On Counteracting Byzantine Attacks in Network Coded Peer-to-Peer Networks

On Counteracting Byzantine Attacks in Network Coded Peer-to-Peer Networks

MinJi Kim, Luísa Lima, Fang Zhao, João Barros, Muriel Médard,
Ralf Koetter, Ton Kalker, Keesook J. Han

***Abstract*—Random linear network coding can be used in peer-to-peer networks to increase the efficiency of content distribution and distributed storage. However, these systems are particularly susceptible to Byzantine attacks. We quantify the impact of Byzantine attacks on the coded system by evaluating the probability that a receiver node fails to correctly recover a file. We show that even for a small probability of attack, the system fails with overwhelming probability. We then propose a novel signature scheme that allows packet-level Byzantine detection. This scheme allows one-hop containment of the contamination, and saves bandwidth by allowing nodes to detect and drop the contaminated packets. We compare the net cost of our signature scheme with various other Byzantine schemes, and show that when the probability of Byzantine attacks is high, our scheme is the most bandwidth efficient.**

***Index Terms*—Network coding, Byzantine, security, peer to peer, distributed storage, content distribution.**

I. Introduction

Network coding [1], an alternative to the traditional forwarding paradigm, allows algebraic mixing of packets in a network. It maximizes throughput for multicast transmissions [2], [3], [4], robustness against failures [5] and erasures [6]. Random linear network coding (RLNC), in which nodes independently take random linear combination of the packets, is sufficient for multicast networks [7], and is suitable for dynamic/unstable networks, such as peer-to-peer (P2P) networks [8], [9].

A P2P network is a cooperative network in which storage and bandwidth resources are shared in a distributed architecture. This is a cost-effective and scalable way to distribute content to a large number of receivers. One such architecture is the BitTorrent system [10], which splits large files into small blocks. After a node downloads a block, it acts as a source for that particular block. The main challenges in these systems are the scheduling and management of rare blocks.

As an alternative to current strategies for these challenges, [8], [9] propose the use of RLNC to increase the efficiency of content distribution in a P2P solution. These schemes are completely distributed and eliminate the need of a scheduler, since each node independently forwards a random linear combination. In addition, there is a high probability that each packet a node receives is linearly independent of the previous ones, and thus, the problem of redundancy caused by the flooding approaches in traditional P2P networks is reduced. RLNC based schemes significantly reduce the downloading time and improve the robustness of the system [8], [11].

Despite their desirable properties, network coded P2P systems are particularly susceptible to *Byzantine attacks* [12], [13], [14] – the injection of corrupted packets into the information flow. Since network coding relies on mixing of packets, a single corrupted packet may easily corrupt the entire information flow [15], [16]. Furthermore, in P2P networks, there is typically no security control over the nodes that join the network and the packets that they redistribute. The topologies of the overlay graphs that arise from traditional P2P networks are often modeled as scale-free and small-world networks [17], [18], which are prone to the dissemination of epidemics, such as worms and viruses [19], [20]. Several authors address these problems in coded P2P networks. We shall discuss these countermeasures in Section II. Most of these can be divided into two main categories: (i) end-to-end error correction and (ii) misbehavior detection.

Motivated by these observations, we address the issues of Byzantine adversaries in coded P2P networks. This paper is based on work from [21], [22], [23]. The main contributions of this paper are as follows:

- We propose a model for the evaluation of the impact of Byzantine attacks in coded P2P networks, and provide analytical results which show that, even for a small probability of attack, the information can become contaminated with overwhelming probability.
- We propose a new efficient, packet-based signature scheme, designed specifically for RLNC systems, to

Manuscript received on April 14, 2009, and revised on December 20, 2009. This work was partially presented at IEEE ISIT 2007 (Nice, France) titled "Signatures for content distribution with network coding", at IEEE MILCOM 2008 (San Diego, USA) titled "Countering Byzantine Adversaries with Network Coding: An Overhead Analysis", and at IEEE ISIT 2009 (Seoul, Korea) titled "Byzantine Attacks against Network Coding in Peer to Peer Distributed Storage".

M. Kim, F. Zhao and M. Médard ({minjikim, zhaof, medard}@mit.edu) are with the Research Laboratory of Electronics at the Massachusetts Institute of Technology, MA USA. L. Lima (luisalima@dcc.fc.up.pt) is with the Instituto de Telecomunicações, Department of Computer Science, Faculdade de Ciências, Universidade do Porto, Portugal. J. Barros (jbarros@fe.up.pt) is with the Instituto de Telecomunicações, Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Portugal. R. Koetter was with the Institute for Communications Engineering of the Technischen Universitaet Muenchen, Germany. R. Koetter passed away recently in February 2009. T. Kalker (ton.kalker@hp.com) is with the Hewlett-Packard Laboratories, CA USA. K. Han (keesook.Han@rl.af.mil) is with Air Force Research Laboratory, NY USA.

This work was partially funded by the National Science Foundation under grants "ITR: Network Coding - From Theory to Practice" (CCR-0325496), "XORs in the Air: Practical Wireless Network Coding" (CNS-0627021), the Air Force Office of Scientific Research (AFOSR) under grant FA9550-06-1-0155, the DARPA and the Space and Naval Warfare System Center, San Diego under Contract No. N66001-08-C-2013, and subcontract #069145 issued by BAE Systems National Security Solutions, Inc. In addition, this work was partially funded by the Luso-American Foundation and the Fundação para a Ciência e Tecnologia under the MIT Portugal program, grant MIT-Pt/TS-ITS/0059/2008, the Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) under grant SFRH/BD/24718/2005, and the European Community under grant FP7-INFSO-ICT-215252 (N-Crave Project).

detect Byzantine attacks by checking the membership of a received packet in the valid vector space. This scheme allows an one-hop containment of the contamination.
- We analyze the overhead in terms of bandwidth associated with our signature scheme, and compare it to that of various Byzantine detection schemes. We also show that our scheme is the most bandwidth efficient if the probability of attack is high.

This paper is organized as follows. Section II gives an overview of network coding in P2P networks and existing Byzantine detection schemes. In Section III, we analyze the impact of Byzantine attacks on the system. We propose our signature scheme in Section IV, and compare its overhead with other schemes in Section V. We conclude in Section VI.

II. Background

A. Network coding in P2P networks

References [6], [7] propose a *random block linear network coding system* – a simple, practical capacity-achieving code, in which nodes independently construct their linear code randomly. In such a system, a source generates information in batches of G packets (called a *generation*). The source then multicasts to its destination nodes using RLNC, where only the packets from the same generation are mixed. Note that RLNC is a distributed protocol, which requires no state information; thus, making it suitable for dynamic and unstable networks.

Several authors have evaluated the performance of network coding in P2P networks. Gkantsidis *et al.* [9] propose a scheme for content distribution of large files in which nodes make forwarding decisions solely based on local information. This scheme improves the expected file download time and the robustness of the system. Reference [8] compares the performance of network coding with traditional coding measures in a distributed storage setting with very limited storage space with the goal of minimizing the storage locations a file-downloader connects to. They show that RLNC performs well without a large amount of additional storage space. Dimakis *et al.* [24] introduce a graph-theoretic framework for P2P distributed system, and show that RLNC minimizes the required bandwidth to maintain the distributed storage architectures.

B. Byzantine detection scheme for network coded systems

1) End-to-end error correction scheme: Reference [25] introduces *network error correction* for coded systems. They bound the maximum achievable rate in an adversarial setting, and generalize the Hamming, Gilbert-Varshamov, and Singleton bounds. Jaggi *et al.* [15] introduce the first distributed polynomial-time rate-optimal network codes that work in the presence of Byzantine nodes and are information-theoretically secure. The adversarial nodes are viewed as a secondary source. The source adds redundancy to help the receivers distill out the source information from the received mixtures. This work is generalized in [26], [27].

2) Generation-based Byzantine detection scheme: Ho *et al.* [28] introduce an information-theoretic approach for detecting Byzantine adversaries, which only assumes that the adversary did not see all linear combinations received by the receivers. Their detection probability varies with the length of the hash, field size, and the amount of information unknown to the adversary. A polynomial hash is added to each packet in the generation. Once the destination node receives enough packets to decode a generation, it can probabilistically detect errors. The intuition behind this scheme is that if a packet is valid, then its data and hash are consistent with its coding vector; and a linear combination of valid packets is also valid.

3) Packet-based Byzantine detection scheme: There are several signature schemes that have been presented in the literature. For instance, [29] proposes a signature scheme for network coding based on Weil pairing on elliptic curves. Elliptic curves are hard to analyze and are known to be computationally expensive [30]. The experimental results in [31] show that this scheme is indeed costly and time-consuming.

Reference [32] uses homomorphic hash functions to verify packets in P2P systems, and [16] extends this approach to secure network coded P2P systems against Byzantine attacks. However, [16] requires a secure channel to transmit the hashes to all receivers before data is delivered. In this paper, we assume that no such secure channel is available.

Reference [31] proposes a homomorphic signature scheme with RSA encryption/decryption to allow authentication and verification of data. Unfortunately, the scheme is incorrect[1]. This homomorphic property does not hold due to an error in the second to last equation in (12) of [31]; that is:

$$(a \bmod p) \times (b \bmod p) \bmod r \neq (ab \bmod p) \bmod r.$$

In this paper, we propose a new homomorphic signature scheme, which is both efficient and does not require a secure channel.

III. Impact of Byzantine attacks on P2P networks

In Section III-A, we introduce our model for evaluating the probability of a distributed denial of service attack (DDoS) caused by Byzantine nodes in a P2P network. We then present analytic results for two distinct scenarios in Section III-B, and a qualitative interpretation of the results in Section III-C.

A. Model

We consider a directed graph with a set of nodes $\mathcal{N}$. A *source* node has a large file to be sent to *receiver* nodes. The file is divided into m packets. The source connects to a subset of nodes, $\mathcal{N}_s \subseteq \mathcal{N}$, chosen uniformly at random, and sends each of them a different random linear combination of the original packets. To ensure that enough degrees of freedom exist in the network, $|\mathcal{N}_s| \geq m$. We refer to the nodes in $\mathcal{N}_s$ as *level-s* nodes. A *tracker* keeps track of the list of *informed nodes* $N(t)$, *i.e.*, nodes that keep information packets.

For a receiver to retrieve the file, it connects to a subset of nodes $\mathcal{N}_r \subseteq \mathcal{N}$, chosen uniformly at random, with $|\mathcal{N}_r| \geq |\mathcal{N}_s|$. We refer to the nodes in $\mathcal{N}_r$ as *level-r* nodes. Note that there may be an overlap between level-s and level-r. In each

[1]This fact has been communicated to the authors of [31] by Anthony E. Kim, Raluca Ada Popa, and Muriel Médard, and acknowledged by the authors.

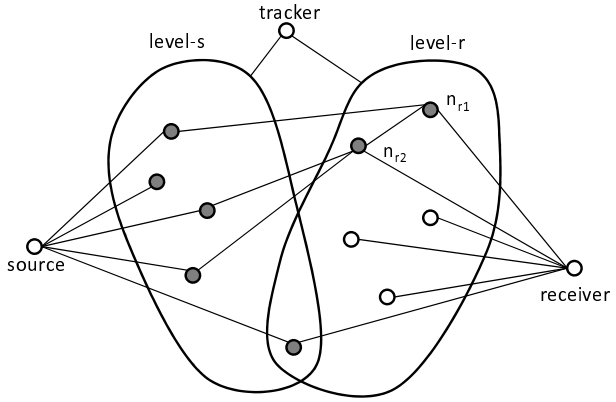

Fig. 1. Network model. The source is connected to the level-s nodes, and the receiver is connected to the level-r nodes. The dark nodes are the informed nodes. The level-r nodes take turns to contact the tracker to connects to $|D| = 2$ level-s nodes based on the list returned by the tracker. Here, nodes n_{r1} and n_{r2} have completed this process, and the other level-r nodes have not.

time slot, one of the uninformed level-r nodes, $n \in \mathcal{N}_r \setminus \mathcal{N}_s$, contacts the tracker to retrieve a random list of d informed nodes, where $d < |\mathcal{N}_s|$. The node n then connects to these informed nodes through a secure overlay connection, retrieves their packets, and stores a single random linear combination of these packets. During the same time slot, the tracker updates its list of informed nodes to $N(t) \cup \{n\}$. This process is repeated for all nodes in $\mathcal{N}_r \setminus \mathcal{N}_s$, and then all level-r nodes forward their stored packets to the receiver. In order to maximize the probability of storing linearly independent combinations in level-r nodes and ensure decodability at the receiver, we set $d \geq 2$. Although we assume that each node in level-s and level-r stores only one packet, the model can be easily generalized to account for higher numbers. An example of this network model is shown in Figure 1. Note that the tracker is considered to be a trusted party in our model – in fact, as in the case of most P2P protocols, a dishonest tracker would yield a protocol failure with overwhelming probability.

We define an *Information Contact Graph* $G(t) = \{N(t), A(t)\}$ to denote the evolving graph formed in the above process, where $N(t)$ is the list of informed nodes and $A(t)$ is the set of overlay links that connect the level-s and level-r nodes. The probability that a node becomes a Byzantine attacker is p_b. An attacker corrupts the packet it stores by generating arbitrary content while complying to the standard packet format. A node independently decides whether it becomes Byzantine at the start of the file dissemination process according to p_b and stays that way throughout the process. We define an indicator variable $I_b(n)$ which is 1 if node n is Byzantine and 0 otherwise. The tracker has no information about which nodes are Byzantine. A *contaminated packet* is a packet that is either directly corrupted by an attacker, or is a linear combination that involves at least one contaminated packet. A *contaminated node* is a node that stores a contaminated packet. The *blocking probability* Ψ is the probability that the receiver collects at least one contaminated packet, and thus, is unable to decode the file. This is equivalent to the attacker successfully carries out a DDoS attack.

B. Analysis of Impact of Byzantine Attacks

We now evaluate the *blocking probability* at the receiver. We then consider the *expected number of contaminated nodes* at any given time. First, we introduce necessary definitions, as follows. We define an indicator variable $I_c(t, n)$ which is equal to 1 if node n is contaminated at time t and 0 otherwise. $C(t)$ is a random variable for the number of contaminated nodes in $N(t)$, and $\overline{C}(t) = |N(t)| - C(t)$ is the number of uncontaminated nodes. The function $B(k, n, p)$ denotes the binomial distribution where

$$B(k, n, p) = \binom{n}{k} p^k (1-p)^{(n-k)}.$$

The function $h(k; N, m, n)$ denotes the hypergeometric distribution, in which

$$h(k; N, m, n) = \binom{m}{k}\binom{N-m}{n-k} / \binom{N}{m}.$$

Let N_b denote the number of informed Byzantine nodes at time $t = 0$, that is, the number of Byzantine nodes in $\mathcal{N}_s$. N_b has a binomial distribution with parameters $(|\mathcal{N}_s|, p_b)$.

We consider two scenarios. In *Theorem 1*, for simplicity, we consider a static informed nodes list, in which the list kept by the tracker is fixed to $\mathcal{N}_s$. In this case, level-r nodes only connect to level-s nodes. Second, in *Theorem 2*, we generalize to the case in which the tracker updates its list of informed nodes to $N(t)$, as stated in Section III-A.

Theorem 1 (Static Informed Nodes List): Let $G(t)$ be an information contact graph in which nodes in $\mathcal{N}_r$ only connect to nodes in $\mathcal{N}_s$. Then its blocking probability Ψ is given by

$$\Psi = 1 - \sum_{y=0}^{|\mathcal{N}_s|} h(y; |\mathcal{N}|, |\mathcal{N}_s|, |\mathcal{N}_r|) \left(\sum_{i=0}^{|\mathcal{N}_s|} B(i, |\mathcal{N}_s|, p_b) f(i, y) \right),$$

$$f(i, y) = \left(1 - \frac{i}{|\mathcal{N}_s|}\right)^y \left[(1-p_b) h(0, |\mathcal{N}_s|, i, d)\right]^{|\mathcal{N}_r|-y}.$$

Proof: We consider two disjoint subsets of $\mathcal{N}_r$: the set of informed nodes at $t = 0$, that is, $\mathcal{N}_r \cap \mathcal{N}_s$, and the uninformed nodes, that is, $\mathcal{N}_r \setminus \mathcal{N}_s$. Let Y be a random variable for the number of nodes in $\mathcal{N}_r \cap \mathcal{N}_s$. Y has a hypergeometric distribution, $P(Y = y) = h(y; |\mathcal{N}|, |\mathcal{N}_s|, |\mathcal{N}_r|)$.

We first consider $n \in \mathcal{N}_r \cap \mathcal{N}_s$. Given $N_b = i$ and $Y = y$, the probability that n is uncontaminated is equal to the probability that it is not initially Byzantine, which is equal to $1 - i/|\mathcal{N}_s|$. Then, the probability that all nodes in $\mathcal{N}_r \cap \mathcal{N}_s$ are uncontaminated is $\left(1 - \frac{i}{|\mathcal{N}_s|}\right)^y$.

Now, at each timeslot $t > 0$, a node $n \in \mathcal{N}_r \setminus \mathcal{N}_s$ becomes informed. For n to be uncontaminated, it must not be Byzantine and it must connect to d uncontaminated nodes. Then,

$$P(I_c(t, n) = 0 | N_b = i, Y = y) = (1-p_b) h(0, |\mathcal{N}_s|, i, d).$$

It follows that the probability that all nodes in $\mathcal{N}_r \setminus \mathcal{N}_s$ are uncontaminated at time t is

$$\left((1-p_b) h(0, |\mathcal{N}_s|, i, d)\right)^t, \text{ for } 0 \leq t \leq |\mathcal{N}_r| - y.$$

Note that since $|\mathcal{N}_r \setminus \mathcal{N}_s|$ nodes are added, the information dissemination process ends at $t = |\mathcal{N}_r| - y$. Now, the

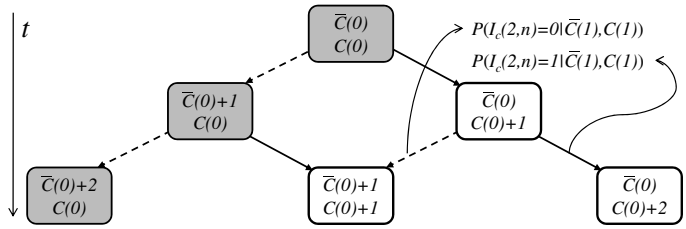

Fig. 2. Markov diagram for the dissemination process, $|\mathcal{N}_r| - Y = 2$. The transitions to the left (dotted arrows) represent the addition of an uncontaminated node, and the transitions to the right (filled arrows) represent the addition of a contaminated node. The grey states are considered in computing Ψ, that is, the states in which no contaminated nodes are added.

probability that only uncontaminated nodes exist in $\mathcal{N}_r$ at time $t = |\mathcal{N}_r| - y$, conditioned on $Y = y$ and $N_b = i$, is

$$f(i,y) = \left(1 - \frac{i}{|\mathcal{N}_s|}\right)^y \left[(1-p_b)h(0,|\mathcal{N}_s|,i,d)\right]^{|\mathcal{N}_r|-y}.$$

N_b has a binomial distribution, Y has a hypergeometric distribution and they are independent of each other. Taking out these two conditions, the probability that all nodes in $\mathcal{N}_r$ are uncontaminated is given by

$$\gamma = \sum_{y=0}^{|\mathcal{N}_s|} h(y;|\mathcal{N}|,|\mathcal{N}_s|,|\mathcal{N}_r|) \left(\sum_{i=0}^{|\mathcal{N}_s|} B(i,|\mathcal{N}_s|,p_b) f(i,y)\right).$$

It follows that the blocking probability is $\Psi = 1 - \gamma$. ■

We now consider that the list of informed nodes $N(t)$ at the tracker is updated with each new informed level-r node.

Theorem 2 (Evolving informed nodes list): Let $G(t)$ be an information contact graph in which nodes in $|\mathcal{N}_r \backslash \mathcal{N}_s|$ connect to nodes in $N(t)$. Then its blocking probability Ψ is:

$$\Psi = 1 - \sum_{y=0}^{|\mathcal{N}_s|} h(y;|\mathcal{N}|,|\mathcal{N}_s|,|\mathcal{N}_r|) \left(\sum_{i=0}^{|\mathcal{N}_s|} B(i,|\mathcal{N}_s|,p_b) f(i,y)\right),$$

$$f(i,y) = \left(1 - \frac{i}{|\mathcal{N}_s|}\right)^y \left[\prod_{t=1}^{|\mathcal{N}_r|-y} (1-p_b) h(0;|\mathcal{N}_s|+t-1,i,d)\right].$$

Proof: Recall from *Theorem 1* that we consider two disjoint subsets of $\mathcal{N}_r$, that is, $\mathcal{N}_r \cap \mathcal{N}_s$ and $\mathcal{N}_r \backslash \mathcal{N}_s$. As before, Y is the number of nodes in $\mathcal{N}_r \cap \mathcal{N}_s$. Again, at time $t = 0$, the probability that all nodes in $\mathcal{N}_r \cap \mathcal{N}_s$ are uncontaminated given $N_b = i$ and $Y = y$ is $(1 - i/|\mathcal{N}_s|)^y$.

We now consider the nodes in $\mathcal{N}_r \backslash \mathcal{N}_s$ and assume $N_b = i, Y = y$. At each time step, there are $C(t)$ contaminated nodes and $\overline{C}(t) = |\mathcal{N}_s| + t - C(t)$ uncontaminated nodes in $N(t)$. The probability of obtaining a contaminated node at time $t+1$ is only dependent on $C(t)$ and $\overline{C}(t)$, and thus, we can model these probabilities by Markov chains $\Xi|N_b, Y = \{S, \mathbf{P}\}$, in which S represents the set of states and $\mathbf{P}$ represents the matrix of transition probabilities. A state in S is represented by $s = (C(t), \overline{C}(t))$. Transitions from s are only possible to $s' = (C(t)+1, \overline{C}(t))$ and to $s'' = (C(t), \overline{C}(t)+1)$. It is also important to note that the depth of the Markov chain is equal to $|\mathcal{N}_r \backslash \mathcal{N}_s| = |\mathcal{N}_r| - y$. The transition probabilities from s when adding a node n are $P(s \to s') = P(I_c(t+1,n) = 1 | C(t), \overline{C}(t), N_b, Y)$ and $P(s \to s'') = P(I_c(t+1,n) = 0 | C(t), \overline{C}(t), N_b, Y)$. $\Xi|N_b, Y$ is illustrated in *Figure 2* for $|\mathcal{N}_r \backslash \mathcal{N}_s| = 2$.

Let us denote $C(t)$ as x and $t' = |\mathcal{N}_s| + t$, it follows that $\overline{C}(t) = t' - x$. Let $p^t_{\{s\}} = p^t_{\{x,t-x\}}$ denote the probability of being in state s at time t, and

$$p^t_{\{x,t'-x\}} = p^{t-1}_{\{x-1,t'-x\}} p_{(\{x-1,t'-x\} \to \{x,t'-x\})} + p^{t-1}_{\{x,t'-x-1\}} p_{(\{x,t'-x-1\} \to \{x,t'-x\})},$$

$$p_{(\{x,t'-x\} \to \{x+1,t'-x\})} = 1 - P(I_c(t,n) = 0 | x, t'-x, N_b = i, Y = y),$$

$$p_{(\{x,t'-x\} \to \{x,t'-x+1\})} = P(I_c(t,n) = 0 | x, t'-x, N_b = i, Y = y),$$

$$p^0_{\{i,|\mathcal{N}_s|-i\}} = 1.$$

Now, consider that node n is active at time t. The probability of n being uncontaminated is the probability that it is not Byzantine and does not connect to contaminated nodes. Thus,

$$P(I_c(t,n) = 0 | \overline{C}(t-1), C(t-1), N_b = i, Y = y) = (1-p_b) h(0;|\mathcal{N}_s|+t-1, C(t-1), d).$$

Now, notice that the probability of only having uncontaminated nodes at time $t = |\mathcal{N}_r| - y$ is the probability of, starting in state $(C(0), \overline{C}(0)) = (i, |\mathcal{N}_s| - i)$, ending in state $(i, |\mathcal{N}_s| - i + |\mathcal{N}_r| - y)$ after $|\mathcal{N}_r| - y$ steps: in that case, no contaminated node is added to the network. The probability of this event, conditioned on $N_b = i$ and $Y = y$, is

$$\prod_{t=1}^{|\mathcal{N}_r|-y} P(I_c(t,n) = 0 | \overline{C}(t-1), C(t-1), N_b = i, Y = y) = \prod_{t=1}^{|\mathcal{N}_r|-y} (1-p_b) h(0;|\mathcal{N}_s|+t-1,i,d).$$

Combining the results for sets $\mathcal{N}_r \cap \mathcal{N}_s$ and $\mathcal{N}_r \backslash \mathcal{N}_s$, we have that the probability that no contaminated nodes exist in $\mathcal{N}_r$ given that $N_b = i$ and $Y = y$ is given by

$$f(i,y) = \left(1 - \frac{i}{|\mathcal{N}_s|}\right)^y \left[\prod_{t=1}^{|\mathcal{N}_r|-y} (1-p_b) h(0;|\mathcal{N}_s|+t-1,i,d)\right].$$

Finally, it follows that Ψ at time $|\mathcal{N}_r \backslash \mathcal{N}_s|$ is

$$\Psi = 1 - \sum_{y=0}^{|\mathcal{N}_s|} h(y;|\mathcal{N}|,|\mathcal{N}_s|,|\mathcal{N}_r|) \left(\sum_{i=0}^{|\mathcal{N}_s|} B(i,|\mathcal{N}_s|,p_b) f(i,y)\right).$$

■

C. Interpretation of the Impact of Byzantine Attacks

The results from Theorems 1 and 2 are illustrated in Figure 3. Not surprisingly, the blocking probability Ψ grows exponentially with p_b for both Theorems. This is because it is sufficient for a single level-r node to connect to a Byzantine node in level-s to contaminate the receiver. Figure 3 indicates that Ψ grows faster for the evolving informed node list than for the static informed node list. This is due to the fact that as more nodes are added to the network, the presence of contaminated nodes becomes more likely, and thus, the probability that a level-r node connects to at least one contaminated node

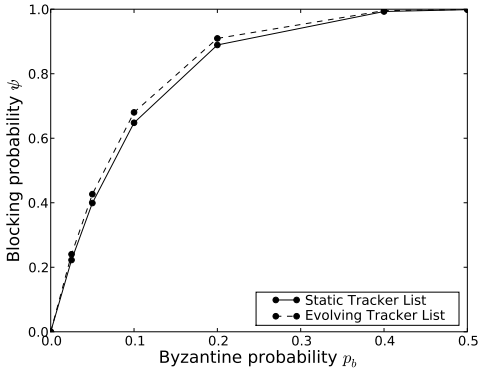

Fig. 3. Blocking probability in function of p_b for $|\mathcal{N}| = 30$, $|\mathcal{N}_s| = 5$, $|\mathcal{N}_r| = 6$ and $d = 3$. The results for the static and evolving informed nodes list are shown in full and dashed, respectively.

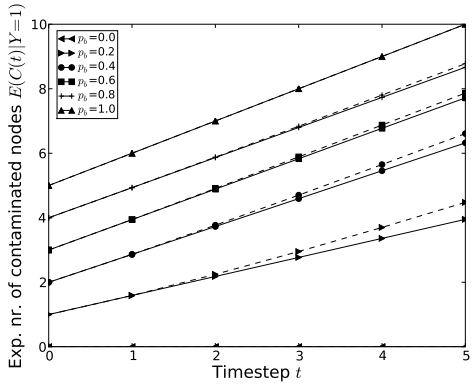

Fig. 4. Expected number of contaminated nodes in function of time, for $|\mathcal{N}| = 30$, $|\mathcal{N}_s| = 5$, $|\mathcal{N}_r| = 6$, $d = 3$ and $Y = 1$. The results for the static and evolving informed nodes list are shown in full and dashed, respectively.

increases. These results show that even the naïve attacker (randomly contaminating nodes) can carry out a very effective DDoS attack on a P2P network.

The probability Ψ also increases with other parameters such as d, $|\mathcal{N}_s|$, and $|\mathcal{N}_r|$. These parameters increase the number of connections as well as the number of nodes within the information contact graph $G(t)$, thus increasing the probability of level-r nodes connecting to contaminated nodes.

It is important to note that Ψ does not capture the impact of the Byzantine attack completely. The probability Ψ represents the likelihood of a successful DDoS attack to a receiver; however, it does not capture how much of the network has been contaminated. The number of contaminated nodes, $C(t)$, is closely tied with the topology of the information contact graph $G(t)$. From Section III-B, we observe that $C(t)$ is dependent on the random variable $Y = |\mathcal{N}_r \cap \mathcal{N}_s|$. Nodes $n \in Y$ are directly connected to the source and the receiver (since they are both level-s and level-r nodes). Therefore, the probability that n is contaminated is only dependent on p_b, while the probability that $n' \in \mathcal{N}_r \setminus \mathcal{N}_s$ is contaminated is dependent on p_b as well as connecting to a contaminated node in $\mathcal{N}_s$. Thus, $C(t)$ decreases as $|Y|$ increases for any given p_b.

For a given topology (conditioned on $Y = y$), we can perform an analysis of $E[C(t)]$. (We do not provide the details of this for want of space.) For the case of static informed node list, $E[C(t)|Y = y]$ is equal to

$$\sum_{i=0}^{|\mathcal{N}_s|} B(i, |\mathcal{N}_s|, p_b)\Big(i + t(1-(1-p_b)h(0; |\mathcal{N}_s|, i, d))\Big).$$

In the case of the evolving informed nodes list,

$$E[C(t)|Y = y] = \sum_{i=0}^{|\mathcal{N}_s|} B(i, |\mathcal{N}_s|, p_b)\Bigg(\sum_{x=i}^{i+t} x p^t_{\{x,t'-x\}}\Bigg).$$

To visualize the above results, we plot $E[C(t)|Y = 1]$ with the same set of parameters as in Figure 3. Figure 4 shows that the expected number of contaminated nodes in the static case is linear with time. For small probabilities p_b, the $E[C(t)|Y = 1]$ is higher for the evolving case; as p_b increases, the expected number of $C(t)$ for both cases behave more similarly.

IV. Signature scheme for Byzantine detection

From the previous Section, we can see that coded P2P networks are highly vulnerable to Byzantine attacks, and the contamination can quickly spread throughout the network. Although we only consider a particular network model in Section III for the purpose of analysis, such problems exist in all network coded systems. Therefore, it is desirable to have a signature scheme that validates each received packet without decoding the whole file. Then the contamination can be contained in one-hop, and we can avoid the decoding delay. In uncoded systems, the source knows all the packets that are transmitted in the network, and therefore, can sign each one of them. However, in a coded system, each node produces "new" packets, and standard digital signature schemes do not apply. Previous work that attempts to solve this problem is based on homomorphic hash functions [8], [32], [31], Secure Random Checkup [16], or Weil pairing on elliptic curves [29]. In this section, we introduce a novel signature scheme for the coded system based on the Discrete Logarithm problem.

We consider a directed graph with a set of nodes $\mathcal{N}$. A *source* node has a large file to be sent to *receiver* nodes. The file is divided into m packets. A node in the network receives linear combinations of the packets from the source or from other nodes. In this framework, a node is also a server to packets it has downloaded, and always sends out random linear combinations of all the packets it has obtained so far to other nodes. When a receiver has received m linearly independent packets, it can re-construct the whole file. We denote the m original packets as $\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_m$, and view them as elements in l-dimensional vector space $\mathbb{F}_p^l$, where p is a prime. The source node adds coding vectors to create $\mathbf{v}_1, ..., \mathbf{v}_m$, $\mathbf{v}_i = (0, ..., 1, ..., 0, \bar{v}_{i1}, ..., \bar{v}_{il})$, where the first m elements are zero except the ith element which is 1, and $\bar{v}_{ij} \in \mathbb{F}_p$ is the jth element in $\bar{\mathbf{v}}_i$. A packet $\mathbf{w}$ received by a node is a linear combination of these vectors,

$$\mathbf{w} = \sum_{i=1}^{m} \beta_i \mathbf{v}_i,$$

where $(\beta_1, ..., \beta_m)$ is the global coding vector.

The key observation for our signature scheme is that the vectors $\mathbf{v}_1, ..., \mathbf{v}_m$ span a subspace V of $\mathbb{F}_p^{m+l}$, and a received

vector $\mathbf{w}$ is a valid linear combination of vectors $\mathbf{v}_1, ..., \mathbf{v}_m$ if and only if it belongs to V. Our scheme is based on standard modulo arithmetic (in particular the hardness of the Discrete Logarithm problem) and on an invariant signature for the linear span V. Each node verifies the integrity of a received vector $\mathbf{w}$ by checking the membership of $\mathbf{w}$ in V based on the signature.

Our scheme is defined by the following ingredients:

- q: a large prime number such that p is a divisor of $q-1$. Note that standard techniques, such as that used in Digital Signature Algorithm (DSA) [33], apply to finding such q.
- g: a generator of the group G of order p in $\mathbb{F}_q$. Since the order of the multiplicative group $\mathbb{F}_q^*$ is $q-1$ (a multiple of p), there exists a subgroup, G, with order p in $\mathbb{F}_q^*$.
- Private key: $\mathbf{K}_s = \{\alpha_i\}_{i=1,...,m+l}$, a random set of elements in $\mathbb{F}_p^*$, only known to the source.
- Public key: $\mathbf{K}_p = \{h_i = g^{\alpha_i}\}_{i=1,...,m+l}$, signed by some standard signature scheme, e.g., DSA, and published by the source.

To distribute a file in a secure manner, the signature scheme works as follows.

1) Using the vectors $\mathbf{v}_1, ..., \mathbf{v}_m$ from the file, the source finds a vector $\mathbf{u} = (u_1, ..., u_{m+l}) \in \mathbb{F}_p^{m+l}$ orthogonal to all vectors in V. Specifically, $\mathbf{u}$ is a non-zero solution, $\mathbf{u} \neq 0$, such that $\mathbf{v}_i \cdot \mathbf{u} = 0$ for $i = 1, ..., m$.
2) The source computes the vector $\mathbf{x} = (u_1/\alpha_1, u_2/\alpha_2, ..., u_{m+l}/\alpha_{m+l})$.
3) The source signs $\mathbf{x}$ with some standard signature scheme and publishes $\mathbf{x}$. We refer to the vector $\mathbf{x}$ as the signature of the file being distributed.
4) The client node verifies that $\mathbf{x}$ is signed by the source.
5) When a node receives a vector $\mathbf{w}$ and wants to verify that $\mathbf{w}$ is in V, it computes

$$d = \prod_{i=1}^{m+l} h_i^{x_i w_i},$$

and verifies that $d = 1$.

To see that d is equal to 1 for any valid $\mathbf{w}$, we have

$$d = \prod_{i=1}^{m+l} h_i^{x_i w_i} = \prod_{i=1}^{m+l} (g^{\alpha_i})^{u_i w_i/\alpha_i} = \prod_{i=1}^{m+l} g^{u_i w_i} = g^{\sum_{i=1}^{m+l}(u_i w_i)} = 1,$$

where the last equality comes from the fact that $\mathbf{u}$ is orthogonal to all vectors in V.

Next, we show that the system described above is secure. In essence, the theorem below shows that given a set of vectors that satisfy the signature verification criterion, it is provably at least as hard as the Discrete Logarithm problem to find new vectors that also satisfy the verification criterion other than those that are in the linear span of the vectors already known.

Definition 1: Let p be a prime number and G be a multiplicative cyclic group of order p. Let k and n be two integers such that $k < n$, and $\mathbf{\Gamma} = \{h_1, ..., h_n\}$ be a set of generators of G. Given a linear subspace, V, of rank k in $\mathbb{F}_p^n$ such that for every $\mathbf{v} \in V$, the equality $\mathbf{\Gamma}^{\mathbf{v}} \triangleq \prod_{i=1}^n h_i^{v_i} = 1$ holds, we define the (p, k, n)-Diffie-Hellman problem as the problem of finding a vector $\mathbf{w} \in \mathbb{F}_p^n$ with $\mathbf{\Gamma}^{\mathbf{w}} = 1$ but $\mathbf{w} \notin V$.

By this definition, the problem of finding an invalid vector that satisfies our signature verification criterion is a $(p, m, m+l)$-Diffie-Hellman problem. Note that in general, the $(p, n-1, n)$-Diffie-Hellman problem has no solution. This is because if V has rank $n-1$ and a $\mathbf{w}'$ exists such that $\mathbf{\Gamma}^{\mathbf{w}'} = 1$ and $\mathbf{w}' \notin V$, then $w' + V$ spans the whole space, and any vector $\mathbf{w} \in \mathbb{F}_p^n$ would satisfy $\mathbf{\Gamma}^{\mathbf{w}} = 1$. This is clearly not true, therefore, no such $\mathbf{w}'$ exists.

Theorem 3: For any $k < n-1$, the (p, k, n)-Diffie-Hellman problem is at least as hard as the Discrete Logarithm problem.

Proof: Assume there exists an efficient algorithm to solve the (p, k, n)-Diffie-Hellman problem, and we wish to compute the discrete logarithm $\log_g(z)$ for some $z = g^x$, where g is a generator of a cyclic group G with order p. We can choose two random vectors $\mathbf{r} = (r_1, ..., r_n)$ and $\mathbf{s} = (s_1, ..., s_n)$ in $\mathbb{F}_p^n$, and construct $\mathbf{\Gamma} = \{h_1, ..., h_n\}$, where $h_i = z^{r_i} g^{s_i}$ for $i = 1, ..., n$. We then find k linearly independent (and otherwise random) solutions $\mathbf{v}_1, ..., \mathbf{v}_k$ to the equations

$$\mathbf{v} \cdot \mathbf{r} = 0 \text{ and } \mathbf{v} \cdot \mathbf{s} = 0.$$

Note that there exist $n-2$ linearly independent vector solutions to the above equations. Let V be the linear span of $\{\mathbf{v}_1, ..., \mathbf{v}_k\}$, then any vector $\mathbf{v} \in V$ satisfies $\mathbf{\Gamma}^{\mathbf{v}} = 1$. Now, if we have an algorithm for the (p, k, n)-Diffie-Hellman problem, we can find a vector $\mathbf{w} \notin V$ such that $\mathbf{\Gamma}^{\mathbf{w}} = 1$. This vector would satisfy $\mathbf{w} \cdot (x\mathbf{r} + \mathbf{s}) = 0$. Since $\mathbf{r}$ is statistically independent from $(x\mathbf{r}+\mathbf{s})$, with probability greater than $1 - 1/p$, we have $\mathbf{w} \cdot \mathbf{r} \neq 0$. In this case, we can compute

$$\log_g(z) = x = \frac{\mathbf{w} \cdot \mathbf{s}}{\mathbf{w} \cdot \mathbf{r}}.$$

Thus, the ability to solve the (p, k, n)-Diffie-Hellman problem implies the ability to solve the Discrete Logarithm problem. ■

This proof is an adaptation of a proof in an earlier publication by Boneh *et. al* [34].

Our signature scheme makes use of the linearity property of RLNC, and enables the nodes to check the integrity of packets without a secure channel, unlike the homomorphic hash function or SRC schemes [16], [32]. In addition, our scheme does not require the nodes to decode coded packets to check their validity – thus, is efficient in terms of delay. The computation involved in the signature generation and verification processes is very simple. Furthermore, our scheme uses the Discrete Logarithm problem, which is more standardized and widely used, compared to the recently developed Weil pairing problem used in [29]. Lastly, we note that our signature scheme is rateless [21], which is not the case in end-to-end or generation based detection schemes.

V. Overhead analysis

In the previous Sections, we showed that our signature scheme is beneficial, as even a small amount of attack can have a devastating effect in coded networks. However, we have not shown that this scheme is efficient in terms of bandwidth (*i.e.* overhead of augmenting the signature scheme),

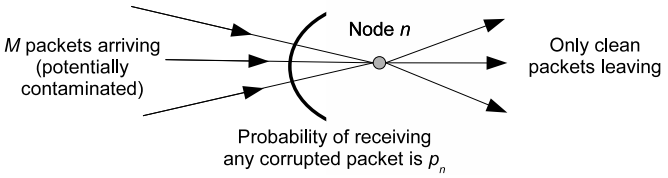

Fig. 5. Diagram of the network and node n

and indeed, it is not always the case that our signature scheme is desirable. We now study the cost and benefit of the following three Byzantine schemes: 1) our signature scheme proposed in Section IV, 2) end-to-end error correction scheme [15], and 3) generation-based Byzantine detection scheme [6]. If we implement Byzantine detection schemes, we can detect contaminated data, drop them, and therefore, only transmit valid data; however, this benefit comes with the overhead of the schemes in the forms of hashes and signatures. It is important to note that, for the dropped data, the receivers perform erasure correction, which is computationally lighter than error correction; thus, there is no need of retransmissions.

We consider a node $n \in \mathcal{N}$ in the network as in Section IV. Node n wishes to check the validity of the data it forwards. Assume that node n receives M packets per time slot. Recall from Section IV that m is the number of original packets of l-dimensional vector space $\mathbb{F}_p^l$. The source adds the coding vector to the original packets, where the first m symbols are the coding coefficients. Therefore, the packet transmitted consists of $(m + l)$ symbols. If n detects an error, then it discards that data; otherwise, it forwards the data. The probability that n receives a contaminated packet is p_n as shown in Figure 5. Note that the probability p_n of an attack is topology dependent. However, in order to compare the performance of various schemes, we use a generic per node model to examine the overhead incurred at a node. We assume that there is an external model of vulnerability which gives an estimate of p_n. Note that the blocking probability Ψ analyzed in Section III provides such an estimate.

A. Overhead analysis of our packet-based signature scheme

We examine the overhead incurred by our signature scheme. Recall from Section IV, the file size is $ml \log p$ bits. The file is divided into m packets, each of which is a vector in $\mathbb{F}_p^l$. Thus, the overhead of the RLNC scheme is m/l times the file size, and in practical networks $m \ll l$. There are two components to the overhead of our scheme. The first is the cost associated with the initial setup – *i.e.* publishing the public key $\mathbf{K}_p$; second is the signature vector $\mathbf{x}$ of the file.

We first consider the cost of initial setup. Note that the public key $\mathbf{K}_p$ is $(m+l)\log(q)$ bits. In typical cryptographic applications, the sizes of p and q are 20 bytes (160 bits) and 128 bytes (1024 bits), respectively; thus, the size of $\mathbf{K}_p$ is approximately $6(m+l)/ml$ times the file size. This overhead is negligible as long as $6 \ll m \ll l$. For example, if we have a 10MB file divided into $m = 100$ packets, then the overhead is approximately 6%. The second part of the overhead is publishing the signature $\mathbf{x}$, which is $(m+l)\log(p)$ bits. Thus, for a 10MB file, the overhead is approximately 1%.

Note that the public key $\mathbf{K}_p$ cannot be fully reused for multiple files, as it is possible for an attacker to generate a vector which is not a valid linear combination of the original vectors yet satisfies the check $d = 1$ using information obtained from previously downloaded files. (We do not provide the details of this for want of space.) To prevent this from happening, we can redistribute keys for each additional file in one of the two methods below. The first method consists of publishing a new public key $\mathbf{K}_p$ for each file, incurring an overhead of $6(m + l)/ml$ times the file size. Note that if we republish $\mathbf{K}_p$ for every file, we can reuse the signature $\mathbf{x}$. The second method is to update $\mathbf{K}_p$ partially and generate a new $\mathbf{x}$ for each file. This incurs less overhead than the previous method, however, requires a high variability in $\mathbf{w}$ for it to be secure. This update incurs negligible amount of overhead. For example, for a 10MB file, the overhead is less than 0.1%.

Thus, the initial setup costs approximately 6% of our first file size. For subsequent files, the incremental update of $\mathbf{K}_p$ and $\mathbf{x}$ is less than 0.1% if we use the second method. Note that if amortized over η files, our signature scheme would have an overhead of $\frac{6+0.1(\eta-1)}{\eta}\%$. For example, if $\eta = 10$, overhead of our signature scheme is less than 1%.

For simplicity, in the remaining of the paper, we assume that we are distributing only one file. Therefore, we shall denote the overhead associated with our signature by $o_p \triangleq \frac{6}{100}(m + l)$ symbols per packet, *i.e.* 6% overhead – which is a gross overestimate if we are distributing more than one file.

If n detects an error in a packet, then it discards it – by doing so, n can filter out all the contaminated packets and use its bandwidth to transmit only valid packets. Therefore, n only forwards on average $1 - \frac{o_p}{m+l}$ fraction of the data received.

Our signature scheme costs $o_p M$ symbols per time slot. However, by discarding the contaminated packets, node n can on average save its bandwidth by $M(m + l)p_n$ symbols per time slot. Therefore, the net cost of the signature scheme as a fraction of the total data received is:

$$\frac{\max\{0, Mo_p - M(m+l)p_n)\}}{M(m+l)} = \frac{\max\{0, o_p - (m+l)p_n\}}{m+l}. \quad (1)$$

When p_n is high, then checking each packet for error saves on bandwidth – *i.e.* $(o_p - (m + l)p_n) < 0$, which shows that the cost of the signature scheme is canceled by the bandwidth gained from dropping the corrupted packets. Therefore, this approach is the most sensible when the network is unreliable or under heavy attack.

B. Overhead analysis of end-to-end error correction

In this subsection, we shall use the rate-optimal error correction codes from Jaggi *et al.* [15]. As long as the attack is within the network capacity, this scheme allows the intermediate nodes to transmit at the remaining network capacity, *i.e.* the end-to-end network capacity minus the capacity the adversary can contaminate. In this scenario, node n just naively performs RLNC and forwards the data it has received. Therefore, node n transmits on average $M(m + l)p_n$ contaminated symbols. Thus, the net cost as a fraction of the total data received is:

$$\frac{M(m + l)p_n}{M(m + l)} = p_n. \quad (2)$$

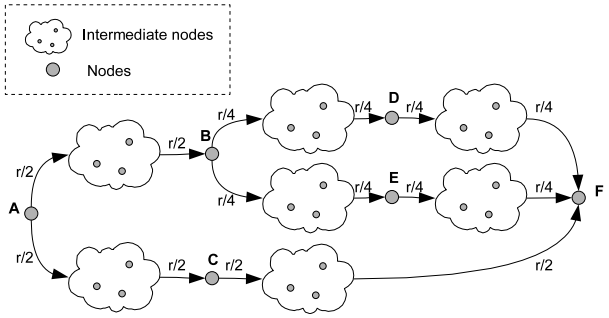

Fig. 6. Network with non-malicious nodes $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{F}$ where node $\mathbf{A}$ is transmitting at a total rate of $\mathbf{r}$ to node $\mathbf{F}$; however, $\mathbf{A}$ sends half of its data through $\mathbf{B}$ and the other half through $\mathbf{C}$. Therefore, $\mathbf{B}$ and $\mathbf{C}$ can check the validity of the *sub-generation* they receive, where by sub-generation, we mean a collection of $G/2$ encoded packets from $\mathbf{A}$. By a similar argument, $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{F}$ can check the validity of a sub-generation of $G/4$, $G/4$, and G packets from $\mathbf{A}$, respectively.

C. Overhead analysis of generation-based detection scheme

We now analyze the performance of the algorithm proposed by Ho *et al.* [28], which uses random block linear network coding with generation size G (although we have focused on RLNC so far, it is possible to extend these results by considering m as the generation size G). This scheme is very cheap – with 2% overhead, the detection probability is at least 98.9%. We denote the overhead associated with this scheme by $o_g \triangleq \frac{2}{100}(m+l)G$ symbols per generation.

After collecting enough packets from the generation, node n checks for possible error in the generation, which can incur large delay. If n detects an error, it discards the entire generation of G packets; otherwise, it forwards the data. This scheme requires only one hash for the entire generation – saving bits on the hashes compared to our signature scheme. However, it can be inefficient, as one contaminated packet can cause n to discard an entire generation. The probability p_g of dropping a generation of G packets is given by:

$$p_g = 1 - \Pr(\text{All } G \text{ packets are valid}) = 1 - (1-p_n)^G.$$

The cost and benefit of this scheme includes three components: (i) the hash of o_g symbols per generation, (ii) valid packets which are discarded if the generation is deemed contaminated, and (iii) bandwidth saved by dropping contaminated packets. The expected number of valid symbols dropped per generation is $p_g(1-p_n)(m+l)G$. The expected number of contaminated symbols per generation is $p_n(m+l)G$. Thus, the net cost as a fraction of the total data received is:

$$\frac{\max\{0, o_g + p_g(1-p_n)(m+l)G - p_n(m+l)G\}}{(m+l)G}. \quad (3)$$

For this scheme to work, n needs to receive at least G packets from each generation to decode and detect errors. This may seem to indicate that this scheme is only applicable as an end-to-end scheme, but it can be extended to a *local* Byzantine detection scheme as shown in Figure 6.

The cost of this scheme increases dramatically with G. If G is large, the probability of at least one corrupted packet in a generation is high even for a small p_n. Thus, a large G is undesirable, as almost every generation is found faulty and dropped, making the throughput approach zero. This can be verified with an asymptotic analysis of Equation 3:

$$\lim_{G\to\infty} \frac{\max\{0, o_g + p_g(1-p_n)(m+l)G - p_n(m+l)G\}}{(m+l)G} \to \max\{0, 1-2p_n\}.$$

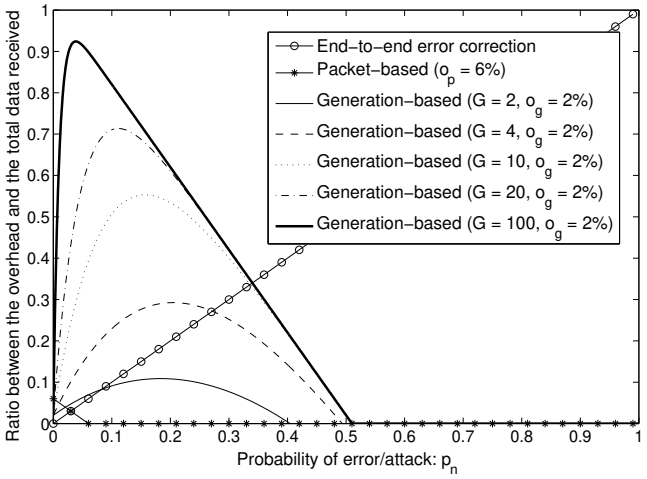

Fig. 7. Ratio between the expected overhead and the total data received by a node with 1000 bit packets. For generation-based detection, G is the generation size and $o_g = \frac{2}{100}(m+l)G$. For packet-based detection, $o_p = \frac{6}{100}(m+l)$.

Note in Figure 7 that the cost peaks at $p_n \approx 0.2$. At $p_n \approx 0.2$, the scheme drops many generations for a few corrupted packets. Thus, at a moderate rate of attack, the generation-based scheme suffers. When $p_n < 0.2$, the generation-based scheme does well, since p_n is low and the cost of hash is distributed across G packets. As p_n increases to 0.5 from 0.2, the throughput to the receiver decreases as more generations are dropped. When $p > 0.5$, this scheme discards almost all generations, thus, the expected throughput is near zero.

D. Trade-offs and comparisons

In Figures 7 and 8, we compare the three schemes. From Section V-B, the expected cost of error correction scheme is linearly proportional to p_n. For large p_n, this scheme performs badly. However, this simple scheme where a node naively forwards all data it receives outperforms the detection schemes when p_n is low ($p_n < 0.03$). When p_n is small, the overhead of detection exceeds the cost introduced by the attackers.

When p_n is low, the overhead of our signature is costly, since we are devoting o_p symbols per packet to detect an unlikely attack. In such a setting, the generation-based scheme performs well, as it distributes the cost of the hash (o_g symbols) over G packets. However, as p_n increases, the cost of our signature becomes negligible since the bandwidth wasted by contaminated packets increases; thus, our signature scheme outperforms the generation-based scheme.

Note that we may be underestimating the overhead associated with our signature scheme as we only take into account the cost of publishing the public key $\mathbf{K}_p$ and the signature $\mathbf{x}$. We do not consider the cost of maintaining a public key distribution infrastructure, which the generation-based scheme does not require. Thus, depending on the public key distribution infrastructure used, our scheme may incur a higher overhead – resulting in an outward shift in the overhead

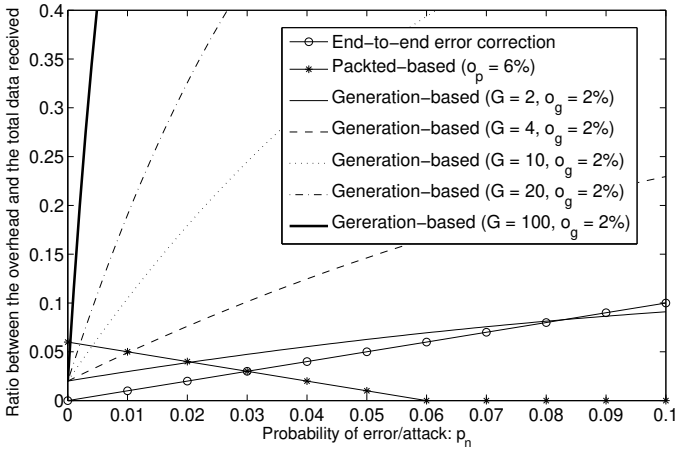

Fig. 8. Figure 7 for $p_n \in [0, 0.1]$

in Figure 7. However, it is important to note that in terms of transmission cost, we overestimate the overhead o_p our signature scheme. Recall from Section V-A, we set $o_p = 6\%$ assuming that we are distributing only one file; however, if we are distributing more files, $o_p \ll 1\%$.

We briefly note the computational cost of these schemes. When using a detection scheme, node n does not waste its bandwidth in transmitting contaminated data by dropping them. Furthermore, there is no need of retransmission of the dropped data as the receivers can perform erasure correction on the packets/generations that have been dropped. It is important to note that for the end-to-end error correction scheme, the receivers need to perform error correction, which is computationally more expensive than erasure correction.

VI. Conclusions

In this paper, we studied the problem of Byzantine attacks in network coded P2P networks. We used randomly evolving graphs to characterize the impact of Byzantine attackers on the receiver's ability to recover a file. As shown by our analysis, even a small number of attackers can contaminate most of the flow to the receivers. Motivated by this result, we proposed a novel signature scheme for any network using RLNC. The scheme makes use of the linearity of the code, and it can be used to easily check the validity of all received packets. Using this scheme, we can prevent the intermediate nodes from spreading the contamination by allowing nodes to detect contaminated data, drop them, and therefore, only transmit valid data. We emphasize that there is no need of retransmission for the dropped data since the receivers can perform erasure correction, which is computationally cheaper than error correction.

We analyzed the cost and benefit of the signature scheme, and compared it with various detection schemes. We showed that the overhead of our scheme is low. Furthermore, when the probability of attack is high, it is the most bandwidth efficient. However, if the probability of attack is low, generation-based detection schemes are more appropriate.

References

[1] 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.

[2] T. Ho, M. Médard, M. Effros, and D. Karger, "The benefits of coding over routing in a randomized setting," in *Proceedings of IEEE ISIT*, Kanagawa, Japan, July 2003.

[3] Z. Li and B. Li, "Network coding: the case of multiple unicast sessions," in *Proceedings of 42nd Annual Allerton Conference on Communication Control and Computing*, September 2004.

[4] D. Lun, M. Médard, and R. Koetter, "Network coding for efficient wireless unicast," in *Proceedings of International Zurich Seminar on Communications*, Zurich, Switzerland, February 2006.

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

[6] D. Lun, M. Medard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, 2008.

[7] T. Ho, M. Médard, R. Koetter, M. Effros, J. Shi, and D. R. Karger, "A random linear coding approach to mutlicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, 2006.

[8] S. Acedański, S. Deb, M. Médard, and R. Koetter, "How good is random linear coding based distributed network storage?" in *Proceedings of 1st Netcod*, Riva del Garda, Italy, April 2005.

[9] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[10] "Bittorrent file sharing protocol," http://www.BitTorrent.com.

[11] C. Gkantsidis, J. Miller, and P. Rodriguez, "Comprehensive view of a live network coding p2p system," in *Proceedings of ACM SIGCOMM/USENIX Internet Measurement Conference*, Rio de Janeiro, Brazil, October 2006.

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

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

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

[15] 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.

[16] C. Gkantsidis and P. Rodriguez, "Cooperative security for network coding file distribution," in *Proceedings of IEEE INFOCOM*, April 2006.

[17] A. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, p. 509, 1999.

[18] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," *Phys. Rev. E*, vol. 64, no. 4, p. 046135, September 2001.

[19] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Phys. Rev. Lett.*, vol. 86, no. 14, pp. 3200–3203, Apr 2001.

[20] R. M. May and A. L. Lloyd, "Infection dynamics on scale-free networks," *Phys. Rev. E*, vol. 64, no. 6, p. 066112, Nov 2001.

[21] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," in *Proceedings of IEEE ISIT*, June 2007.

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

[23] L. Lima, J. Barros, and R. Koetter, "Byzantine attacks against network coding in peer to peer distributed storage," in *Proceedings of IEEE ISIT*, June 2009.

[24] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

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

[26] R. Koetter and F. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, pp. 3579–3591, 2008.

[27] D. Silva and F. Kschischang, "Adversarial error correction for network coding: Models and metrics," in *Proceedings of Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, September 2008.

[28] T. Ho, B. Leong, R. Koetter, M. Médard, and M. Effros, "Byzantine modification detection in multicast networks using randomized network coding," in *Proceedings of IEEE ISIT*, June 2004.

[29] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proceedings of Conference on Information Sciences and Systems*, March 2006.

[30] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, no. 2-3, pp. 173–193, March 2000.
[31] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proceedings of IEEE INFOCOM*, Pheonix, AZ, April 2008.
[32] M. Krohn, M. Freedman, and D. Mazières, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Proceedings of IEEE Symposium on Security and Privacy*, May 2004.
[33] National Institute of Standards and Technology, "Digital signature standard (DSS)," *FIBS PUB 186-2*, 2000.
[34] D. Boneh and M. Franklin, "An efficient public key traitor tracing scheme," in *Lecture Notes in Computer Science*, vol. 1666. Springer-Verlag, 1999, pp. 338–353.

MinJi Kim is currently pursuing a Ph.D. degree in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology (MIT), Cambridge, USA. She received B.S. degrees in Electrical Engineering and Computer Science, and in Mathematics in 2006; a Master of Engineering degree in Electrical Engineering and Computer Science in 2007, all from the Massachusetts Institute of Technology (MIT), Cambridge, USA. Her research interests include wireless communications and networks, security and reliability of networks, algorithms and optimization on networks, and network coding.

Luísa Lima is currently pursuing the Ph.D. degree at the Universidade do Porto (UP), Porto, Portugal. She is also a researcher in the Networking and Information Processing Group (NIP) of the Instituto de Telecomunicações (IT) and collaborates regularly with the Research Laboratory of Electronics at MIT. Her research interests include network coding, security, random graphs, video streaming and computer simulation.

Fang Zhao is currently pursuing a Ph.D. degree in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology (MIT), Cambridge, USA. She received a B.Eng. degree (First Class Honours) in 2000 and an M.Eng degree in 2001, both in Electrical Engineering from the National University of Singapore. Her research interests include network coding, optimization in networks, distributed algorithms, and network security.

João Barros is an Associate Professor at the Department of Electrical and Computer Engineering of the University of Porto and the coordinator of the Porto Laboratory of the Instituto de Telecomunicações. In February 2009, Dr. Barros was appointed National Director of the CMU-Portugal Program, a five-year international partnership between Carnegie Mellon University and 12 Portuguese Universities and Research Institutions, with a total budget of 56M Euros. He received his undergraduate education in Electrical and Computer Engineering from the Universidade do Porto (UP), Portugal and Universitaet Karlsruhe, Germany, until 1999, and the Ph.D. degree in Electrical Engineering and Information Technology from the Technische Universitaet Muenchen (TUM), Germany, in 2004. From 2005 to 2008, João Barros was an assistant professor at the Department of Computer Science of the University of Porto. The focus of his research lies in the general areas of information theory, communication networks and data security. Dr. Barros received a Best Teaching Award from the Bavarian State Ministry of Sciences, Research and the Arts, as well as scholarships from several institutions, including the Fulbright Commission and the Luso-American Foundation. He held visiting positions at Cornell University and the Massachusetts Institute of Technology, where he spent a sabbatical in 2008. Beyond his duties as Secretary of the Board of Governors of the IEEE Information Theory Society, his service included co-chairing the 2008 IEEE Information Theory Workshop in Porto, Portugal, and participating in several Technical Program Committees, including ITW 2009, WiOpt (2008 and 2009), ISIT 2007, IS 2007, and IEEE Globecom (2007 and 2008).

Muriel Médard is a Professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology. She was previously an Assistant Professor in the Electrical and Computer Engineering Department and a member of the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign. From 1995 to 1998, she was a Staff Member at MIT Lincoln Laboratory in the Optical Communications and the Advanced Networking Groups. Professor Médard received B.S. degrees in EECS and in Mathematics in 1989, a B.S. degree in Humanities in 1990, a M.S. degree in EE in 1991, and a Sc. D. degree in EE in 1995, all from the Massachusetts Institute of Technology (MIT), Cambridge. She serves as an Associate Editor for the Optical Communications and Networking Series of the IEEE Journal on Selected Areas in Communications, as an Associate Editor in Communications for the IEEE Transactions on Information Theory and as a Guest Editor for the Joint special issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transactions on Networking on Networking and Information Theory. She has served as a Guest Editor for the IEEE Journal of Lightwave Technology and as an Associate Editor for the OSA Journal of Optical Networking.

Professor Médard's research interests are in the areas of network coding and reliable communications, particularly for optical and wireless networks. She was awarded the IEEE Leon K. Kirchmayer Prize Paper Award 2002 for her paper, "The Effect Upon Channel Capacity in Wireless Communications of Perfect and Imperfect Knowledge of the Channel," IEEE Transactions on Information Theory, Volume 46 Issue 3, May 2000, Pages: 935–946. She was co-awarded the Best Paper Award for G. Weichenberg, V. Chan, M. Médard, "Reliable Architectures for Networks Under Stress", Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003), October 2003, Banff, Alberta, Canada. She received a NSF Career Award in 2001 and was a co-winner of the 2004 Harold E. Edgerton Faculty Achievement Award, established in 1982 to honor junior faculty members "for distinction in research, teaching and service to the MIT community." She was named a 2007 Gilbreth Lecturer by the National Academy of Engineering. Professor Médard is a Fellow of IEEE.

Ralf Koetter (S'91—M'96) received the Diploma in electrical engineering degree from the Technical University Darmstadt, Darmstadt, Germany in 1990 and the Ph.D. degree from the Department of Electrical Engineering at Linköping University, Linköping, Sweden. From 1996 to 1997, he was a visiting scientist at the IBM Almaden Research Laboratory, San Jose, CA. He was a Visiting Assistant Professor at the University of Illinois at Urbana-Champaign and a Visiting Scientist at CNRS in Sophia Antipolis, France. He joined the faculty of the University of Illinois at Urbana-Champaign in 1999 and is currently the head of the Institute for Communications Engineering at Technische Universität München.

His research interests included coding and information theory and their application to communication systems. Professor Koetter served as Associate Editor for Coding Theory and Techniques for the IEEE Transactions on Communications from 1999 to 2001. In 2003, he concluded a term as Associate Editor for Coding Theory of the IEEE Transactions on Information Theory. He received an IBM Invention Achievement Award in 1997, an NSF CAREER Award in 2000, and an IBM Partnership Award in 2001. He was a co-recipient of the 2004 Paper Award of the IEEE Information Theory Society. Since 2003, he was a Member of the Board of Governers of the IEEE Information Theory Society. Ralf Koetter passed away recently in February 2009.

Ton Kalker is a Distinguished Technologist at Hewlett-Packard Laboratories. He made significant contributions to the field of media security, in particular digital watermarking, robust media identification and interoperability of Digital Rights Managements systems. His history in this field of research started in 1996, submitting and participating in the standardization of video watermarking for DVD copy protection. His solution was accepted as the core technology for the proposed DVD copy protection standard and earned him the title of Fellow of the IEEE. His subsequent research focused on robust media identification, where he laid the foundation of the Content Identification business unit of Philips Electronics, successful in commercializing watermarking and other identification technologies. In his Philips period he has co-authored 30 patents and 39 patent applications. His interests are in the field of signal and audio-visual processing, media security, biometrics, information theory and cryptography.

Joining Hewlett-Packard in 2004, he focused his research on the problem of non-interoperability of DRM systems. He became one of the three lead architects of the Coral consortium, publishing a standard framework for DRM interoperability in the summer of 2007. He also participates actively in the academic community, through students, publications, keynotes, lectures, membership in program committees and serving as conference chair. Together with Pierre Moulin he is one of the two co-founders of the IEEE Transactions on Information Forensics. He is the former chair of the associated Technical Committee of Information Forensics and Security. He served for 6 years as visiting faculty at the University of Eindhoven. He is currently a visiting professor at the Harbin Institute of technology.

Keesook J. Han received the B.E.E., M.S., and Ph.D. degrees in electrical engineering from University of Minnesota, Minneapolis, MN. Following receipt of the Ph. D. degree, she held postdoctoral position as a National Research Council Research Associate at the Air Force Research Laboratory. Then, she joined the Information Directorate of the Air Force Research Laboratory. Her research interests in the areas of multimedia applications, video communication, visual quality analysis, network coding, compression, computer and network risk assessment/management, intrusions detection, network security, wireless information assurance, assessment of information damage, and cyber forensics. She received AFRL Invention Award. She is a researcher of AFRL Center for Integrated Transmission and Exploitation (CITE), Cornell Information Assurance Institute (AIA), and Net Centric Operations (NCO) Working Group. She is an advisor of National Research Council Research (NRC) and American Society for Engineering Education (ASEE). She is a reviewer of IEEE Transaction on Signal Processing, IEEE Transaction on Multimedia, IEEE Transaction on Image Processing, IEEE Transaction on Image Processing, IEEE Transaction on Information Forensics and Security, IEEE Transaction on Audio, Speech, and Language Processing, IEEE Signal Processing Letter. She is a member of The National Honor Society (Phi Kappa Phi), The National Engineering Honor Society (Tau Beta Pi) and The National Electrical Engineering Honor Society (Eta Kappa Nu).