I.J. Computer Network and Information Security, **2011, 2, 57-63**
Published Online March 2011 in MECS (http://www.mecs-press.org/)

The Research of Unconditionally Secure Authentication Code For Multi-Source Network Coding

Hong Yang
Sch. of Comp. Sci. and Tech., Wuhan Uni. of Tech.， Wuhan, China
Email: yhheghy@whut.edu.cn

Mingxi Yang*
Sch. of Comp. Sci. and Tech., Wuhan Uni. of Tech.， Wuhan, China
Email: yangmx@whut.edu.cn

***Abstract*—in a network system, network coding allows intermediate nodes to encode the received messages before forwarding them, thus network coding is vulnerable to pollution attacks. Besides, the attacks are amplified by the network coding process with the result that the whole network maybe polluted. In this paper, we proposed a novel unconditionally secure authentication code for multi-source network coding, which is robust against pollution attacks. For the authentication scheme based on theoretic strength, it is robust against those attackers that have unlimited computational resources, and the intermediate nodes therein can verify the integrity and origin of the encoded messages received without having to decode them, and the receiver nodes can check them out and discard the messages that fail the verification. By this way, the pollution is canceled out before reaching the destinations.**

Index Terms -secure network coding;multi-source; pollution attack;authentication code

I. INTRODUCTION

In a traditional communication network, the messages are transmitted from the source to the destination via intermediate nodes. Network coding was first proposed by Ahlswede et al. [1] in order to maximize the throughput of multicast networks, intermediate nodes not only can store and forward the messages, but also can encode the received messages before forwarding them. Li et al. [2] showed that linear coding suffices to achieve the max-flow from the source to each receiving node in multicast network, where intermediate nodes generate outing messages as linear combinations of their incoming messages. With the application of the network coding, the usage of network resources was improved. So the network coding was widely used.

However, as network coding allows intermediate nodes to encode the received messages, the result is that network coding is very vulnerable to pollution attacks. Pollution attacks, which consist of injecting malicious messages in the network. The malicious messages may come from the modification of received messages by a malicious inter-mediate node or from the creation of bogus messages by an outside adversary. As a result, with using network coding, the detection for integrity and origin of the messages received is very important. For using unconditionally secure authentication code to prevent pollution attacks, the main innovation of our scheme is that it can be used for multi-source network coding systems. Our scheme is based on the method [9] of Frederique et al, who proposed an authentication code against pollution attacks for single source network coding.

II. BACKGROUND

A. Secure network coding

As network coding allows intermediate nodes to encode the received messages, the result is that network coding is very vulnerable to pollution attacks. Pollution attacks, which consist of injecting malicious messages in the network. If the networks don't have the detection for integrity and origin of the received messages, the polluted messages can quickly propagate into the whole network and infect a large proportion of messages, because they will be transmitted by the downstream nodes. So the secure network coding is very essential. There are two methods to design secure network coding, one is based on computational hypothesis, and the other is on theoretic strength. Gkantsidis et al. proposed a scheme [3] for network-coded content distribution allows intermediate nodes to detect malicious messages injected in the network; it uses a homomorphism hash function to generate hash values of the encoded blocks of data. While it requires fresh keys for each file, so the scheme is not practical. Charles et al. designed a homomorphism signature scheme [4] based on Weil pairing over elliptic curves, but the idea is conditionally secure. Zhao et al. used a standard signature scheme [5] based on the hardness of the discrete logarithm problem, besides, it also requires fresh keys for each file, and it can't support multi-source network coding. All in all, the previous expatiation methods mainly relay on computational hypothesis, these schemes are conditionally secure; besides, they can't support multi-source situation. While the idea based on theoretic strength, unconditionally

* Corresponding Author

secure authentication code, provide another method to design secure network coding which is robust against an attacker even it has unlimited computational resources.

B. unconditionally secure authentication code

In order to prevent pollution attacks, previous methods about secure network coding, mainly based on computational hypothesis, while the idea of designing secure network coding on theoretic strength is proposed less. So the way of theoretic strength provides another method to achieve secure network coding. Unconditionally secure authentication code promotes the development of multi-receiver authentication code [6] [7] [8]. Frederique et al. proposed a method [9] to prevent pollution attacks for single source node network coding, the scheme introduces unconditionally secure authentication code in multicast network, and it is robust against pollution attacks. Intermediate nodes can verify the integrity and origin of the messages received without having to decode the encoded messages, and will discard the messages that fail the verification. By this way, the pollution is canceled out before reaching the destinations.

III. Multi-source network coding model

Yan et al. [10] proposed a multi-source network coding model example in multi-source network coding situation, each source node transmit message separately. The multi-source network is modeled by a directed graph $G = (E, V)$, where E is the set of links and V is the set of vertices in the network. Suppose there are n source nodes, each source node transmit only one message vector to the intermediate node. In this situation, each edge of the graph carries a symbol $f(e) \in \mathbb{F}_q$ at a time. For a node of the graph, the symbols on its outgoing edges are linear combinations. Thus to any receiver node, if it gets message vector $t_1 \ldots t_n$ from n source nodes, then it has the following expression on each edge;

$$f(e) = \sum_{i=1}^{n} g_i(e) t_i \quad (1)$$

Where the coefficients $g_i(e)$ describes the coding operation. The vector $g(e)=[g_1(e)\ldots g_n(e)]$ is thus called the global encoding vector along the edge e. So to a receiver node, if it gets message from n source nodes, with it have n incoming edges, there is a following matrix equation:

$$\begin{pmatrix} f(e_1) \\ \vdots \\ f(e_n) \end{pmatrix} = \begin{pmatrix} g_1(e_1) & \cdots & g_n(e_1) \\ \vdots & \vdots & \vdots \\ g_1(e_n) & \cdots & g_n(e_n) \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix} = G_D \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix} \quad (2)$$

At the same time, the symbols $f(e)$ flowing on each edge e can be packetized into vectors $y(e)=[f_1(e),.,f_N(e)]$, and likewise, the message vector t_i from each source also can be grouped as $x_i=[t_{i,1},.,t_{i,N}]$. So the above equation can be rewritten as:

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

$$= G_D \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,N} \end{pmatrix} \quad (3)$$

Where $x_1, x_2, \ldots, x_n$ is each source node which sends one message a time to the receiver node. So to any node v_i in the multi-source network, with its incoming edges $e_{i1}\ldots e_{ih}$. Each source node is sends one message a time to the receiver node. So to any node v_i in the multi-source network, with its incoming edges $e_{i1}\ldots e_{ih}$, it has below matrix equation:

$$\begin{pmatrix} y(e_{i1}) \\ \vdots \\ y(e_{ih}) \end{pmatrix} = \begin{pmatrix} g_1(e_{i1}) & \cdots & g_n(e_{i1}) \\ \vdots & \vdots & \vdots \\ g_1(e_{ih}) & \cdots & g_n(e_{ih}) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad (4)$$

IV. Unconditionally Secure Authentication Code For Multi-source Network Coding

A. Proposed authentication scheme

1) Private key generation

A trusted authority randomly generates polynomials for each source node, to source node S_1; it has M+1 polynomials

$$P_0^1(x), \ldots, P_M^1(x),$$

And likewise, source node S_n has M+1 polynomials

$$P_0^n(x), \ldots, P_M^n(x),$$

And choose V difference variants $x_1 \ldots x_V \in \mathbb{F}_q$, these polynomials are of degree k-1, the specific situation is made as follows:

$$P_i^1(x) = a_{i0}^1 + a_{i1}^1 x + a_{i2}^1 x^2 + \ldots + a_{i,k-1}^1 x^{k-1} \quad (5)$$

$$\vdots$$

$$P_i^n(x) = a_{i0}^n + a_{i1}^n x + a_{i2}^n x^2 + \ldots + a_{i,k-1}^n x^{k-1} \quad (6)$$

Where i=1…M

2) Private key distribution

The trusted authority gives as private key to each source node, for source node S_1, its private key is $(P_0^1(x), \ldots, P_M^1(x))$. And likewise, the private key for source node S_n is $(P_0^n(x), \ldots, P_M^n(x))$. At the same time, the authority distribute private key for V verifier nodes. Suppose R_i is the ith verifier, then its private key is defined as follows

$$(\lambda_0^1 P_0^1(x_i) + \lambda_0^2 P_0^2(x_i) + \ldots + \lambda_0^n P_0^n(x_i)), \ldots$$

$$(\lambda_M^1 P_M^1(x_i) + \lambda_M^2 P_M^2(x_i) + \ldots + \lambda_M^n P_M^n(x_i))$$

3) Authentication code generation:

Suppose such a situation that each source node sends message vector one time, the sequence is made as $a_1,\ldots,a_n$, each message has the length l, then compute the following polynomial:

$$A_{ai}(x) = [\lambda_0^1 P_0^1(x) + \lambda_0^2 P_0^2(x) + \ldots + \lambda_0^n P_0^n(x)]$$

$$+ a_i[\lambda_1^1 P_1^1(x) + \lambda_1^2 P_1^2(x) + ... + \lambda_1^n P_1^n(x)]$$
$$+ a_i^q[\lambda_2^1 P_2^1(x) + \lambda_2^2 P_2^2(x) + ... + \lambda_2^n P_2^n(x)] +.....+$$
$$a_i^{q^{(M-1)}}[\lambda_M^1 P_M^1(x) + \lambda_M^2 P_M^2(x) + ... + \lambda_M^n P_M^n(x)] \quad (7)$$

Which forms the authentication code of each message, i=1…*n*

4) Encoded message transimittance

In our multi-source network, as previous description, to a verifier node R_i, if it has *n* incoming edges, gets the message sequence from the source nodes.

As $a_1 \ldots a_n$, thus the final formation of the packet likes this: $x_i=[1,a_i,A_{ai}(x)]$, $i=1\ldots n$, so it can write as follows:

$$\begin{pmatrix} y(e_{i1}) \\ \vdots \\ y(e_{in}) \end{pmatrix} = \begin{pmatrix} g_1(e_{i1}) & \cdots & g_n(e_{i1}) \\ \vdots & \vdots & \vdots \\ g_1(e_{in}) & \cdots & g_n(e_{in}) \end{pmatrix} \begin{pmatrix} 1 & a_1 & A_{a1}(x) \\ \vdots & \vdots & \vdots \\ 1 & a_n & A_{an}(x) \end{pmatrix} \quad (8)$$

While verifier node R_i has the private key

$$(\lambda_0^1 P_0^1(x_i) + \lambda_0^2 P_0^2(x_i) + ... + \lambda_0^n P_0^n(x_i)),.....$$
$$(\lambda_M^1 P_M^1(x_i) + \lambda_M^2 P_M^2(x_i) + ... + \lambda_M^n P_M^n(x_i))$$

So, to each edge e_k of the verifier node, k=1, 2... n separately. Compute the following polynomials as below:

$$A_0 = [\lambda_0^1 P_0^1(x_i) + ... + \lambda_0^n P_0^n(x_i)] \sum_{j=1}^{n} g_j(e_k) \quad (9)$$
$$\vdots$$
$$A_M = [\lambda_M^1 P_M^1(x_i) + ... + \lambda_M^n P_M^n(x_i)] * \sum_{j=1}^{n} g_j(e_k) a_j^{q^{(M-1)}} \quad (10)$$

Meanwhile, it has following equation:

$$\sum_{j=1}^{n} g_j(e_k) A_{aj}(x) = \sum_{j=1}^{n} g_j(e_k) \{[\lambda_0^1 P_0^1(x) + \lambda_0^2 P_0^2(x) + ... + \lambda_0^n P_0^n(x)] + a_j[\lambda_1^1 P_1^1(x) + \lambda_1^2 P_1^2(x) + ... + \lambda_1^n P_1^n(x)] +..... + a_j^{q^{(M-1)}}[\lambda_M^1 P_M^1(x) + \lambda_M^2 P_M^2(x) + ... + \lambda_M^n P_M^n(x)]\} \quad (11)$$

From the above equation, we can learn that the authentication code of after-encoded message is the formation of the combination with source authentication codes, thus it does not need extra cost to compute the authentication codes of the encoded messages.

5) Authentication process

When an intermediate verifier node receives the message, then it makes the authentication for the message. So to the verifier node R_i, bases on its private key and x_i, it can compute A_0, A_1... A_M while it also can get the equation

$$B = \sum_{j=1}^{n} g_j(e_k) A_{aj}(x_i) \quad (12)$$

If $A_0+A_1+\ldots.+A_M=B$, then receives the message, otherwise, it discards the message.

6) Decodding message received

Verifier node receives the message that passes the authentication, because the after-encoded message is the formation of the combination with source authentication codes, the encoded message can be decoded by Gauss Eliminate.

B. The analysis of authentication scheme efficiency

For the authentication scheme, the analysis is made as follows: communication cost, computational cost and storage cost.

- Communication cost: The cost mainly relies on the size of the authentication tag $|A_{ai}|$, as the length of tag is *nkl*, so the computation complexity is $O(nkl)$.
- Computational cost: The cost involves computing and appending the authentication code at the source, and verifying the authentication code at some intermediate nodes and at the destinations. On the one hand, cost at the source: For a message a_i, in order to generate authentication code, source node need to compute the following polynomial:

$$A_{ai}(x) = [\lambda_0^1 P_0^1(x) + \lambda_0^2 P_0^2(x) + ... + \lambda_0^n P_0^n(x)] + a_i[\lambda_1^1 P_1^1(x) + \lambda_1^2 P_1^2(x) + ... + \lambda_1^n P_1^n(x)] + a_i^q[\lambda_2^1 P_2^1(x) + \lambda_2^2 P_2^2(x) + ... + \lambda_2^n P_2^n(x)] +..... + a_i^{q^{(M-1)}}[\lambda_M^1 P_M^1(x) + \lambda_M^2 P_M^2(x) + ... + \lambda_M^n P_M^n(x)] \quad (13)$$

Which involves $n(M\text{-}1)l$ exponentiations, besides, it includes $nkMl$ multiplications; On the other hand, Cost at the verifying nodes: For a verifying node R_i, it has to do two things to check the authentication code. First, it has to compute the following expressions:

$$A_0 = [\lambda_0^1 P_0^1(x_i) + \lambda_0^2 P_0^2(x_i) + ... + \lambda_0^n P_0^n(x_i)] \sum_{j=1}^{n} g_j(e_k) \quad (14)$$

$$A_1 = [\lambda_1^1 P_1^1(x_i) + \lambda_1^2 P_1^2(x_i) + ... + \lambda_1^n P_1^n(x_i)] \sum_{j=1}^{n} g_j(e_k) a_j \quad (15)$$

$$A_2 = [\lambda_2^1 P_2^1(x_i) + \lambda_2^2 P_2^2(x_i) + ... + \lambda_2^n P_2^n(x_i)] \left(\sum_{j=1}^{n} g_j(e_k) a_j \right)^q \quad (16)$$

$$\vdots$$

$$A_M = [\lambda_M^1 P_M^1(x_i) + \lambda_M^2 P_M^2(x_i) + ... + \lambda_M^n P_M^n(x_i)] \left(\sum_{j=1}^{n} g_j(e_k) a_j \right)^{q^{(M-1)}}$$

(17)

Which contains $n(M\text{-}1)l$ exponentiations, and also include $n(M+1)l$ multiplications; Second, it has to compute *B*. Since the polynomial is of degree *k*-1, so to $x_i^j, j = 2,.....,k-1$, it involves $n(k\text{-}2)l$ exponentiations, to $x_i^j, j = 1,.....,k-1$, it contains $n(k\text{-}1)l$ multiplications

- Storage cost: The cost of storing private key at the source is $O(n(M+1)lk)$; while the cost of storing private key at the verifying nodes is $O(n(M+1)l)$.

C. The analysis of authentication scheme security

For this scheme, our object mainly prevents malicious node to make a substitution attack, that is, to send a fake message such that a node which checks the authentication code, we consider two situations. One is for a single malicious intermediate encoded node, and the other is a group of malicious intermediate encoded nodes.

1) Against one malicious node:Suppose that a malicious node V_i has h incoming edges, its received vectoris thus has the following equation:

$$\begin{pmatrix} y(e_{i1}) \\ \vdots \\ y(e_{ih}) \end{pmatrix} = \begin{pmatrix} g_1(e_{i1}) & \cdots & g_h(e_{i1}) \\ \vdots & \vdots & \vdots \\ g_1(e_{ih}) & \cdots & g_h(e_{ih}) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_h \end{pmatrix}$$

$$= \begin{pmatrix} g_1(e_{i1}) & \cdots & g_h(e_{i1}) \\ \vdots & \vdots & \vdots \\ g_1(e_{ih}) & \cdots & g_h(e_{ih}) \end{pmatrix} \begin{pmatrix} 1 & a_1 & A_{a1}(x) \\ \vdots & \vdots & \vdots \\ 1 & a_h & A_{ah}(x) \end{pmatrix} \quad (18)$$

If we write

$$A_{aj}(x) = b_{j0} + b_{j1}x + ... + b_{j,k-1}x^{k-1} \quad (19)$$

So we have that for all incoming edges e_m, $m=i_1 \ldots i_h$

$$\sum_{j=1}^{h} g_j(e_m) A_{aj}(x) = \sum_{j=1}^{h} g_j(e_m)(b_{j0} + b_{j1}x + ... + b_{j,k-1}x^{k-1})$$

$$= c_{m0} + c_{m1}x + ... + c_{m,k-1}x^{k-1} \quad (20)$$

So malicious node knows c_{mi}, where i =1, 2... k-1. For each incoming edge of the malicious node, it can obtain the following system of linear equation: $AG=C$, where A is a matrix with $k \times (M+1)$, contains the coefficients of the private key, C is a matrix with $k \times h$, which known to the malicious node. For the authentication scheme to be secure, we at least need $(M+1)>h$.

For this situation, we will give an example.

Suppose in a multi-source network, it has two source nodes, the verifier node R_1 receives two messages at a time. Thus a node R_1 has received the following vector:

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

With previous description, we can get

Consider the node is malicious, instead of checking the authentication code it actually wants to make a substitution attack. Since we have that:

$$A_{a1}(x) = [\lambda_0^1 P_0^1(x) + \lambda_0^2 P_0^2(x)] + a_1[\lambda_1^1 P_1^1(x) + \lambda_1^2 P_1^2(x)] + a_1^2[\lambda_2^1 P_2^1(x) + \lambda_2^2 P_2^2(x)]$$

$$= [\lambda_0^1(a_{00}^1 + a_{01}^1 x) + \lambda_0^2(a_{00}^2 + a_{01}^2 x)]$$

$$+ a_1[\lambda_1^1(a_{10}^1 + a_{11}^1 x) + \lambda_1^2(a_{10}^2 + a_{11}^2 x)]$$

$$+ a_1^2[\lambda_2^1(a_{20}^1 + a_{21}^1 x) + \lambda_2^2(a_{20}^2 + a_{21}^2 x)]$$

$$= [(\lambda_0^1 a_{00}^1 + \lambda_0^2 a_{00}^2) + (\lambda_1^1 a_{10}^1 + \lambda_1^2 a_{10}^2)a_1 + (\lambda_2^1 a_{20}^1 + \lambda_2^2 a_{20}^2)a_1^2]$$

$$+ x[(\lambda_0^1 a_{01}^1 + \lambda_0^2 a_{01}^2) + (\lambda_1^1 a_{11}^1 + \lambda_1^2 a_{11}^2)a_1 + (\lambda_2^1 a_{21}^1 + \lambda_2^2 a_{21}^2)a_1^2]$$

$$=: b_{10} + x b_{11}$$

$$A_{a2}(x) = [\lambda_0^1 P_0^1(x) + \lambda_0^2 P_0^2(x)] + a_2[\lambda_1^1 P_1^1(x) + \lambda_1^2 P_1^2(x)] + a_2^2[\lambda_2^1 P_2^1(x) + \lambda_2^2 P_2^2(x)]$$

$$= [\lambda_0^1(a_{00}^1 + a_{01}^1 x) + \lambda_0^2(a_{00}^2 + a_{01}^2 x)]$$

$$+ a_2[\lambda_1^1(a_{10}^1 + a_{11}^1 x) + \lambda_1^2(a_{10}^2 + a_{11}^2 x)]$$

$$+ a_2^2[\lambda_2^1(a_{20}^1 + a_{21}^1 x) + \lambda_2^2(a_{20}^2 + a_{21}^2 x)]$$

$$= [(\lambda_0^1 a_{00}^1 + \lambda_0^2 a_{00}^2) + (\lambda_1^1 a_{10}^1 + \lambda_1^2 a_{10}^2)a_2 + (\lambda_2^1 a_{20}^1 + \lambda_2^2 a_{20}^2)a_2^2]$$

$$+ x[(\lambda_0^1 a_{01}^1 + \lambda_0^2 a_{01}^2) + (\lambda_1^1 a_{11}^1 + \lambda_1^2 a_{11}^2)a_2 + (\lambda_2^1 a_{21}^1 + \lambda_2^2 a_{21}^2)a_2^2]$$

$$=: b_{20} + x b_{21}$$

We can rewrite:

$$g_1(e_1)A_{a1}(x) + g_2(e_1)A_{a2}(x)$$

$$= g_1(e_1)(b_{10} + x b_{11}) + g_2(e_1)(b_{20} + x b_{21})$$

$$= g_1(e_1)b_{10} + g_2(e_1)b_{20} + x(g_1(e_1)b_{11} + g_2(e_1)b_{21})$$

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:

$$g_1(e_1)A_{a1}(x) + g_2(e_1)A_{a2}(x)$$

$$= g_1(e_1)$$

$$[(\lambda_0^1 a_{00}^1 + \lambda_0^2 a_{00}^2) + (\lambda_1^1 a_{10}^1 + \lambda_1^2 a_{10}^2)a_1 + (\lambda_2^1 a_{20}^1 + \lambda_2^2 a_{20}^2)a_1^2]$$

$$+ g_1(e_1) \times$$

$$x[(\lambda_0^1 a_{01}^1+\lambda_0^2 a_{01}^2)+(\lambda_1^1 a_{11}^1+\lambda_1^2 a_{11}^2)a_1+(\lambda_2^1 a_{21}^1+\lambda_2^2 a_{21}^2){a_1}^2]$$

$$+ g_2(e_1)\times$$

$$[(\lambda_0^1 a_{00}^1+\lambda_0^2 a_{00}^2)+(\lambda_1^1 a_{10}^1+\lambda_1^2 a_{10}^2)a_2+(\lambda_2^1 a_{20}^1+\lambda_2^2 a_{20}^2){a_2}^2]$$

$$+ g_2(e_1)\times$$

$$x[(\lambda_0^1 a_{01}^1+\lambda_0^2 a_{01}^2)+(\lambda_1^1 a_{11}^1+\lambda_1^2 a_{11}^2)a_2+(\lambda_2^1 a_{21}^1+\lambda_2^2 a_{21}^2){a_2}^2]$$

$$=(\lambda_0^1 a_{00}^1+\lambda_0^2 a_{00}^2)(g_1(e_1)+g_2(e_1))$$

$$+(\lambda_1^1 a_{10}^1+\lambda_1^2 a_{10}^2)(g_1(e_1)a_{1`}+g_2(e_1)a_2)$$

$$+(\lambda_2^1 a_{20}^1+\lambda_2^2 a_{20}^2)(g_1(e_1){a_1}^2+g_2(e_1){a_2}^2)$$

$$+x\,[(\lambda_0^1 a_{01}^1+\lambda_0^2 a_{01}^2)(g_1(e_1)+g_2(e_1))$$

$$+(\lambda_1^1 a_{11}^1+\lambda_1^2 a_{11}^2)(g_1(e_1)a_1+g_2(e_1)a_2)$$

$$+(\lambda_2^1 a_{21}^1+\lambda_2^2 a_{21}^2)(g_1(e_1){a_1}^2+g_2(e_1){a_2}^2)]$$

Since the malicious node knows:

$$g_1(e_1)+g_2(e_1)$$
$$g_1(e_1)a_1+g_2(e_1)a_2$$
$$g_1(e_1){a_1}^2+g_2(e_1){a_2}^2$$

Because the malicious has two incoming edges, it can get the following equation:

$$\begin{pmatrix}\lambda_0^1 a_{00}^1+\lambda_0^2 a_{00}^2 & \lambda_1^1 a_{10}^1+\lambda_1^2 a_{10}^2 & \lambda_2^1 a_{20}^1+\lambda_2^2 a_{20}^2\\ \lambda_0^1 a_{01}^1+\lambda_0^2 a_{01}^2 & \lambda_1^1 a_{11}^1+\lambda_1^2 a_{11}^2 & \lambda_2^1 a_{21}^1+\lambda_2^2 a_{21}^2\end{pmatrix}G=\begin{pmatrix}c_{10} & c_{20}\\ c_{11} & c_{21}\end{pmatrix} \quad (21)$$

Where

$$G=\begin{pmatrix} g_1(e_1)+g_2(e_1) & g_1(e_2)+g_2(e_2)\\ g_1(e_1)a_1+g_2(e_1)a_2 & g_1(e_2)a_1+g_2(e_2)a_2\\ g_1(e_1){a_1}^2+g_2(e_1){a_2}^2 & g_1(e_2){a_1}^2+g_2(e_2){a_2}^2\end{pmatrix} \quad (22)$$

We can learn that G is a 3×2 matrix , thus it satisfy the security condition . otherwise, if only need two polynomials to create the authentication code, then the matrix G would be a 2×2 matrix, and thus could be very likely invertible, so the malicious node can recover the secret coefficients of the source private key, in other word, the authentication is not secure.

2) Against a group of malicious nodes: Suppose a network with n source nodes, there are K nodes $v_1,\ldots,v_K$ collaborate to make a substitution attack. For each node can get a vector of the data from the network, so we can obtain the following equation:

$$AG_i=C_i \quad i=1, 2\ldots K \quad (23)$$

We can rewrite:

$$A\,[G_1,G_2\ldots G_K]=[C_1,C_2\ldots C_K] \quad (24)$$

Likewise, the authentication secure condition is that

$$M+1\geqslant h_1+h_2+\ldots+h_K \quad (25)$$

We consider a situation where some of the nodes who are given the private keys to check the authentication could be corrupted, for we consider that K nodes $v_1,\ldots,v_K$ collaborate, thus we assume the worst case, namely that all of them actually have the private key:

$$(\ (\lambda_0^1 P_0^1(x_i)+\lambda_0^2 P_0^2(x_i)+.....+\lambda_0^n P_0^n(x_i)),.....$$
$$(\lambda_M^1 P_M^1(x_i)+\lambda_M^2 P_M^2(x_i)+.....+\lambda_M^n P_M^n(x_i))\ ).$$

Where i=1, 2,…, K.

Since the values $x_1,\ldots,x_T$, the group of adversaries can get the following equation with their knowledge of the private key, namely $XA=P$, where

$$X=\begin{pmatrix}1 & x_1 & \cdots & {x_1}^{k-1}\\ 1 & x_2 & \cdots & {x_2}^{k-1}\\ \vdots & \vdots & \vdots & \vdots\\ 1 & x_K & \cdots & {x_K}^{k-1}\end{pmatrix} \quad (26)$$

where the $K\times k$ matrix X contains the public key values,

where the $k\times(M+1)$ matrix A contains the coefficients of the private key,

where the $k\times(M+1)$ matrix P contains the private key of the malicious nodes. Since the degree of the polynomials is k-1, namely K can be at most k-1, otherwise from the knowledge of only the private and public keys, the group of malicious nodes can recover the source nodes private key. The following description will prove that suppose the adversaries know the private key and the one gathered from all the received vectors, the adversaries still can not do better than guess the source nodes private key.

The following description will prove that the adversaries know the private key and the one gathered from all the received vectors, the adversaries still can not do better than guess the source nodes private key.

Lemma: There exist q matrices A with coefficients, such that: $AG'=C'$, $XA=P$

Where A is the matrix of $k\times(M+1)$,

X is the matrix of $(k\text{-}1)\times k$,

G' is the matrix of $(M+1)\times H$,

C' is the matrix of $k\times H$,

P is the matrix of $(k\text{-}1)\times(M+1)$,

and $H=h_1+h_2+\ldots.+h_K$.

Proof: Since the matrix G' is of the form

$$\begin{pmatrix}\sum_{j=1}^{n} g_j(e_{i1}) & \cdots & \sum_{j=1}^{n} g_j(e_{iH})\\ \sum_{j=1}^{n} g_j(e_{i1})a_j & \cdots & \sum_{j=1}^{n} g_j(e_{iH})a_j\\ \vdots & \vdots & \vdots\\ \sum_{j=1}^{n} g_j(e_{i1}){a_j}^{q(M-1)} & \cdots & \sum_{j=1}^{n} g_j(e_{iH}){a_j}^{q(M-1)}\end{pmatrix}$$

For any invertible matrix D, we have that

$$AG^{'}=C^{'} \rightarrow AG^{'}D=C^{'}D \quad (27)$$

We can rewrite that: there exists an invertible matrix D, such that $G^{'}D$ is of the Vandermonde like form:

$$\begin{pmatrix} 1 & \cdots & 1 \\ \gamma_1 & \cdots & \gamma_H \\ \gamma_1^{\,q} & \cdots & \gamma_H^{\,q} \\ \vdots & \vdots & \vdots \\ \gamma_1^{\,q^{(M-1)}} & \cdots & \gamma_H^{\,q^{(M-1)}} \end{pmatrix}$$

Specially, if all the coefficients of the first row of $G^{'}$ are not zero, we can rewrite as:

$$D = diag((\sum_{j=1}^{n} g_j(e_{i1}))^{-1},.....,(\sum_{j=1}^{n} g_j(e_{iH}))^{-1}) \quad (28)$$

So the problem change into another formation, namely $AG^{'}=C^{'}$, $XA=P$ with $G^{'}$ satisfy the Vander monde form.

Firstly, we solve a homogeneous system of equation:

$$AG^{'}=0,\ XA=0. \quad (29)$$

We define $f(x, y) = (1, x \ldots x^{k-1})\ A\ (1\ y \ldots y^{q(M-1)})$ (30)

Since A contains $f(x, y)$ in two indeterminate x and y, then it exists a polynomial $f(x,y)$ whose roots are $x_1,.....,x_{k-1}$ and $\gamma_1,.....,\gamma_H$, thus we can get $XA=0$ in $x=x_1 \ldots x_k$, likewise, to $AG^{'}=0$. We can get corresponding y, since these coefficients of the matrices in F_q also satisfy the equation, this gives q suitable matrices. So based on homomorphism, the lemma can be proved.

Proposition: The above scheme is suitable for multi-source network coding situation, and it uses an unconditionally secure authentication code to prevent pollution attacks. The scheme is robust against a coalition of up to k-1 adversaries in which every key can be used to authenticate up to M messages.

Proof: To make a substitution attack, the k-1 malicious nodes want to generate a message such that it is accepted as authentic by the receiver R_i that they are trying to cheat, but this message is bogus, then these malicious nodes need to guess their private key:

$$((\lambda_0^1 P_0^1(x_i) + \lambda_0^2 P_0^2(x_i) + ... + \lambda_0^n P_0^n(x_i)),.....$$
$$(\lambda_M^1 P_M^1(x_i) + \lambda_M^2 P_M^2(x_i) + ... + \lambda_M^n P_M^n(x_i)))$$

And choose a polynomial:

$$\begin{aligned} A_a(x_i)^{'} = & [\lambda_0^1 P_0^1(x_i) + \lambda_0^2 P_0^2(x_i) + ... + \lambda_0^n P_0^n(x_i)] \\ & + a^q [\lambda_1^1 P_1^1(x_i) + \lambda_1^2 P_1^2(x_i) + ... + \lambda_1^n P_1^n(x_i)] \\ & + ... \\ & + a^{q^{(M-1)}} [\lambda_M^1 P_M^1(x_i) + \lambda_M^2 P_M^2(x_i) + ... + \lambda_M^n P_M^n(x_i)] \end{aligned} \quad (31)$$

These malicious nodes can get the following equation by seeing the message transmittance:

$$A_{k\times(M+1)} G_{(M+1)\times H}^{'} = C_{k\times H}^{'} \quad (32)$$

$$X_{K\times k} A_{k\times(M+1)} = P_{K\times(M+1)} \quad (33)$$

If there is no matrix $A_{k\times(M+1)}$ satisfy the equation, these malicious nodes collect the information will be useless. While if the matrix exists, then there are q matrices, namely, there are q different $(M+1)$ tuple of polynomial $(P_0^i(x)^{'},.....,P_M^i(x)^{'})$, likely to be the source nodes private key. Thus the probability of the guess is $1/q$.

V. Conclusion

In this paper, we proposed an unconditionally secure authentication code scheme that is suitable for multi-source network coding; our scheme is robust against an attacker even it has unlimited computational resources, for our scheme is based on theoretic strength. Besides, the authentication scheme is robust against pollution attacks either from outsides or coalition of k-1 malicious insiders. In multi-source network coding, intermediate nodes can verify the integrity and origin of the messages received without having to decode, and detect and discard the messages that fail the verification. By this way, the pollution is canceled out before reaching the destinations. Besides, in our paper, we compare several schemes with our method, the specific result is listed as the following table. From the table, we know that our scheme supports multi-source network coding, and also is immune from the savage attack. Since the research of multi-source network coding was studied fewer, our scheme about the security analysis is not perfect; it needs still to be improved on.

Table 1. The comparison of several schemes

Scheme	***Multi-source support***	***Savage attack limitation***
Yu's[11]	No	$2^{a/2}$
Zhao's[5]	No	$2^{b/2}$
Charles's[4]	No	$2^{c/2}$
Frederique[9]	No	None
Our scheme	Yes	None

Where: $a=d$ stands for the RSA private key in [11]

$b=\alpha n$ Stands for the private key in [5]

$c=sn$ Stands for the private key in [4]

The results of savage attack limitation are got by the Birthday Paradox [12]

Acknowledgment

This research was supported by the National Natural Science Foundation of China (under Grant No.60672137).

References

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. "Network information flow". IEEE Transactions on Information Theory, July 2000, in press.

[2] S.Li, R.Yeung, and N.Cai, “Linear Network Coding”, in IEEE Transaction on Information Theory,Vol.49, No.2, pp.37138,2003, in press.

[3] C.Gkantsidis and P.Rodriguez , “Cooperative Security for Network Coding File Distribution ” ,IEEE INFOCOM ,2006.

[4] D.Charles , K.Jain ,and K.Lauter , “Signatures for Network Coding” , Conference on Information Sciences and Systems ,2006, in press.

[5] F.Zhao ,T.Kalker ,M.Medard ,and K.J.Han , “Signatures for Content Distribution with Network Coding” , IEEE International Symposium on Information Theory ,2007,in press.

[6] Y.Desmedt ,Y.Frankel ,and M.Yung , “Multi-Receiver/Multi-Sender NetworkSecurity: Efficient Authenticated Multicast/Feedback” ,IEEE INFOCOM ,1992, in press.

[7] R.Safavi-Naini ,and H.Wang , “New results on multi-receiver authentication codes” , Eurocrypt’98, LNCS 1403, pp.527-541,1998, in press.

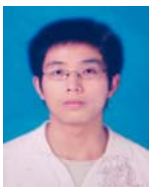
[8] R.Safavi-Naini , H.Wang , “Multire-ceiver Authentication Codes: Models ,Bounds, Constructions and Extensions” ,Volume 151, Issues 1-2, 25 May 1999 ,Pages 148-172, in press.

[9] Frédérique Oggier and Hanane Fathi , “An Authentication Code against Pollution Attacks in Network Coding”, Information Theory ; Cryptography and Security , arXiv:0909.3146v1,September 17 ,2009, in press.

[10] Wenjie Yan ,Mingxi Yang ,Layuan Li ,Huajing Fang , “Short Signatures for Multi-source Network Coding” ,2009 International Conference on Multimedia Information Networking and Security, in press.

[11] Z.Yu, Y.Wei, B.Ramkumar, and Y.Guan. “An Efficient Signature-based Scheme for SecuringNetwork Coding against Pollution Attacks. In Proc.27th Annual IEEE Conf. on Computer Commun., INFOCOM,2008, in press.

[12] William Stallings,Cryptography and Network Security Principles and Practices,P346-350.



Hong Yang, borned in Yingcheng City of Hubei Province on May 8th, 1987 and received his B.S.degree in Computer Science and Technology from Wuhan University of Technology in 2008. He is now working toward his M.S.degree in school of Computer Science and Technology at Wuhan University of Technology,China.His research interest is network security.

Mingxi Yang received her Ph.D. degree in Computer Applied Technology from Wuhan University of Technology, China in 2007, and B.S. degree from Huazhong University of Science and Technology, China in 1982. She works as an associate professor and Director of the Institute of Software in School of Computer Science and Technology at Wuhan University of Technology, in China now. Her research interests are in computer network and network security.

