Computer Communications 35 (2012) 344–351

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Short signature scheme for multi-source network coding

Wenjie Yan [a,c], Mingxi Yang [a,*], Layuan Li [a], Huajing Fang [b]

[a] School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China
[b] Dept. of Control Science and Engineering, Huazhong University of Science and Technology, China
[c] Dept. of Planning and Information, SINOPEC Chemical Sales Central China Company, China

ARTICLE INFO

Article history:
Received 20 January 2010
Received in revised form 13 September 2011
Accepted 13 October 2011
Available online 23 October 2011

Keywords:
Multi-source network coding
Signature
Homomorphic hash function
Pollution attacks

ABSTRACT

It has been proven that network coding can provide significant benefits to networks. However, network coding is very vulnerable to pollution attacks. In recent years, many schemes have been designed to defend against these attacks, but as far as we know almost all of them are inapplicable for multi-source network coding system. This paper proposed a novel homomorphic signature scheme based on bilinear pairings to stand against pollution attacks for multi-source network coding, which has a broader application background than single-source network coding. Our signatures are publicly verifiable and the public keys are independent of the files so that our scheme can be used to authenticate multiple files without having to update public keys. The signature length of our proposed scheme is as short as the shortest signatures of a single-source network coding. The verification speed of our scheme is faster than those signature schemes based on elliptic curves in the single-source network.

1. Introduction

Network coding was first proposed by Ahlswede et al. [1] in order to maximize the throughput of multicast networks. In contrast to traditional "store and forward" routing, network coding allows intermediate nodes to process and modify the data packets in transit. Later, Li et al. [2] further proved that linear network coding is sufficient to achieve this purpose. Based on it, Ho et al. [3,4] proposed a random linear network coding, and as a result, which no longer needs for decoders to know the topology of the network. Network coding has been shown to offer a number of advantages, such as lesser network congestion, higher reliability, and lower power consumption.

However, network coding poses new security challenges. One main challenge is pollution attacks, in which the adversary nodes intentionally modify or forge the transmitted packets and inject them into the coding packets. What is more, the polluted packets will quickly spread into the networks and infect a large number of packets, as they are transmitted by the downstream nodes.

1.1. Related work

Recently, several schemes have been proposed to provide protection against pollution attacks for network coding applications. These schemes can be classified in two categories: information theoretic approaches and cryptographic approaches.

Information theoretic approaches: A major method of information theoretic approaches is introducing redundant information into original packets for enabling recovery from malicious faults, such as in [5–7]. These approaches have the advantage of not relying on any computational assumptions so the process speed is faster, but they are secure only against relatively limited kinds of adversaries, and the communication overhead of these approaches is heavier.

Cryptographic approaches: Cryptographic approaches rely on the computational assumptions. By this kind of approaches the source node uses cryptographic techniques to generate authentication information and append it to corresponding packets and allow the intermediate node to encode, sign or verify the packets. In the following, we mainly introduce the cryptographic approaches.

Homomorpic hashing function is first proposed by Krohn et al. [8]. In their scheme the source node computes the hash values $h_1, h_2, \ldots, h_m$ for the packets $X_1, X_2, \ldots, X_m$ and distributes these hash values to all the nodes in the networks. When a packet w combined from $X_1, X_2, \ldots, X_m$ is received, a receiver can compare the hash value h_w of w, which is worked out from the hash values $h_1, h_2, \ldots, h_m$ via additive homomorphic computation, with HASH (w), then the encoded packet w is verified. In this scheme the source node needs to renew a batch of hash values for every new massage.

Zhao et al. [9] uses a signature vector X to authenticate the vector space $V = \text{span}\{v_1, \ldots, v_m\}$, but the size of this signature vector is as long as that of each packet. And the scheme requires updating the public keys for authenticating new files.

* Corresponding author.
E-mail address: yangmx@whut.edu.cn (M. Yang).

0140-3664/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2011.10.012

Dong et al. [10] proposed a scheme which uses time-based checksum to allow the intermediate nodes to authenticate the received packets. However, this scheme requires time synchronization between senders and receivers, which is unpractical. In addition the size of each checksum is twice as long as the size of the packet and the received packet cannot be verified at once until the right checksum arrived. Besides, the checksums are flooded in the networks. This would greatly consume the bandwidth.

All of these schemes in [8–10] require the sender to know the entire file in advance, before the authentication information can be computed, so these schemes do not support the transmission of streaming packets, where the sender transmits packets as they are generated rather than buffering them and transmitting them all at once.

Yu et al. [11] take advantage of a homomorphic signature function RSA to sign the hash value of packets and append these signatures to corresponding packets so the forwarders can compose the signatures for their encoded packets without knowing the source private key and its downstream nodes can verify the encoded packets with the source public key. One of the drawbacks of this scheme is the signature size is a bit too long and it has to refresh the public keys while to sign a new file in case of being replay attacked. And furthermore, Aaram Yun et al. [21] point out that this scheme is in fact not homomorphic.

Charles et al. [12] proposed a homomorphic signature scheme which is built on Weil pairing operations [13,14] over elliptic curves, but this scheme also need to refresh the public key when used to sign a new file.

Katz and Waters [15] proposed also a signature scheme based on bilinear pairing to defend against pollution attacks, in which a file number *id* was introduced to thwart the replay attacks.

Jiang et al. [16] proposed a homomorphic scheme on the elliptic curves. Especially in their scheme, a file identifier k dynamically updated by a one way hash chain was introduced against the replay attacks. That paper described an efficient method and allowed the forwarders to verify multiple received packets synchronously.

Here we want to point out especially that the above published authentication schemes could only be applicable to single source network coding system, and not for multi-source network coding system which has a broader application background in networks.

In the signature scheme for multi-source network coding system, there are multiple source nodes. Each source has to generate signature with distinct private key so that other source nodes cannot fake it. However in the previous signature schemes for network coding [11,12,15,16], these distinct private keys will destroy the homomorphism of the signature algorithms, which means that the intermediate nodes cannot generate a valid homormophic signature for an encoded packet without knowing the source private keys.

The authentication schemes in [8–10] cannot be applied in multi-source network coding too, the authentication information generated by a source node can only be used to verify those packets from this single node, while combination of different source packets could not be verified correctly by forwarders.

Recently, Agrawal et al. [19] and [20] proposed their schemes to defend against pollution attacks in multi-source network coding. Agrawal et al. [19] introduced a *merge* algorithm into their work to generate hthe public keys and signatures at intermediate nodes. Laszlo's work [20] is built on bilinear pairing, and the way they sign the packets is similar to Jonathan's [15] method. Both [19,20], however, have a common drawback that the size of signature grows linearly with the number of the sources. If a packet is mixed by l original packets, then the length of the signature on this packet is l times the signature length in single source network coding. That is unpractical.

1.2. Our contribution

In this paper, we proposed a homomorphic signature scheme based on bilinear pairings to provide protection against pollution attacks for multi-source linear network coding models even when the adversaries can corrupt an arbitrary number of nodes, eavesdrop on all links in networks. Every source node in our scheme has its own distinct private key pair and public key and that our signature scheme will remain its homomorphism. Thus the intermediate nodes in our scheme can authenticate the received encoded packets signed by different source nodes with the corresponding public keys, and can also sign the encoded packets without knowing the various source private keys. In addition, the public keys are independent of the transmitted file, which means our signature scheme supports transmitting multiple files without having to update the public keys.

The signature in our scheme has constant size, which is as short as the shortest signatures of a single-source network (about 160 bits in practical networks). All of these signatures are elements in a single group. Our scheme also supports the transmission of streaming data packets, i.e. each source node need not to know all the packets in advance.

From a computational point of view, we define that there are m source packets, and each packet is a $(m+n)$-dimension vector. With signatures size equally short, the verification in our scheme requires only once of pairing computation and $m+n$ times of point multiplications. In contrast to similar schemes based on bilinear pairings, e.g. [12], which require $m+n+1$ times of pairing computations, and [15,16], which need twice of pairing computations and $m+n$ times of point multiplications, and so on, our signature scheme is more efficient.

1.3. Outline of this paper

The remainder of the paper is organized as follows: Section 2 introduces the system model and threat model; Section 3 introduces our proposed signature scheme; Section 4 proves its security; Section 5 analyzes our scheme's performance; and Section 6 is the conclusion.

2. Background

2.1. System model

Multi-source network coding is a very rich model which encompasses many communication situations, but our model is only relevant to multiple sources in the networks and the random linear network coding approach, so we give the model as follows.

We model the network similar to [12,17] by a directed graph $G_d = (E, V)$, where E is a set of links and V is a set of vertices in the network. We assume that there are m source nodes $S = (s_1, s_2, \ldots, s_m) \subset V$ in the network, each source wants to send a file to a set of destination nodes $T \subset V$ and each destination node wants to receive all m source files, we suppose that each source file is a vector of dimension n, thus the source file from source s_i can be considered as: $\overline{X}_i = (\bar{x}_{i1}, \ldots, \bar{x}_{in})$, before outputting this file, source s_i augments its $\overline{X}_i$ by appending an original coding vector to create X_i as follows:

$$X_i = \left(\bar{x}_{i_1}, \ldots, \bar{x}_{in}, \overbrace{\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0}^{m}\right) = (x_{i1}, \ldots, x_{i,m+n}) \in F_q^{m+n},$$

where F is a finite field, prime q is a pre-determined security parameter and vector $\overbrace{\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0}^{m}$ can be considered as the identifier