Multi-receiver authentication code for network coding

Oggier, Frederique; Fathi, Hanane

2008

<https://hdl.handle.net/10356/91655>

© 2006 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Downloaded on 26 Apr 2023 22:05:00 SGT

Multi-receiver Authentication Code for Network Coding

Frédérique Oggier

School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
Email: frederique@ntu.edu.sg

Hanane Fathi

Center for TeleInfrastuktur
Aalborg University
Denmark
Email: hf@es.kom.aau.dk

Abstract—Systems exploiting network coding suffer greatly from pollution attacks which consist of injecting malicious packets in the network. The pollution attacks are amplified by the network coding process resulting in a greater damage than under traditional routing. In this paper, we address this issue with unconditionally secure multi-receiver authentication codes designed for network coding. Our scheme is robust against pollution attacks from outsiders and even from a coalition of $k-1$ malicious insiders. With our authentication code, intermediate nodes can verify the message integrity and origin of the packets received without having to decode and thus detect and discard the malicious packets in-transit that fail the verification. This way, the pollution is canceled out before reaching the destinations.

I. INTRODUCTION

Network coding was first introduced in [1] as an innovative approach in achieving the capacity of a network for multicast communications. Network coding allows intermediate nodes between the source and the destinations not only to store and forward but also to encode the received packets before forwarding them. In [2], Li et. al showed that it is sufficient to use linear coding that allows intermediate nodes to generate outgoing packets as linear combinations of their incoming packets. In line with [2], [3] gave an algebraic framework for network coding with further developments for arbitrary networks and robust networking. For practical issues, [4] proposed a network coding framework that allows to deal with random packet loss, change of topology and delays.

Network coding offers various advantages not only for maximizing the usage of network resources but also for robustness to network impairments and packet losses. Various applications of network coding have therefore appeared ranging from file download and content distribution in peer-to-peer networks to distributed file storage systems.

These systems exploiting network coding suffer greatly from pollution attacks which consist of injecting malicious packets in the network. The malicious packets may come from the modification of received packets by a malicious intermediate node, or from the creation of bogus packets by a malicious intermediate node or rogue source. With no integrity check performed for packets in transit in the network, an honest intermediate node receiving a malicious packet would perform the encoding of the malicious packet with other packets resulting in corrupted outgoing packets that are

then forwarded on to the next nodes. The corrupted packets propagates then all through the network which creates a more severe damage than under traditional routing. The pollution attack or injection of malicious packets in a network is thus amplified by the network coding process.

One way to address this issue is through authentication that consists of protecting the integrity of a message, validating the identity of the source, and guaranteeing the non-repudiation of the source. This involves the use of cryptographic primitives:

- (1) digital signatures, or message authentication codes (MAC) for computational security (i.e. vulnerable against an attacker that has unlimited computational resources);
- (2) authentication codes for unconditional security (i.e. robust against an attacker that has unlimited computational resources).

Several schemes have been proposed to detect polluted packets at intermediate nodes [5], [6], [7], [8]. However all of them are based on cryptographic functions with computational assumptions (see Section II for more details).

II. RELATED WORK

To combat pollution attacks in network coding, messages at the intermediate nodes should be authenticated before being encoded and forwarded. The messages must go through origin authentication and message authentication to verify their origin and their content. The goal is to achieve authentication even in presence of attackers that can observe the messages flowing through the network and inject selected messages in the network. The success of their attacks depends on their ability in altering the identity information (i.e. impersonation attack) or the message content (i.e. substitution attack) so that intermediate nodes cannot detect it.

To allow authentication to take place, messages are appended at the source either a signature, a MAC or an authentication code (also called tag). The differences between these techniques are that signatures and MACs are proven to be computationally secure while the security of authentication codes is unconditional [9]. Also an authentication tag or a MAC appended to a message can be verified only by intended receivers while anyone can verify a signature with a public verification algorithm. [9]

The schemes proposed for pollution-robust network coding aim for computational security. The scheme in [5] uses a homomorphic hash function. The signature scheme in [6] is based on Weil pairing over elliptic curves while the signature scheme in [7] is based on RSA and the one proposed in [8] on the hardness of the discrete logarithm problem.

III. ORGANIZATION AND CONTRIBUTION

In this paper, we propose an unconditionally secure solution that provides network coding with robustness against pollution attacks. Our solution allows intermediate nodes to verify the message integrity of the packets received and thus to detect and discard the malicious packets that fail the verification.

Our scheme here aims for unconditional security. We rely on information-theoretic strength rather than on problems that are thought to be hard. Unconditional authentication codes have led to the development of multi-receiver authentication codes [10], [11] that are highly relevant in the context of network coding. Multi-receiver authentication codes allow any one of the receivers to verify the integrity and origin of a received message but require the source to be designated. Our scheme is inspired from the (k, V) multi-receiver authentication code proposed in [11] that is robust against a coalition of $k - 1$ malicious receivers amongst V and in which every key can be used to authenticate up to M messages. We define and adapt the use of (k, V) multi-receiver authentication codes to network coding so that intermediate nodes can detect malicious packets without having to decode.

The rest of the paper is organized as follows. In Section IV, we briefly present the network coding model we consider. The contributions of this paper are contained in Section V, where the authentication code is presented, and in Section VI, where it is analyzed. Future work is addressed in the conclusion.

IV. THE NETWORK CODING MODEL

The model of network we consider is an acyclic graph having unit capacity edges, with a source S , which wants to broadcast a set of n messages to T destinations $D_1, \dots, D_T$. Messages are seen as sequences of elements of a finite field $\mathbb{F}_q$. Each edge e of the graph carries a symbol $y(e) \in \mathbb{F}_q$ at a time. For a node v of the graph, the symbols on its outgoing edges are linear combinations of the symbols entering v through its incoming edges. If $x_1, \dots, x_n$ are the symbols to be sent by the source S , we have by induction that on any edge e , $y(e)$ is actually a linear combination of the source symbols, that is $y(e) = \sum_{i=1}^n g_i(e)x_i$, where the coefficients $g_i(e)$ describe the coding operation. The vector $g(e) = [g_1(e), \dots, g_n(e)]$ is called the *global encoding vector* along the edge e . Any destination D_i , $i = 1, \dots, T$, which receives along its at least

n incoming edges $e_1, \dots, e_n$ the symbols

$$\begin{pmatrix} y(e_1) \\ \vdots \\ y(e_n) \end{pmatrix} = \begin{pmatrix} g_1(e_1) & \dots & g_n(e_1) \\ \vdots & & \vdots \\ g_1(e_n) & & g_n(e_n) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ = G_{D_i} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

can recover the source symbols $x_1, \dots, x_n$, assuming that the *transfer matrix* G_{D_i} has rank n , $i = 1, \dots, T$. Destination nodes are assumed to be able to decode the received packets correctly.

We can packetize the symbols $y(e)$ flowing on each edge e into vectors $\mathbf{y}(e) = [y_1(e), \dots, y_N(e)]$, and likewise, the source symbols x_i can be grouped as $\mathbf{x}_i = [x_{i,1}, \dots, x_{i,N}]$, so that the equation at each receiver can be rewritten as

$$\begin{pmatrix} \mathbf{y}(e_1) \\ \vdots \\ \mathbf{y}(e_n) \end{pmatrix} = \begin{pmatrix} g_1(e_1) & \dots & g_n(e_1) \\ \vdots & & \vdots \\ g_1(e_n) & & g_n(e_n) \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} \\ = G_{D_i} \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,N} \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,N} \end{pmatrix}$$

where $\mathbf{x}_1, \dots, \mathbf{x}_n$ are the messages to be sent by the source.

Let v_i be an arbitrary node in the network, with incoming edges $e_{i1}, \dots, e_{ih}$. Note that its received vector is thus

$$\begin{pmatrix} \mathbf{y}(e_{i1}) \\ \vdots \\ \mathbf{y}(e_{ih}) \end{pmatrix} = \begin{pmatrix} g_1(e_{i1}) & \dots & g_n(e_{i1}) \\ \vdots & & \vdots \\ g_1(e_{ih}) & & g_n(e_{ih}) \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}. \quad (1)$$

V. THE AUTHENTICATION CODE

Recall that we have a source S , which wants to multicast a set of n messages $s_1, \dots, s_n$ to T destinations $D_1, \dots, D_T$. Each message s_i is of length l , $s_i = (s_{i,1}, \dots, s_{i,l})$, so that while each symbol $s_{i,j}$ belongs to $\mathbb{F}_q$, we can see the whole message as part of $\mathbb{F}_q^l$. We also assume a set of nodes $R_1, \dots, R_V$ which can verify the authentication. A priori, this set can include the destinations, but can also be larger. Typically we will assume that $V > T$.

A. Set-up and Authentication tag generation

We propose the following authentication scheme:

- 1) **Key generation:** A trusted authority randomly generates $M + 1$ polynomials $P_0(x), \dots, P_M(x) \in \mathbb{F}_q[x]$ and chooses V distinct values $x_1, \dots, x_V \in \mathbb{F}_q$. These polynomials are of degree $k - 1$, and we denote them by

$$P_i(x) = a_{i0} + a_{i1}x + a_{i2}x^2 + \dots + a_{i,k-1}x^{k-1},$$

$$i = 1, \dots, M.$$

- 2) **Key distribution:** The trusted authority gives as private key to the source S the $M + 1$ polynomials

$(P_0(x), \dots, P_M(x))$, and as private key for each verifier R_i the $M + 1$ polynomials evaluated at $x = x_i$, namely $(P_0(x_i), \dots, P_M(x_i))$, $i = 1, \dots, V$. The values $x_1, \dots, x_V$ are made public.

- 3) **Authentication tag:** Let us assume that the source wants to send n messages $s_1, \dots, s_n$. The source computes the following n polynomials

$$A_{s_i}(x) = P_0(x) + s_i P_1(x) + s_i^q P_2(x) \dots + s_i^{q^{(M-1)}} P_M(x),$$

$i = 1, \dots, n$, which form the authentication of each s_i . The packets $\mathbf{x}_i$, $i = 1, \dots, n$ to be actually sent by the source are of the form

$$\mathbf{x}_i = [1, s_i, A_{s_i}(x)], \quad i = 1, \dots, n.$$

Before going on, we make a few remarks on the parameters included in the authentication scheme:

- T is the number of receivers of the messages, V is the number of nodes which can verify the authentication, n is the number of messages sent by the source, M is the number of polynomials used for generating the private keys, k is the degree of each of these polynomials, and finally l is the length of the message which thus belongs to $\mathbb{F}_{q^l}$.
- We will typically take $V > T$, which means that more nodes than just the destinations will check the authentication. We could imagine $V < T$ if we do not even want all the destinations to check the authentication. However our goal is to have enough nodes in the network (though not necessarily all of them) verifying the integrity of the packets to avoid the propagation of polluted packets.
- We have $q \geq V$, since private verification keys are obtained by evaluating the polynomials in x_i , $i = 1, \dots, V$. If $V \geq q$, then we are forced to use some values of $\mathbb{F}_q$ more than once, and the private keys are not unique anymore.
- The choice of the value of M will be discussed below, but we will assume $M \geq n$, to be able to protect with the same key all the messages to be sent, and also $M \leq l$.

Remark 1: Note that while making public the values $x_1, \dots, x_T$ still may help an attacker, we prefer to make them public and prove that actually this does not help the attacker, in order to minimize the amount of secret information given to the nodes.

B. Verification and correctness of the authentication tag

In order to discuss how to check the authentication, let us recall from (1) what is the received vector at a node R_i which

can verify the authentication:

$$\begin{aligned} & \begin{pmatrix} \mathbf{y}(e_{i_1}) \\ \vdots \\ \mathbf{y}(e_{i_h}) \end{pmatrix} \\ &= \begin{pmatrix} g_1(e_{i_1}) & \dots & g_n(e_{i_1}) \\ \vdots & & \vdots \\ g_1(e_{i_h}) & & g_n(e_{i_h}) \end{pmatrix} \begin{pmatrix} 1 & s_1 & A_{s_1}(x) \\ \vdots & \vdots & \vdots \\ 1 & s_n & A_{s_n}(x) \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^n g_j(e_{i_1}) & \sum_{j=1}^n g_j(e_{i_1})s_j & \sum_{j=1}^n g_j(e_{i_1})A_{s_j}(x) \\ \vdots & \vdots & \vdots \\ \sum_{j=1}^n g_j(e_{i_h}) & \sum_{j=1}^n g_j(e_{i_h})s_j & \sum_{j=1}^n g_j(e_{i_h})A_{s_j}(x) \end{pmatrix} \end{aligned}$$

Recall that R_i has a private key given by

$$P_0(x_i), \dots, P_M(x_i).$$

For each incoming edge e_k , the node can thus, on the one hand, compute

$$\begin{aligned} & P_0(x_i) \sum_{j=1}^n g_j(e_k) \\ & P_1(x_i) \sum_{j=1}^n g_j(e_k)s_j \\ & P_2(x_i) \left(\sum_{j=1}^n g_j(e_k)s_j \right)^q = P_2(x_i) \sum_{j=1}^n (g_j(e_k)s_j)^q \\ & = P_2(x_i) \sum_{j=1}^n g_j(e_k)s_j^q \\ & \vdots \\ & P_M(x_i) \left(\sum_{j=1}^n g_j(e_k)s_j \right)^{q^{(M-1)}} = P_M(x_i) \sum_{j=1}^n (g_j(e_k)s_j)^{q^{(M-1)}} \\ & = P_M(x_i) \sum_{j=1}^n g_j(e_k)s_j^{q^{M-1}}, \end{aligned}$$

and on the other hand, it can evaluate

$$\sum_{j=1}^n g_j(e_k)A_{s_j}(x)$$

in x_i , which is public. This yields

$$\begin{aligned} & \sum_{j=1}^n g_j(e_k)A_{s_j}(x_i) \\ &= \sum_{j=1}^n g_j(e_k)(P_0(x_i) + s_j P_1(x_i) + s_j^q P_2(x_i) + \dots \\ & \quad + s_j^{q^{(M-1)}} P_M(x_i)) \\ &= \sum_{j=1}^n g_j(e_k)P_0(x_i) + \sum_{j=1}^n g_j(e_k)s_j P_1(x_i) + \dots \\ & \quad + \sum_{j=1}^n g_j(e_k)s_j^{q^{(M-1)}} P_M(x_i). \end{aligned}$$

The node R_i accepts a packet on its incoming edge e_k if the two computations coincide, which they do if there is no alteration of the protocol.

Example 1: Consider the case where we have a very small network where network coding is achieved over $\mathbb{F}_2$, and where we have $V = 2$ nodes which verify the authentication. The source S wants to send two messages $s_1, s_2 \in \mathbb{F}_2^3$, that is $s_1 = (s_{1,1}, s_{1,2}, s_{1,3})$ and $s_2 = (s_{2,1}, s_{2,2}, s_{2,3})$ with $s_{i,j} \in \mathbb{F}_2 = \{0, 1\}$. During the key generation and distribution, we have that

- the source is given the 3 polynomials $P_0(x) = a_{00} + a_{01}x$, $P_1(x) = a_{10} + a_{11}x$, and $P_2(x) = a_{20} + a_{21}x$.
- the values x_1 and x_2 are made public,
- the node R_1 receives the secret values $P_0(x_1), P_1(x_1), P_2(x_1)$ as its private key.
- the node R_2 receives the secret values $P_0(x_2), P_1(x_2), P_2(x_2)$ as its private key.

The source computes two authentication tags:

$$\begin{aligned} A_{s_1}(x) &= P_0(x) + s_1 P_1(x) + s_1^2 P_2(x) \\ &= (a_{00} + a_{10}s_1 + a_{20}s_1^2) + x(a_{01} + a_{11}s_1 + a_{21}s_1^2) \\ &=: b_{10} + x b_{11} \\ A_{s_2}(x) &= P_0(x) + s_2 P_1(x) + s_2^2 P_2(x) \\ &= (a_{00} + a_{10}s_2 + a_{20}s_2^2) + x(a_{01} + a_{11}s_2 + a_{21}s_2^2) \\ &=: b_{20} + x b_{21} \end{aligned}$$

The two packets to be sent are

$$\begin{aligned} \mathbf{x}_1 &= [1, s_1, A_{s_1}(x)] = [1, s_{1,1}, s_{1,2}, s_{1,3}, b_{10}, b_{11}] \in (\mathbb{F}_2)^{10} \\ \mathbf{x}_2 &= [1, s_2, A_{s_2}(x)] = [1, s_{2,1}, s_{2,2}, s_{2,3}, b_{20}, b_{21}] \in (\mathbb{F}_2)^{10} \end{aligned}$$

Suppose that the first node R_1 has two input edges e_1, e_2 . Its received vector is thus

$$\begin{aligned} \begin{pmatrix} \mathbf{y}(e_1) \\ \mathbf{y}(e_2) \end{pmatrix} &= \begin{pmatrix} g_1(e_1) & g_2(e_1) \\ g_1(e_2) & g_2(e_2) \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} g_1(e_1) & g_2(e_1) \\ g_1(e_2) & g_2(e_2) \end{pmatrix} \begin{pmatrix} 1 & s_1 & A_{s_1}(x) \\ 1 & s_2 & A_{s_2}(x) \end{pmatrix} = \\ &= \begin{pmatrix} g_1(e_1) + g_2(e_1), g_1(e_1)s_1 + g_2(e_1)s_2, g_1(e_1)A_{s_1}(x) + g_2(e_1)A_{s_2}(x) \\ g_1(e_2) + g_2(e_2), g_1(e_2)s_1 + g_2(e_2)s_2, g_1(e_2)A_{s_1}(x) + g_2(e_2)A_{s_2}(x) \end{pmatrix} \end{aligned}$$

The data which is public is x_1, x_2 . Using x_1 and its private key $P_0(x_1), P_1(x_1), P_2(x_1)$, R_1 can compute from $\mathbf{y}(e_1)$ the following things:

$$(g_1(e_1) + g_2(e_1))P_0(x_1), P_1(x_1)(g_1(e_1)s_1 + g_2(e_1)s_2),$$

and

$$\begin{aligned} &P_2(x_1)(g_1(e_1)s_1 + g_2(e_1)s_2)^2 \\ &= P_2(x_1)(g_1(e_1)^2 s_1^2 + g_2(e_1)^2 s_2^2) \\ &= P_2(x_1)(g_1(e_1)s_1^2 + g_2(e_1)s_2^2) \end{aligned}$$

which gives

$$(g_1(e_1) + g_2(e_1))P_0(x_1) + P_1(x_1)(g_1(e_1)s_1 + g_2(e_1)s_2) + P_2(x_1)(g_1(e_1)s_1^2 + g_2(e_1)s_2^2).$$

Since R_1 has also received $g_1(e_1)A_{s_1}(x) + g_2(e_1)A_{s_2}(x)$, R_1 can check whether

$$\begin{aligned} &g_1(e_1)A_{s_1}(x_1) + g_2(e_1)A_{s_2}(x_1) = \\ &g_1(e_1)P_0(x_1) + g_2(e_1)P_0(x_1) + P_1(x_1)(g_1(e_1)s_1 + g_2(e_1)s_2) \\ &+ P_2(x_1)(g_1(e_1)s_1^2 + g_2(e_1)s_2^2). \end{aligned}$$

A similar check can be done on e_2 .

VI. ANALYSIS OF THE SCHEME

For the analysis of the scheme, two cases are to be considered. To start with, we consider the case where one malicious node is trying to make a substitution attack, that is, to send a fake packet such that a node which checks for authentication will actually accept the faked authentication tag. The case of a group of malicious nodes is considered later on.

A. Against one malicious node

Suppose that a malicious node v_i has h incoming edges. Its received vector is thus

$$\begin{aligned} &\begin{pmatrix} \mathbf{y}(e_{i_1}) \\ \vdots \\ \mathbf{y}(e_{i_h}) \end{pmatrix} = \begin{pmatrix} g_1(e_{i_1}) & \dots & g_n(e_{i_1}) \\ \vdots & & \vdots \\ g_1(e_{i_h}) & \dots & g_n(e_{i_h}) \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} \\ &= \begin{pmatrix} g_1(e_{i_1}) & \dots & g_n(e_{i_1}) \\ \vdots & & \vdots \\ g_1(e_{i_h}) & \dots & g_n(e_{i_h}) \end{pmatrix} \begin{pmatrix} 1 & s_1 & A_{s_1}(x) \\ \vdots & & \vdots \\ 1 & s_n & A_{s_n}(x) \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^n g_j(e_{i_1}) & \sum_{j=1}^n g_j(e_{i_1})s_j & \sum_{j=1}^n g_j(e_{i_1})A_{s_j}(x) \\ \vdots & \vdots & \vdots \\ \sum_{j=1}^n g_j(e_{i_h}) & \sum_{j=1}^n g_j(e_{i_h})s_j & \sum_{j=1}^n g_j(e_{i_h})A_{s_j}(x) \end{pmatrix}. \end{aligned}$$

If we write

$$A_{s_j}(x) = b_{j0} + b_{j1}x + \dots + b_{j,k-1}x^{k-1},$$

we have that for all incoming edges e_m

$$\begin{aligned} &\sum_{j=1}^n g_j(e_m)A_{s_j}(x) \\ &= \sum_{j=1}^n g_j(e_m)(b_{j0} + b_{j1}x + \dots + b_{j,k-1}x^{k-1}) \\ &= c_{m0} + c_{m1}x + \dots + c_{m,k-1}x^{k-1}, \end{aligned}$$

where

$$c_{mi} = \sum_{j=1}^n g_j(e_m)b_{ji}x^i.$$

Thus, the node actually knows c_{mi} , $i = 1, \dots, k-1$, for every incoming edge e_m , $m = i_1, \dots, i_h$. Upon reception of its incoming vector, the malicious node can obtain the following system of linear equations:

$$AG = C \quad (2)$$

where

$$A = \begin{pmatrix} a_{0,0} & a_{1,0} & a_{M,0} \\ a_{0,1} & a_{1,1} & a_{M,1} \\ \vdots & \vdots & \vdots \\ a_{0,k-1} & a_{1,k-1} & a_{M,k-1} \end{pmatrix},$$

$$G = \begin{pmatrix} \sum_{j=1}^n g_j(e_{i_1}) & \cdots & \sum_{j=1}^n g_j(e_{i_h}) \\ \sum_{j=1}^n g_j(e_{i_1})s_j & \cdots & \sum_{j=1}^n g_j(e_{i_h})s_j \\ \vdots & \vdots & \vdots \\ \sum_{j=1}^n g_j(e_{i_1})s_j^{q^{(M-1)}} & \cdots & \sum_{j=1}^n g_j(e_{i_h})s_j^{q^{(M-1)}} \end{pmatrix},$$

$$C = \begin{pmatrix} c_{10} & \cdots & c_{h,0} \\ \vdots & \vdots & \vdots \\ c_{1,k-1} & \cdots & c_{h,k-1} \end{pmatrix},$$

with the $k \times (M+1)$ matrix A containing the coefficients of the private key, and both the $(M+1) \times h$ matrix G containing the network coding coefficients, and the $k \times h$ matrix C known to the malicious node.

Lemma 1: For the proposed authentication to be secure, we at least need $(M+1) > h$.

Proof: We have a linear system of equations of the form

$$AG = C$$

where A contains the unknown (the secret coefficients that the adversary would like to find). This is a system of kh equations in $k(M+1)$ unknown. If $h \geq M+1$, then the matrix G is of rank $M+1$, meaning that some of the columns are linearly dependent, and we get $k(M+1)$ equations for $k(M+1)$ unknown. ■

Corollary 1: We need to choose $M > \max(n-1, h-1)$.

Proof: This follows from that fact that we at least need one tag per message sent. ■

Note that we do not take into account here the fact that the malicious node may have knowledge of one private key (this is the scenario where a node able to verify is actually corrupted). This will be treated as a particular case of attack by a group of corrupted nodes.

Example 2: We go on with the previous example. Suppose a node R_1 has received the following vector:

$$\begin{pmatrix} \mathbf{y}(e_1) \\ \mathbf{y}(e_2) \end{pmatrix}$$

with

$$\mathbf{y}(e_1) = (g_1(e_1) + g_2(e_1), g_1(e_1)s_1 + g_2(e_1)s_2, g_1(e_1)A_{s_1}(x) + g_2(e_1)A_{s_2}(x))$$

and

$$\mathbf{y}(e_2) = (g_1(e_2) + g_2(e_2), g_1(e_2)s_1 + g_2(e_2)s_2, g_1(e_2)A_{s_2}(x) + g_2(e_2)A_{s_2}(x)).$$

This time, the node is malicious, and instead of checking the authentication tag, it actually wants to make a substitution attack.

Since we have that

$$\begin{aligned} A_{s_1}(x) &= P_0(x) + s_1P_1(x) + s_1^2P_2(x) \\ &= (a_{00} + a_{10}s_1 + a_{20}s_1^2) + x(a_{01} + a_{11}s_1 + a_{21}s_1^2) \\ &=: b_{10} + xb_{11} \\ A_{s_2}(x) &= P_0(x) + s_2P_1(x) + s_2^2P_2(x) \\ &= (a_{00} + a_{10}s_2 + a_{20}s_2^2) + x(a_{01} + a_{11}s_2 + a_{21}s_2^2) \\ &=: b_{20} + xb_{21}, \end{aligned}$$

we can rewrite

$$\begin{aligned} &g_1(e_1)A_{s_1}(x) + g_2(e_1)A_{s_2}(x) \\ &= g_1(e_1)(b_{10} + xb_{11}) + g_2(e_1)(b_{20} + xb_{21}) \\ &= g_1(e_1)b_{10} + g_2(e_1)b_{20} + x(g_1(e_1)b_{11} + g_2(e_1)b_{21}). \end{aligned}$$

The malicious node thus knows

$$c_{10} = g_1(e_1)b_{10} + g_2(e_1)b_{20}, \quad c_{11} = g_1(e_1)b_{11} + g_2(e_1)b_{21}.$$

Alternatively, we can rewrite

$$\begin{aligned} &g_1(e_1)A_{s_1}(x) + g_2(e_1)A_{s_2}(x) \\ &= g_1(e_1)(a_{00} + a_{10}s_1 + a_{20}s_1^2) + g_2(e_1)(a_{00} + a_{10}s_2 + a_{20}s_2^2) + xg_1(e_1)(a_{01} + a_{11}s_1 + a_{21}s_1^2) \\ &\quad + xg_2(e_1)(a_{01} + a_{11}s_2 + a_{21}s_2^2) \\ &= a_{00}(g_1(e_1) + g_2(e_1)) + a_{10}(g_1(e_1)s_1 + g_2(e_1)s_2) \\ &\quad + a_{20}(g_1(e_1)s_1^2 + g_2(e_1)s_2^2) + x[a_{01}(g_1(e_1) + g_2(e_1)) \\ &\quad + a_{11}(g_1(e_1)s_1 + g_2(e_1)s_2) + a_{21}(g_1(e_1)s_1^2 + g_2(e_1)s_2^2)] \end{aligned}$$

Since the malicious node knows $g_1(e_1) + g_2(e_1)$, $g_1(e_1)s_1 + g_2(e_1)s_2$ and $g_1(e_1)s_1^2 + g_2(e_1)s_2^2$, and by iterating the computations for the second incoming edge, it can form the following system of linear equations:

$$\begin{pmatrix} a_{0,0} & a_{1,0} & a_{2,0} \\ a_{0,1} & a_{1,1} & a_{2,1} \end{pmatrix} G = \begin{pmatrix} c_{10} & c_{2,0} \\ c_{1,1} & c_{2,1} \end{pmatrix}$$

where

$$G = \begin{pmatrix} g_1(e_1) + g_2(e_1) & g_1(e_2) + g_2(e_2) \\ g_1(e_1)s_1 + g_2(e_1)s_2 & g_1(e_2)s_1 + g_2(e_2)s_2 \\ g_1(e_1)s_1^2 + g_2(e_1)s_2^2 & g_1(e_2)s_1^2 + g_2(e_2)s_2^2 \end{pmatrix}.$$

We can see from this example that if only two polynomials P_0 and P_1 were used to create the authentication tag, then the matrix G would be a 2×2 matrix, and thus could be very likely invertible, thus allowing the malicious node to recover the secret coefficients of the source private key, although the node cannot decode the message.

B. Against a group of malicious nodes

Let us now assume that K nodes $v_1, \dots, v_K$ collaborate to make a substitution attack. Each of them obtains a vector of data from the network, and can thus collect a system of linear equations of the form

$$AG_i = C_i, \quad i = 1, \dots, K,$$

as explained in (2). All together, this gives a new system

$$A[G_1 \ G_2 \ \dots \ G_K] = [C_1, \dots, C_K].$$

Following Lemma 1, a safe (but maybe loose) bound for the authentication to stay secure is to require $(M + 1) \geq h_1 + \dots + h_K$, where h_i denotes the number of incoming edges at node v_i , $i = 1, \dots, K$.

We now take into account the fact that some of the nodes who are given the private keys to check the authentication could be corrupted. Since we assume a group of K malicious nodes, let us furthermore assume the worst case, namely that all of them actually possess a private key $(P_0(x_i), \dots, P_M(x_i))$, where i belongs to a subset of cardinality K of $\{1, \dots, T\}$. Without loss of generality we can assume that i goes from 1 to K .

Since the values $x_1, \dots, x_T$ are made public, the group of adversaries can actually build another system of linear equations which exploits their knowledge of the private keys, namely

$$XA = P$$

where

$$X = \begin{pmatrix} 1 & x_1 & \dots & x_1^{k-1} \\ 1 & x_2 & \dots & x_2^{k-1} \\ \dots & \dots & \dots & \dots \\ 1 & x_K & \dots & x_K^{k-1} \end{pmatrix},$$

as before

$$A = \begin{pmatrix} a_{0,0} & a_{1,0} & \dots & a_{M,0} \\ a_{0,1} & a_{1,1} & \dots & a_{M,1} \\ \dots & \dots & \dots & \dots \\ a_{0,k-1} & a_{1,k-1} & \dots & a_{M,k-1} \end{pmatrix}$$

and

$$P = \begin{pmatrix} P_0(x_1) & P_1(x_1) & \dots & P_M(x_1) \\ P_0(x_2) & P_1(x_2) & \dots & P_M(x_2) \\ \dots & \dots & \dots & \dots \\ P_0(x_K) & P_1(x_K) & \dots & P_M(x_K) \end{pmatrix},$$

where the $K \times k$ matrix X contains the public key values, the $k \times (M + 1)$ matrix A contains the coefficients of the private key to be found by the group of attackers, and the $K \times (M + 1)$ matrix P contains the private keys of the corrupted nodes.

Since the polynomials $P_0, \dots, P_M$ have degree $k - 1$, it is clear that K can be at most $k - 1$, otherwise from the knowledge of only the private and public keys, the group of attackers can recover the source's private key, i.e., they can solve the system of equations and recover A .

We are now left to prove that even by putting together the information given by the private keys and the one gathered from all the received vectors, the group of adversaries still cannot do better than guess the source private key.

Lemma 2: There exist q matrices A with coefficients in $\mathbb{F}_q$ such that

$$AG = C, \quad XA = P,$$

where the matrix dimensions are $k \times (M + 1)$ for A , $(M + 1) \times H$ for G , $k \times H$ for C , $(k - 1) \times k$ for X and $(k - 1) \times (M + 1)$ for P , and $H = h_1 + \dots + h_K$ is the aggregated number of incoming edges for all corrupted nodes.

Proof: Recall that the matrix G is of the form

$$\begin{pmatrix} \sum_{j=1}^n g_j(e_{i_1}) & \dots & \sum_{j=1}^n g_j(e_{i_H}) \\ \sum_{j=1}^n g_j(e_{i_1})s_j & \dots & \sum_{j=1}^n g_j(e_{i_H})s_j \\ \dots & \dots & \dots \\ \sum_{j=1}^n g_j(e_{i_1})s_j^{q^{(M-1)}} & \dots & \sum_{j=1}^n g_j(e_{i_H})s_j^{q^{(M-1)}} \end{pmatrix}.$$

Note that for any invertible matrix D , we have that

$$AG = C \iff AGD = CD,$$

and there exists a invertible matrix D such that GD is of the Vandermonde like form

$$\begin{pmatrix} 1 & \dots & 1 \\ \gamma_1 & \dots & \gamma_H \\ \gamma_1^q & \dots & \gamma_H^q \\ \dots & \dots & \dots \\ \gamma_1^{q^{M-1}} & \dots & \gamma_H^{q^{M-1}} \end{pmatrix}. \quad (3)$$

In particular, if all the coefficients of the first row of G are non zero, we can take D to be

$$D = \text{diag}((\sum_{j=1}^n g_j(e_{i_1}))^{-1}, \dots, (\sum_{j=1}^n g_j(e_{i_H}))^{-1}).$$

Thus we can assume that we want to solve the system

$$AG = C, \quad XA = P$$

with G is of the form (3).

Note first that it is enough to look at the homogeneous system of equations given by

$$AG = 0, \quad XA = 0. \quad (4)$$

To finish the proof, we slightly generalize the proof techniques of [11], where the main idea consists of seeing the matrix A as containing the coefficients of a polynomial $f(x, y)$ in two indeterminates x and y , such that

$$f(x, y) = (1, x, \dots, x^{k-1})A \begin{pmatrix} 1 \\ y \\ \vdots \\ y^{q^{M-1}} \end{pmatrix}.$$

Now, take such a polynomial $f(x, y)$ whose roots are $x_1, \dots, x_{k-1}$ and $\gamma_1, \dots, \gamma_H$. Note that the matrix multiplication

$$XA = 0$$

exactly means evaluating $f(x, y)$ in $x = x_1, \dots, x_k$ and finding zero with each of these values for all y (which exactly means that $x = x_1, \dots, x_k$ are roots of the polynomial). Similarly we get that

$$AG = 0$$

in evaluating $f(x, y)$ in $y = \gamma_1, \dots, \gamma_H$ for all x . Thus the matrix containing the coefficients of f gives one suitable matrix A satisfying our conditions. Since matrices obtained by multiplication with a scalar in $\mathbb{F}_q$ also satisfy the equation, this gives q suitable matrices. ■

Proposition 1: The above scheme is a (k, V) unconditionally secure multi-receiver authentication code against a coalition of up to $k-1$ adversaries among V receivers in which every key can be used to authenticate up to M messages.

Proof: We mainly follow the proof given in [11]. To make a substitution attack, the malicious $k-1$ receivers want to generate a message such that it is accepted as authentic by the receiver R_k that they are trying to cheat. However, for that, they need to guess its secret key $[P_0(x_k), \dots, P_M(x_k)]$, and choose a polynomial $\tilde{A}_s(x)$ such that

$$\tilde{A}_s(x_k) = P_0(x_k) + s^q P_1(x_k) + \dots + s^{q(M-1)} P_M(x_k)$$

for some message s . We also know from the above lemma that there are q matrices satisfying the equation (4), meaning that there are q different $(M+1)$ -tuple of polynomials $(\tilde{P}_0(x), \dots, \tilde{P}_M(x))$ likely to be the source's private key, from which we can show that there are q equally likely private keys for R_k . Thus the probability of the $k-1$ receivers to guess $A(x_k)$ correctly is $1/q$. ■

Example 3: We continue our example, where one malicious node R_1 is trying to make a substitution attack. We have already seen that after receiving its incoming vector, it gets the following system of linear equations:

$$\begin{pmatrix} a_{0,0} & a_{1,0} & a_{2,0} \\ a_{0,1} & a_{1,1} & a_{2,1} \end{pmatrix} G = \begin{pmatrix} c_{10} & c_{2,0} \\ c_{1,1} & c_{2,1} \end{pmatrix}$$

where

$$G = \begin{pmatrix} g_1(e_1) + g_2(e_1) & g_1(e_2) + g_2(e_2) \\ g_1(e_1)s_1 + g_2(e_1)s_2 & g_1(e_2)s_1 + g_2(e_2)s_2 \\ g_1(e_1)s_1^2 + g_2(e_1)s_2^2 & g_1(e_2)s_1^2 + g_2(e_2)s_2^2 \end{pmatrix}.$$

Now if furthermore R_1 has a private key $[P_0(x_1), P_1(x_1), P_2(x_1)]$, this helps it to know that

$$(1, x_1) \begin{pmatrix} a_{0,0} & a_{1,0} & a_{2,0} \\ a_{0,1} & a_{1,1} & a_{2,1} \end{pmatrix} = (P_0(x_1), P_1(x_1), P_2(x_1)).$$

Let us assume for this example that the first row of G has non-zero coefficients, so that both coefficients are invertible. We set

$$\begin{aligned} \gamma_1 &= (g_1(e_1)s_1 + g_2(e_1)s_2)(g_1(e_1) + g_2(e_1))^{-1} \\ \gamma_2 &= (g_1(e_2)s_1 + g_2(e_2)s_2)(g_1(e_2) + g_2(e_2))^{-1} \end{aligned}$$

and we can rewrite G as

$$\begin{pmatrix} g_1(e_1) + g_2(e_1) & g_1(e_2) + g_2(e_2) \\ \gamma_1(g_1(e_1) + g_2(e_1)) & \gamma_2(g_1(e_2) + g_2(e_2)) \\ \gamma_1^2(g_1(e_1) + g_2(e_1)) & \gamma_2^2(g_1(e_2) + g_2(e_2)) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \gamma_1 & \gamma_2 \\ \gamma_1^2 & \gamma_2^2 \end{pmatrix} \begin{pmatrix} g_1(e_1) + g_2(e_1) & 0 \\ 0 & g_1(e_2) + g_2(e_2) \end{pmatrix}.$$

It is a straightforward computation to check that the matrices

$$rA = r \begin{pmatrix} -x_1\gamma_1\gamma_2 & x_1\gamma_1 + x_1\gamma_2 & -x_1 \\ \gamma_1\gamma_2 & -\gamma_1 - \gamma_2 & 1 \end{pmatrix}, \quad r \in \mathbb{F}_q$$

satisfy the system of equations $AG = \mathbf{0}$, $XA = \mathbf{0}$, where $X = (1, x_1)$.

VII. CONCLUSION

In this paper, we have proposed an unconditionally secure authentication scheme that provides network coding with message integrity protection and source authentication. The resulting scheme offers robustness against pollution attacks from outsiders and from $k-1$ insiders. Our solution allows the source to generate authentication tags for up to M messages with the same key and the intermediate nodes to verify the authentication tags of the packets received and thus to detect and discard the malicious packets that fail the verification.

Future work will involve optimization of the parameters involved in the authentication scheme for a more efficient solution. Another aspect to consider in the future is to offer more flexibility over the sender as the scheme proposed here requires the sender to be designated.

ACKNOWLEDGMENT

The work of Frédérique Oggier is supported in part by the Nanyang Technological University under Research Grant M58110049. Hanane Fathi would like to acknowledge the support of the National Institute of Advanced Industrial Science and Technology, Japan. Most of the ideas of this work were discussed while both authors were visiting the AIST Research Center for Information Security, Tokyo, Japan.

REFERENCES

- [1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network Information Flow", *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204- 1216, July 2000.
- [2] S.-Y.R. Li, and R.W. Yeung, "Linear Network Coding", *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371- 381, February 2003.
- [3] R. Koetter, and M. Medard, "An algebraic approach to network coding", *IEEE International Symposium on Information Theory*, 2001.
- [4] P. Chou, Y. Wu and K. Jain, "Practical Network Coding", *Allerton Conference on Communication, Control and Computing*, 2003.
- [5] C. Gkantsidis and P. Rodriguez, "Cooperative Security for Network Coding File Distribution", *IEEE INFOCOM*, 2006.
- [6] D. Charles, K. Jain, and K. Lauter, "Signatures for Network Coding", *Conference on Information Sciences and Systems*, 2006.
- [7] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Signature-based Scheme for Securing Network Coding against Pollution Attacks", *IEEE INFOCOM*, 2008.
- [8] F. Zhao, T. Kalker, M. Medard, and K.J. Han, "Signatures for Content Distribution with Network Coding", *IEEE International Symposium on Information Theory*, 2007.
- [9] D.R. Stinson, *Cryptography: Theory and Practice*, CRC Press, Florida, 1995.
- [10] Y. Desmedt, Y. Frankel, and M. Yung, "Multi-Receiver/Multi-Sender Network Security: Efficient Authenticated Multicast/Feedback", *IEEE INFOCOM*, 1992.
- [11] R. Safavi-Naini, and H. Wang, "New results on multi-receiver authentication codes", *Eurocrypt'98*, LNCS 1403, pp.527-541, 1998.