

Homomorphic Subspace MAC Scheme for Secure Network Coding

Guangjun Liu and Xiao Wang

Existing symmetric cryptography-based solutions against pollution attacks for network coding systems suffer various drawbacks, such as highly complicated key distribution and vulnerable security against collusion. This letter presents a novel homomorphic subspace message authentication code (MAC) scheme that can thwart pollution attacks in an efficient way. The basic idea is to exploit the combination of the symmetric cryptography and linear subspace properties of network coding. The proposed scheme can tolerate the compromise of up to $r-1$ intermediate nodes when r source keys are used. Compared to previous MAC solutions, less secret keys are needed for the source and only one secret key is distributed to each intermediate node.

Keywords: Network coding, pollution attacks, homomorphic MAC, network security, homomorphic cryptography.

I. Introduction

As an important breakthrough technology of information transmission, network coding has been shown to improve the capacity and robustness in trusted networks [1], [2]. However, network coded systems are highly susceptible to pollution attacks, due to the mixing nature of network coding, in which an inside or outside adversary can inject fake or modified packets into the information flow. This problem can be particularly serious because even a small number of polluted packets may propagate and contaminate all data being transmitted, which results in epidemic degradation of the transmission performance.

So far, great research efforts have been devoted to securing network coding against pollution attacks. The existing secure

solutions are widely categorized into two kinds: information-theoretic and cryptographic. Existing information-theoretic schemes, not relying on any computational assumptions, only offer relatively limited pollution detection/correction at receivers in an end-to-end fashion [3]. Hence, it cannot help the forwarders to prevent the transmission of useless polluted packets. In contrast, cryptography-based schemes enable the forwarders to verify the veracity of their received packets en route and hence reject ill-formed packets before polluting the downstream flows. Two typical representatives of cryptographic schemes that have been proposed to thwart pollution attacks for network coding are public-key cryptography (PKC)-based approaches and symmetric-key cryptography-based approaches.

Several PKC cryptographic schemes for pollution attacks have been presented using homomorphic PKC cryptography [4], [5]. However, all of these schemes involve expensive arithmetic operations over a large extension field, which results in substantial computational delays and performance degradation. To mitigate the computational cost, the solutions using symmetric cryptography have received widespread attention. Yu and others [6] constructed multiple message authentication codes (MACs) based on probabilistic key predistribution for all verifiers. Agrawal and Boneh [7] proposed two homomorphic MACs, and only the receivers in their basic scheme are able to verify data packets, whereas their broadcast homomorphic MAC scheme can detect pollution attacks using the technique of cover-free family for key predistribution. Similar schemes were subsequently presented in [6], [8]. Unfortunately, these schemes suffer complicated key predistribution, which needs to be carefully designed for thwarting the coalition of vicious nodes and thus results in the network being more fragile. Additionally, most symmetric-key-

Manuscript received Apr. 23, 2012; revised Sept. 30, 2012; accepted Oct. 9, 2012.

Guangjun Liu (phone: +86 139 9122 8740, liuguangjuns@gmail.com) and Xiao Wang (wangxiaoxiao@mail.xidian.edu.cn) are with the State Key Laboratory of ISN, Xidian University, Xi'an, Shaanxi, China.

<http://dx.doi.org/10.4218/etrij.13.0212.0166>

based solutions become vulnerable to tag pollution attacks. To address this problem, Li and others [9] proposed RIPPLE, an in-network authentication scheme that leverages delayed release of secret keys, as in DART [10]. Although these schemes are considered to be of low complexity to achieve optimal network throughput, they require time asymmetry of level keys, which is not easily realized in distributed settings.

With the observations of the randomization and the linear subspace properties of random network coding, Kehdi and Li [11] proposed an elegant scheme based on null keys, which are simply vectors of the orthogonal space of the linear space spanned by the source vectors. Actually, the scheme can be viewed as a special case of the work of Cai and Yeung [12]. However, the scheme suffers considerable communication overhead since the vectors of orthogonal space must be distributed to all intermediate nodes unimpaired, which requires a great deal of communication and computation. Another drawback of this scheme is the lack of data streaming support since the entire file is needed before sending.

In this letter, we propose an efficient scheme against pollution attacks for network coding inspired by the idea of securing network coding against wiretapping in [12], in which each transmitted message can be considered as a solution of a public linear system for securing network coding against wiretapping. We inherit and extend this technique for resisting pollution attacks, that is, the elements of each source vector are regarded as the free variables of a secret linear system generated by source secret keys. The proposed scheme demonstrates several advantages in implementation compared to existing symmetric cryptography-based solutions.

II. System and Adversary Model

We consider a typical single source multicast scenario, in which each link is assumed to transport, free of errors, an n -length symbol vector over \mathbb{F}_q^n , where \mathbb{F}_q is a finite field. The source message is divided into generations, each of which is a subsequence represented as a matrix containing m vectors $\bar{\mathbf{v}}_i = (v_{i1}, v_{i2}, \dots, v_{in}) \in \mathbb{F}_q^n$ ($i = 1, 2, \dots, m$).

The source first creates m augmented vectors $\mathbf{v}_i \in \mathbb{F}_q^{m+n}$ ($i = 1, 2, \dots, m$) for each generation by prefixing $\bar{\mathbf{v}}_i$ with the i -th unit vector of dimension m . Any practical network coding system requires that m is much less than n so that the network overhead of network coding is small. Then, the source sends the random linear combinations of these augmented vectors to its neighboring nodes.

Upon receiving l vectors from inputting links, each intermediate node picks l random coefficients in \mathbb{F}_q and then forwards a vector \mathbf{w} (the linear combination of the received l

vectors) to the downstream network. The first m symbols of \mathbf{w} are considered global encoding coefficients. Note that only vectors from the same generation are encoded. A receiver can recover a generation using Gaussian elimination after receiving m linearly independent vectors belonging to the generation.

A forwarding vector \mathbf{w} , as illustrated above, is considered valid if it lies in the space spanned by the original augmented vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$. However, in realistic scenarios, there is always an adversary that aims to inject or modify valid vectors, even compromising one or more intermediate nodes (but not all) to launch pollution attacks. Our goal is to provide an efficient defense against these adversaries and prevent pollution attacks that destroy the data of honest nodes.

III. Proposed Homomorphic Subspace MAC Scheme

1. Definition of Subspace MAC Scheme

Our definition is similar to the model of [7]. A (q, n, m) homomorphic subspace MAC is defined by a tuple of probabilistic polynomial-time (PPT) algorithms: Setup, Sign, Combine, and Verify.

Setup. Input: the basic security parameter 1^k and the system parameters m and n for encoding and authentication. Output: a prime number q and a secret key for each node.

Sign. Input: an augmented vector $\mathbf{v} \in \mathbb{F}_q^N$ ($N = m + n$), a generation identity id , and a source key subspace \mathbf{K} . Output: a (vector, tags) pair $\mathbf{v}^* = (\mathbf{v}, t_1, t_2, \dots, t_r) \in \mathbb{F}_q^{N+r}$.

Combine. Input: l vectors $\mathbf{v}_1^*, \mathbf{v}_2^*, \dots, \mathbf{v}_l^*$ and l random coefficients $\alpha_1, \alpha_2, \dots, \alpha_l \in \mathbb{F}_q^*$. Output: $\mathbf{v}^* = \sum_{i=1}^l \alpha_i \mathbf{v}_i^*$.

Verify. Input: a vector \mathbf{v}^* , a secret key $\mathbf{K} \in \mathbf{K}$, and id . Output: either 1 (accept) or 0 (reject).

2. Construction of Subspace MAC Scheme

Based on the above model and definition, we present our construction of a (q, m, n) homomorphic subspace MAC scheme.

Setup. Given the system parameters 1^k , m , n , and q , a key distribution center (KDC) distributes the secret keys of all participants as follows:

(i) KDC generates an $r \times (N+r)$ matrix:

$$\mathbf{S} = (\mathbf{s}_1^T, \mathbf{s}_2^T, \dots, \mathbf{s}_r^T)^T = (\mathbf{s}_{ij}),$$

where $\mathbf{s}_{ij} \in \mathbb{F}_q$, $\mathbf{s}_i \in \mathbb{F}_q^{N+r}$ ($i=1, 2, \dots, r, j=1, 2, \dots, N+r$). Vector \mathbf{s}_i can be generated by a public pseudo-random generator $G: \{0, 1\}^* \rightarrow \mathbb{F}_q^{N+r}$ with a random seed sd_i . Note that $\mathbf{T}=(\mathbf{s}_{ik})$ ($k=N+1, N+2, \dots, N+r$) is an $r \times r$ full rank matrix, and $\mathbf{K}=\text{span}\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_r\}$.

(ii) KDC distributes the secret seeds sd_i ($i=1, 2, \dots, r$) to the

source.

(iii) For each nonsource node z , KDC computes and distributes a unique secret key $\kappa_z = \sum_{i=1}^r \beta_i s_i \in K$, where each β_i is selected randomly in \mathbb{F}_q .

Sign. To generate the MAC tags for an i -th augmented basis vector $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iN}) \in \mathbb{F}_q^N$, the source works as follows:

(i) The source computes its secret keys $s_i = G(sd_i)$ ($i=1, 2, \dots, r$) and then constructs a linear equation system:

$$(\mathbf{T}^{-1} \cdot \mathbf{S}) \cdot (x_1, \dots, x_N, x_{N+1}, \dots, x_{N+r})^T = 0.$$

(ii) Let $(x_1, x_2, \dots, x_N) = (v_{i1}, v_{i2}, \dots, v_{iN})$; the source computes the MAC tags of \mathbf{v}_i such that

$$(t_{i,N+1}, t_{i,N+2}, \dots, t_{i,N+r}) = (x_{N+1}, x_{N+2}, \dots, x_{N+r}).$$

Here, \mathbf{v}_i is signed as $\mathbf{v}_i^* = (\mathbf{v}_i, t_{i,N+1}, t_{i,N+2}, \dots, t_{i,N+r})$.

Combine. After receiving l forwarding vectors $\mathbf{v}_i^* = (\mathbf{v}_i, t_{i1}, t_{i2}, \dots, t_{ir})$ ($i=1, 2, \dots, l$), an intermediate node outputs $\mathbf{v}^* = \sum_{i=1}^l \alpha_i \mathbf{v}_i^* = (\sum_{i=1}^l \alpha_i \mathbf{v}_i, \sum_{i=1}^l \alpha_i t_{i1}, \dots, \sum_{i=1}^l \alpha_i t_{ir})$, in which $\alpha_1, \alpha_2, \dots, \alpha_l$ are chosen at random in \mathbb{F}_q .

Verify. Let κ_z be a secret key and \mathbf{v}^* be a forwarding vector; an intermediate node z verifies $\kappa_z \cdot (\mathbf{v}^*)^T = 0$. If it holds, output 1; otherwise, output 0.

IV. Security Analysis

We assume that each node can accurately determine the generation identity of every forwarding vector. The source can additionally compute a regular MAC (such as HMAC) on the transmitted vectors prior to encoding using network coding as the basic scheme in [7]. Then, the security of the proposed scheme is considered from two aspects.

First, we consider that an adversary \mathcal{A} tries to forge an illegal vector that can pass verification through the transmission.

Theorem 1. For any PPT adversary \mathcal{A} , the probability that \mathcal{A} conducts a successful attack is at most $1/q^{r-1}$.

Proof. Let $V_1 = \text{span}\{s_1, s_2, \dots, s_r\}$, $V_2 = \text{span}\{\kappa_{\mathcal{A}}\}$. It is not hard to find vectors $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N, \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{N+r-1}$ such that $V_1^\perp = \text{span}\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N\}$ and $V_2^\perp = \text{span}\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{N+r-1}\}$. Since s_1, s_2, \dots, s_r are mutually linearly independent, it is easy to see that

$$\text{span}\{s_1, \dots, s_r, \mathbf{b}_1, \dots, \mathbf{b}_N\} = \text{span}\{\kappa_{\mathcal{A}}, \mathbf{c}_1, \dots, \mathbf{c}_{N+r-1}\}.$$

Clearly, $\kappa_{\mathcal{A}}$ is a linear combination of s_1, s_2, \dots, s_r . Therefore, $\hat{\mathbf{v}} \in V_2^\perp$ for any vector $\hat{\mathbf{v}} \in V_1^\perp$.

To forge a valid vector, \mathcal{A} can randomly choose a vector $\mathbf{u} \in \mathbb{F}_q^{N+r}$ that is orthogonal to $\kappa_{\mathcal{A}}$ such that $\kappa_{\mathcal{A}} \cdot \mathbf{u}^T = 0$.

Since there are q^{N+r-1} different vectors in V_2^\perp , thus for any PPT adversary \mathcal{A} , the probability that it wins is $q^N/q^{N+r-1} = 1/q^{r-1}$. On the other hand, \mathcal{A} may randomly choose a vector $\mathbf{u} \in \text{span}\{\kappa_{\mathcal{A}}, \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{N+r-1}\}$, such that $\kappa_{\mathcal{A}} \cdot \mathbf{u}^T = 0$, in which case, the probability that \mathcal{A} succeeds is $q^N/q^{N+r} = 1/q^r$. \square

Theorem 1 shows that the scheme can resist both normal pollution and tag pollution attacks. However, in practical situations, some nodes can launch collusion attacks or be compromised; thus, we further consider the collusion resistance of the proposed scheme for each key update round.

Theorem 2. The proposed scheme can tolerate a coalition of up to $r-1$ compromised nodes and can ensure the security of at most $\lceil \frac{m+n}{m} \rceil - 1$ generations for each key update round.

Proof. For the proof concept of the first part, we herein assume that r nodes z_1, z_2, \dots, z_r are compromised at some point, which leads to the secret keys $\kappa_1, \kappa_2, \dots, \kappa_r$ (linearly independent with a high probability), all revealed by an adversary \mathcal{A} . Then, \mathcal{A} obtains the information of source key subspace $K = \text{span}\{\kappa_1, \kappa_2, \dots, \kappa_r\}$ and can inject any malformed vector if more than $r-1$ nodes are compromised. The second part of this theorem follows likewise. \square

V. Comparison with Existing Schemes

In addition to preventing normal and tag pollution attacks, the proposed scheme provides several significant advantages.

First, the scheme can effectively simplify the key distribution procedure to defeat collusion attacks. Using r source secret keys can be enough to tolerate a compromise of up to $r-1$ nodes. Works [6]-[9] required large numbers of secret keys to achieve comparable security. For example, to defeat a coalition of two nodes, only three secret keys are required for the source rather than at least 49, as in [7].

Second, the scheme incurs relatively low bandwidth overhead. The online bandwidth overhead per packet includes only r MAC tags, where r is a tunable system parameter. Let θ denote the size of all participants except the source. In [7], each packet had to carry no less than $\epsilon r^2 \ln \theta$ MAC tags, which does not scale when the network size θ is large. Similarly, the bandwidth overhead in [8] was far more than that of the proposed scheme, even when using a double-random key. Owing to a delayed key disclosure running method, [9], [10] exhibited similar inefficiency issues.

The main online computation overhead during the runtime of the scheme is the result of general linear combinations, the basic operations of network coding. More important is that the

MAC tag computation and verification operation are fast. For each source vector, the source only needs r tag calculations at the Sign phase, which is far less than that of the schemes in [6]-[9]. Only one vector multiplication is performed to verify a forwarding vector at each node. Thus, our scheme is significantly efficient in implementation.

VI. Conclusion

This letter proposed a novel symmetric-key-based subspace MAC scheme against pollution attacks for network coding. Different from traditional solutions, the MAC tags of each source vector are constructed by employing the linear system generated by the source keys. Also, the source key subspace is utilized to effectively simplify the key distribution procedure in the existing schemes. The proposed scheme can resist a coalition of up to $r-1$ intermediate nodes when r source keys are used, while incurring a relatively low overhead in bandwidth and computation. This threshold-tolerance property is often desired in practical applications.

References

- [1] R. Ahlswede et al., "Network Information Flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, 2000, pp. 1204-1216.
- [2] S.-Y.R. Li, R.W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, 2003, pp. 371-381.
- [3] T. Ho et al., "Byzantine Modification Detection in Multicast Networks Using Randomized Network Coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, 2008, pp. 2798-2803.
- [4] D. Boneh et al., "Signing a Linear Subspace: Signature Schemes for Network Coding," *Proc. 12th Int. Conf. Practice Theory Public Key Cryptography*, 2009, pp. 68-87.
- [5] F. Zhao et al., "Signatures for Content Distribution with Network Coding," *Proc. IEEE Int. Symp. Inf. Theory*, 2007, pp. 556-560.
- [6] Z. Yu et al., "An Efficient Scheme for Securing XOR Network Coding Against Pollution Attacks," *Proc. IEEE INFOCOM*, Apr. 2009, pp. 406-414.
- [7] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-Based Integrity for Network Coding," *Proc. Appl. Cryptography Netw. Security*, 2009, pp. 292-305.
- [8] P. Zhang et al., "Padding for Orthogonality: Efficient Subspace Authentication for Network Coding," *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1026-1034.
- [9] Y. Li et al., "RIPPLE Authentication for Network Coding," *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1-9.
- [10] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical Defenses against Pollution Attacks in Intra-flow Network Coding for Wireless Mesh Networks," *Proc. 2nd ACM Conf. Wireless Netw. Security*, 2009, pp. 111-122.
- [11] E. Kehdi and B. Li, "Null Keys: Limiting Malicious Attacks via Null Space Properties of Network Coding," *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1224-1232.
- [12] N. Cai and R.W. Yeung, "Secure Network Coding," *Proc. Int. Symp. Inf. Theory*, 2002, p. 323.