A Novel Homomorphic Against Intersession Coding attacks Using InterMac.

P.Sravani
Department of Computer Science & Engineering,
Gudalavalleru Engineering College

D.Deewena Raju
Department of Computer Science & Engineering,
Gudalavalleru Engineering College

Abstract – Network coding allows intermediate nodes to encode data packets, which increases network throughput and enhances robustness. However, once the node injects corrupted data blocks into a network, the polluted data blocks will propagate quickly with network coding. In the existing approach a new homomorphic MAC scheme called SpaceMac, which makes it possible for an intermediate node validating whether received packets remain in a selected subspace, even when the subspace is expanding eventually. Then, by using SpaceMac to be the building block to develop a cooperative scheme that gives complete defense against pollution attacks: (i) it might detect polluted packets early at intermediate nodes, and (ii) it could know the exact location of every, even colluding, attackers, thereby making it a possibility to eliminate them. This system fails to detects the malicious attacks in inter-session attacks of souce packets. In this proposed work, we'll first define precisely damaged packets in inter-session pollution based on the commitment of a given source packets. Within this system a new detection scheme: one hash-based and other MAC-based schemes i.e(Improved InterMac and SpaceMac). InterMac is the first multisource homomorphic HMAC scheme that supports multiple keys. All schemes provide in-network detection, are collusion-resistant, and have very low online bandwidth and computation overhead.

Keywords – **SpaceMac, Packet Session, Node, MaliciousNodes ,Pollutant Attack.**

I. INTRODUCTION

Network coding-based storage has undoubtedly received a great deal of attention in the network coding community. Independently, another body at the office has proposed integrity checking schemes for cloud storage, none of which, however, is customized for network coding storage or can efficiently support the repair of corrupted data. We observe that checking for integrity of coded packets resembles detecting corrupted packets of pollution attacks.

Network coding were first introduced in [1] is an innovative approach in achieving the capability associated with a network for multicast communications. Network coding allows intermediate nodes involving the source and to discover the destinations not just on store and forward but as well as to encode the received packets before forwarding them. In [2], Li et. al showed that it is sufficient to use linear coding which permits intermediate nodes to generate outgoing packets as linear combos of their incoming packets.

Regardless of whether intra- or inter-session coding is designed, network coding especially sensitive pollution attacks since it hinges on intermediate nodes to perform coding operations. Malicious nodes can inject corrupted packets into your network, and these packets are combined and forwarded by downstream nodes, which resulted in a lot of corrupted packets propagate within the network. This wastes network resources, such as bandwidth and CPU time, and subsequently prevents the decoding of one's original packets along at the receivers. 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 which 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 also for robustness to network impairments and packet losses. Various applications of network coding have therefore appeared directly from file download and content distribution in peer to-peer networks to distributed file storage systems. One method to deal with this problem is through authentication that consists of protecting the integrity of a message, validating the identity of the source, and guaranteeing the non-repudiation of one's source. This involves the usage of cryptographic primitives:(1) digital signatures, or message authentication codes(MAC) for computational security (i.e. vulnerable against an attack that's unlimited computational resources;(2) authentication codes for unconditional security (i.e.robust against an attack having unlimited computational resources)[1].

Assume the single-source multicast network coding model includes one source node, several intermediate nodes and receiving nodes, as shown in Figure 1. Here, the roles of source node, intermediate nodes and receiving nodes, which can be generally found in complex network coding environment.

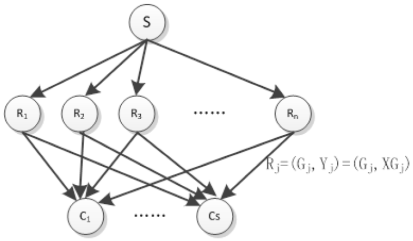


Fig. 1. Single-source multicast network coding model

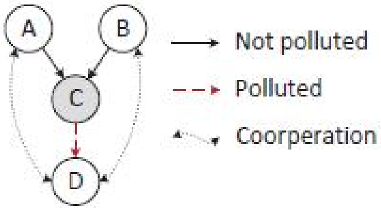
II. LITERATURE SURVEY

Wang et al. [5] introduced a light-weight non-repudiation protocol and use it to support their identification scheme forP2P systems. This scheme requires a distribution of multiple checksums (of most the blocks sent via the source) to the peers when an attack is detected, which is significant communication overhead. TESLA give the non-repudiation property, our identification scheme is to achieve significantly higher security per byte overhead than [5]while avoiding the need of checksum dissemination. However, the information-mixing nature of network coding also renders it more susceptible to pollution attacks than traditional store-and-forward paradigm. Think about a scenario in which a commercial data center is distributing a file to a range of customers using a network coded P2P network. An adversary pretends as a normal customer, by downloading and contributing packets of a given file. In this process, it generates corrupted packets and contributes them to be able to its peers. After being coded with other packets, a single corrupted packet can result in tens or even hundreds of polluted ones. This may cause legitimate users incapable of download the file properly[2].The present approaches against pollution attack to network coding can easily be categorized into two classes, information theory based and cryptography based. With information theory based approaches, solely sink is able to detect or correct data packet errors and the schemes of this type cannot prevent pollution propagation in networks, which leads to much bandwidth waste. Cryptographic approaches mainly focus regarding the best approach to create all nodes able to detect fake or polluted data packets and filter them. They prevent pollution propagation and reduce the bandwidth waste. There are a couple of classes of cryptographic approaches, public key cryptography(PKC)based [5] and symmetric cryptography based [4].

PKC based approaches enjoy high security but requires an enough large field . Their high computational complexities will lead to long time delay for data transmission and a lot more resource consumption at each node that data packets deal with. Dictated by broadcast security protocol TESLA [11], Y.Li et al. designs a RIPPLE scheme [6]. Their scheme enjoys low computational cost and allows arbitrary range of nodes to be compromised. However, their scheme requires loose clock synchronization of every nodes inside a network. Further, its implementation is founded on key delay distribution, which leadsto the largest data transmission delay.

Agrawal and Boneh [6] proposed a defense mechanism based throughout the homomorphism MAC scheme. This procedure relies on cover-free set systems for pre-distributing keys to provide in-network detection, and in consequence, only c-collusion resistant. It is also susceptible to tag-pollution attacks, where malicious node stamper with subsets of tags of one's packets they receive, as presented in [7].

Threat model: Within the existing approach both the source and to discover the receivers are trustworthy nevertheless the intermediate nodes might be malicious. Multiple pollution attackers, situated within an arbitrary variety of intermediate nodes inside the network. Each attacker may inject corrupted packets right into a single or multiple downstream edges to pollute the network. Some may also modify other data related with the packets, which can include, authentication tags. We perceive both cases wherein the pollution attackers launch their attacks independently or collude and coordinate their attacks. We further assume that the attackers are aware our defense scheme, i.e., the construction and use of SpaceMac. However, clone of other cryptographic approaches, we assume that the attackers’ running time is polynomial within the security parameter.



III. PROPOSED SYSTEM

The Proposed approach consists of the following four basic steps Modification Over the existing system:

1) *Key distribution:* The KDC distributes to the source *N* keys, which are used to produce *N* vectors. The KDC distributes a unique vector, which is a linear combination of the *N* vectors, to each node *g*.
2) *Tag generation:* The source uses the *N* keys to generate *N* tags for each data packet to be sent.
3) *Encoding:* An intermediate node encodes the correct data packets it received and sends the encoded data

packets to its downstream nodes.

4) *Verification:* Node *g* verifies the correctness of its received data packets using the vector it received.

Assume that a network consists of a source *s*, intermediate nodes and a set *R* of sinks. The source *s* multicasts a generation of *n* messages *M*1, ・ ・ ・,*Mn* to the sinks in *R*, where each *Mi*(1 ≤ *i* ≤ *n*) is a vector of *m* dimensions over field *Fp*

Before sending *Mi*, source *s* sets *M'* = (*Ei*,*Mi*), where *Ei* is a unit vector of *n* dimensions with *i* − *th* coordinate 1.

The random linear network coding is adopted. Let *W* = (*w*1, *w*2, ・ ・ ・, *wm*+*n*) be an encoded data packet, then

$$W = (w_1, \cdots, w_n) \cdot ((M_1')^T, \cdots, (M_n')^T)^T.$$

A key distribution center(KDC) is needed, which is used to support key distribution. Formally, a node judges that a message *W* = (*w*1, ・ ・ ・, *wm*+*n*) is fake or polluted by

$$W \neq (w_1, \cdots, w_n) \cdot ((M_1')^T, \cdots, (M_n')^T)^T.$$

we introduce two pseudo random generators(*PRG*), *G*1 and *G*2 are publicly known to the KDC and all nodes in the network

1) *Key Distribution:* KDC distributes *N* keys *k*1, ・ ・ ・, *kN* to source *s*, where *ki* □ *KG*1(1 ≤ *i* ≤ *N*). *V*1, ・ ・ ・, *VN* are used to generate tags for each data block to be sent. *G*1 is publicly known and KDC also knows *V*1, ・ ・ ・, *VN*.

For each node *g* except for source *s* in the network, KDC randomly selects a unique key *bg* □ *KG*2 as a seed. Set *Ug* = *G*2(*bg*) and *Vg* = *Ug***Vi*. KDC distributes a pair of keys (*bg*, *Vg*) to node *g*. (*bg*, *Vg*) is used to verify the correctness of data packets which node *g* receives subsequently.

2) *Tag Generation:* For each data packet *Mi* (1 ≤ *i* ≤ *n*) to be sent, the source generates *N* tags *ti*,1, *ti*,2, ・ ・ ・, *ti*,*N* with

$$t_{i,j} = V_j \cdot (M_i')^T, (1 \leq j \leq N).$$

Then, the source sends all data packets with their tags to its downstream nodes.

3) *Encoding: N* basic tags of each data block will be linearly combined in the same way as the data block.

4) *Verification:* Once node *g* with (*bg*, *Vg*) receives a data block with its tags (*W*, *t*1, ..., *tN*), where *W* = (*w*1, *w*2, ・ ・ ・, *wm*+*n*), it verifies the correctness of *W* by means of judging Whether

$$V_g \cdot W^T = U_g \cdot (t_1, t_2, \cdots, t_N)^T.$$

is satisfied or not, where *Ug* = *G*2(*bg*).

If equation is satisfied, the data packet *W* ia regarded as right; otherwise it is regarded as wrong and is discarded directly.

In this system we will use of *N* keys allows *N* − 1 nodes be compromised, which may attain the maximal number of compromised nodes for a scheme based on key pre-distribution and can assure that all nodes are able to detect the polluted data blocks.

IV. RESULTS

All experiments were performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operating system platform is Microsoft Windows XP Professional (SP2).

Existing results:

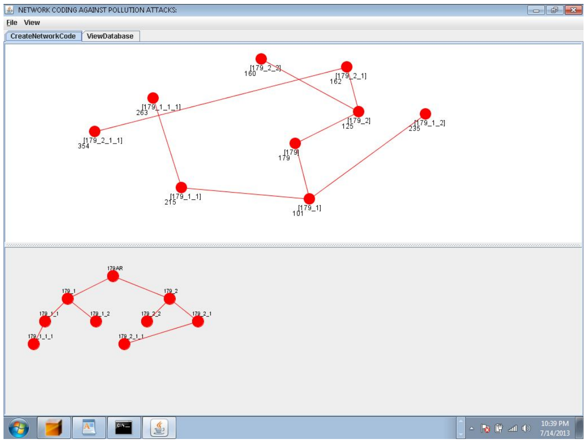
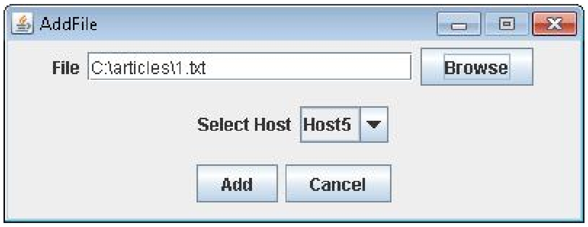


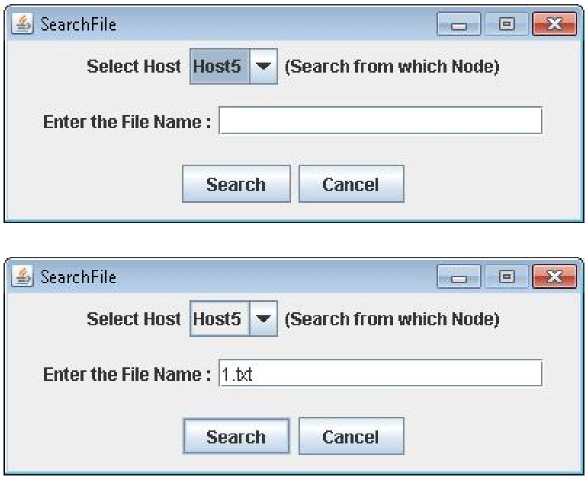
Fig1: Nodes creation with dynamic approach



File C:\articles\1.txt Browse

Select Host Host5

Fig2: Attaching Files to each nodes in the list



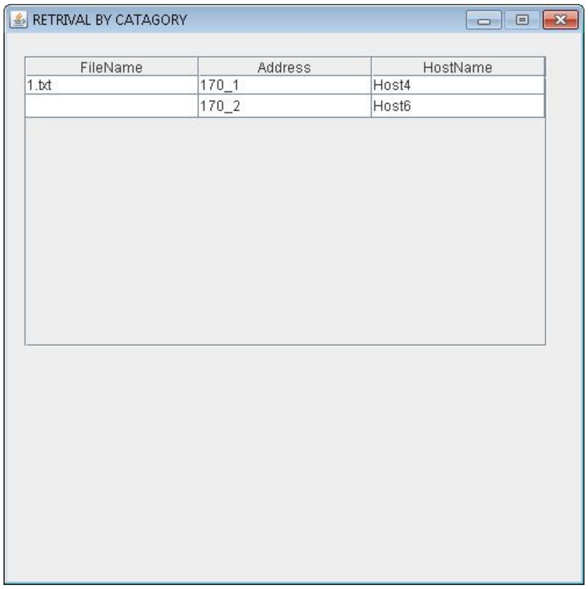
Select Host Host5 (Search from which Node)

Enter the File Name :

Select Host Host5 (Search from which Node)

Enter the File Name : 1.txt

Fig3: Searching each node attached file in the list



FileName	Address	HostName
1.txt	170_1	Host4
	170_2	Host6

Fig4: Node path list for each attached file

Proposed Results:

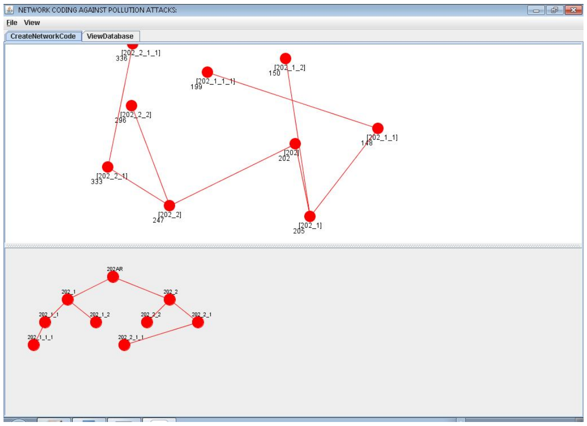
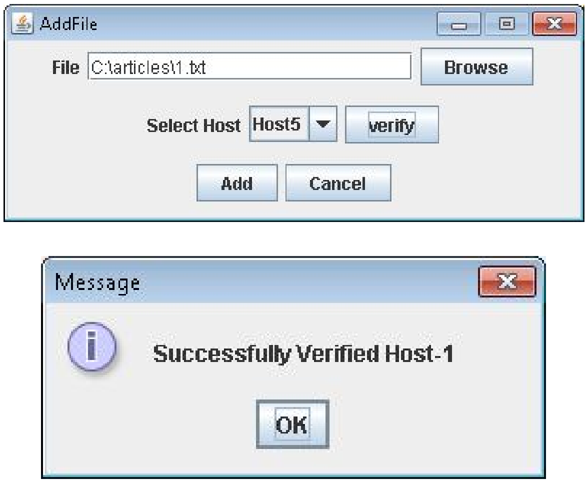


Fig5: Dynamic tag and node change



File C:\articles\1.txt

Select Host Host5

Successfully Verified Host-1

Fig6: Verifying each node before attaching file to the list of nodes

Host5 1.txt 202_2 fad94ffa1f5192b4ae813b6577147b55

Host4 1.txt 202_1 fad94ffa1f5192b4ae813b6577147b55

Fig7: Each host hashing details

V. CONCLUSION AND FUTURE SCOPE

The main idea behind our schemes is the use of commitment To mitigate the pollution attacks in network coding systems, a number of authentication schemes have been proposed in the recent years. However, there is lack of a systematical strategy to investigate the pros and cons of those schemes, and a comparison of their overhead of source packets to a trusted controller. Our work is an innovative adaptive security scheme which use time and space properties of network coding to dynamically adjust the authentication strategy of participating nodes according to the security situation. The first scheme is a novel combination of homomorphic and traditional hash functions. This paper improves the pollution detection algorithm, which reduces the false positive probability of pollution detection.

REFERENCES

[1] Multi-receiver Authentication Code for Network Coding Fr´ed´erique Oggier and Hanane Fathi, Forty-Sixth Annual Allerton Conference Allerton House, UIUC, Illinois, USA September 23-26, 2008

[2] Padding for Orthogonality: Efficient Subspace Authentication for Network Coding Peng Zhang, IEEE INFOCOM 2011

[3] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum energy multicast in mobile ad hoc networks using network coding," IEEE Trans. On Communications, vol. 54, no. 11, Nov. 2005.

[4] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in Proc. of IEEE International Symposium on Iriformation Theory, Jun. 2003.

[5] C. Gkantsidis and P. Rodriguez, "Network coding for large scale file distribution," in Proc. of IEEE INFOCOM, Mar. 2005.

[6] J. Dong, R. Curtmola, and C. Nita-Rotaru, “Practical defenses against pollution attacks in intra-flow network coding for wireless mes networks,” in *Proc. Second ACM Conference on Wireless Network Security*, 2009.

[7] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin, “Secure network coding over the integers,” in *Proc. International Conference on Practice and Theory in Public Key Cryptography*, 2010.

[8] M. Krohn, M. Freedman, and D. Mazieres, “On-the-fly verification of rateless erasure codes for efficient content distribution,” in *Proc. IEEE Symposium on Security and Privacy*, 2004.