hoc networks," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2001, pp. 146–155.

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

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, 2000.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transaction on Networking*, vol. 11, pp. 782–795, 2003.

[5] D. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20.

[6] R. W. Yeung and N. Cai, "Network error correction," *Communications in Information and Systems*, no. 1, pp. 19–54, 2006.

[7] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of Byzantine adversaries," in *Proceedings of IEEE INFOCOM*, March 2007, pp. 616 – 624.

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

[9] M. Kim, M. Médard, and J. Barros, "Countering Byzantine adversaries with network coding: An overhead analysis," in *Proceedings of MILCOM*, 2008.

[10] R. Perlman, "Network layer protocols with Byzantine robustness," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, October 1988.

[11] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Symposium on Operating Systems Design and Implementation (OSDI)*, February 1999.

[12] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, pp. 382–401, 1982.

[13] D. B. Johnson, "Routing in ad hoc networks of mobile hosts," in *Proceedings of the Workshop on Mobile Computing Systems and Applications*, 1994, pp. 158–163.

[14] U. M. Maurer, "Authentication theory and hypothesis testing," *IEEE Transaction on Information Theory*, vol. 46, pp. 1350–1356, 2000.

[15] M. Dietzfelbinger, J. Gil, Y. Matias, and N. Pippenger, "Polynomial hash functions are reliable," in *Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, vol. 623. Springer-Verlag, 1992, pp. 235–246.