ISSN:0975-9646

Schemes against Pollution Attack in Network Coding: A Survey

Neha V. Mamidwar[#1], Ms. Deepali Gothawal[*2]

[#]*Master of Computer Engineering*
D. Y. Patil college of Engineering, Akrudi,Pune, India.

[*] *Department of Computer Engineering*
D. Y. Patil college of Engineering, Akrudi,Pune, India.

***Abstract*— In Network coding method the flow of digital data in a network can be optimized by broadcasting digital evidence about messages. The "Digital Evidence" is the composition of two or more messages. When the bits of digital evidence reach at the destination the broadcasted messages are deduced, rather than directly recombine. Network coding allows all the intermediate nodes to encode incoming data packets before forwarding to downstream nodes. Network coding improves network throughput, robustness and reduce the network consumption. However if the malicious node injects fake data packet into the network, the pollution attack occurs into the network and it increases the propagation speed of the polluted data packet. There are various schemes against pollution attack which are discussed in this paper. These schemes are depends on some assumptions like topology, network controller. Hence the key predistribution based tag encoding scheme has been developed to deal with pollution attack and tag pollution attack which also has been discussed in this paper.**

***Keywords*— Network coding, pollution attack**

I. INTRODUCTION

Network coding is a method of optimizing the flow of digital data in a network by transmitting digital evidence about messages. The "digital evidence" is a composite of two or more messages. When the bits of digital evidence arrives at the destination, the transmitted messages are deduced rather than directly reassembling it [2]. Ahlswede et al. [1] proposed network coding which allows intermediate nodes in networks to encode several received packets into a single coded packet before forwarding. Contrarily, traditional coding techniques are referred to as source based coding, where only source nodes encode packets. Network coding is considered as a generalization of conventional store-and-forward routing techniques and it was originally proposed in order to achieve multicast data delivery at the maximum data transfer rate in single-source multicast networks. This feature had a great impact on the research field of information theory and research on network coding was first activated in the information theory community [3].

The concept of network coding was first introduced by R. W. Yeung and Z. Zhang in 1999 as an alternative to routing. In a traditional packet-switched network, data flow is defined as discrete "pieces" from the source to the destination like corpuscles in the bloodstream. At the transmitting station the outgoing message is broken into packets each of which contains some of the message with data intact in it. All the packets do not necessarily travel along the same route but eventually arrives at the same destination, where the receiving computer reassembles them into the original message. The main problem with this method is that when the overall network traffic volume is high, bottlenecks are common which results in long delays. Packets tends to bunch up at certain nodes, sometimes in excess of the nodes' ability to process them. Other routes and nodes may remain under-utilized [2].

In network coding data not only depends on transmitted messages but also on the contents of other messages that used to share the route at the time of transmission. For this reason, network coding is more resistant to hacking, eavesdropping and other forms of attack than traditional data transmission. The extent of throughput improvement that network coding can provide depends on the network topology, frequency and severity of bottlenecks. Network coding may prove to be useful in multicast networks, wireless sensor networks, digital file distribution and peer-to-peer (P2P) file sharing [2]. Although network coding was originally proposed in order to maximize throughput, to improve robustness performance in multicast communications, it has been adopted in a wide range of applications in computer networking [3].

As there are many benefits in Network Coding but with this there are some security issues with it. If there are a malicious node in a network and the malicious node injects fake data packets into its downstream nodes, the fake data packets will be encoded together with correct data packets by the downstream nodes and the outputs of the downstream nodes will be polluted and fake. The pollution propagates in the network quickly with the transmission of polluted data packets, which not only leads to incorrect decoding at sinks but also wastes network resources. So it is crucial to prevent pollution attacks in practical applications of network coding.

For example, the applications built on top of network coding are vulnerable to *pollution attacks* in which the compromised forwarders can intentionally pollute the transmitted messages or inject the forged messages into networks. These attacks prevent the sinks from recovering the source messages correctly. A more severe problem is pollution propagation that is even a small number of polluted messages can quickly propagate into the networks

and infect a large proportion of nodes, because each polluted message can be used by all downstream nodes. Therefore, the polluted messages should be detected and filtered as early as possible. There are many existing systems to deal with the pollution attack which are discussed in this paper.

This paper surveys literature on several approaches against pollution attack in network coding. In this paper the focus is on the Error Correction based scheme, Malicious node localization based scheme, and Detection of pollution attack schemes. The remainder paper is organised as follows: In section 2, the related work is given where all the existing techniques against pollution attack has been discussed. In section 3, network model and assumptions and the tag encoding based scheme has been discussed.

II. Related Work

A. Error Correction based Scheme

Error Correction is one of the scheme against pollution attack. Whenever any malicious node present in the network, it injects fake or false data packets into the network, due to which network gets polluted. That means when some sought of error occurs into the any of the packet in the network, it affects the whole network. So correcting errors occur into the network is called error Correction. Error can caused by an reason like noise or intermediate jamming. In Linear network coding the schemes are largely divided into error correction and error detection based scheme. In error detection based schemes, errors are normally detected at intermediate forwarding node while in error correction based scheme errors are generally corrected t sink nodes. These schemes are generally designed based on the knowledge of the network topology which makes the these schemes less flexible to the current network.

Korhn et al. [7] proposes to use homomorphic hash function to guaranty the correctness of network. The main idea is that each intermediate node will check the correctness of the messages. If the packet does not pass the check at an intermediate node, it will be discarded. This approach can reduce the communication overhead and can use in random network coding. However computational complexity is still very high. When the network scale is large, computing too many hash values also creates high delay. To address all these defects Khedi and Li [6] developed a simple error detection based Null keys. The main idea is to partition the n-dimensional linear space over GF(q) into two orthogonal subspaces of dimension k and n-k.

Comparing to the homomorphic hash function more excitant and virtually no message delay. Unfortunately all these schemes have one weakness, all the corrupted packets will be discarded. As it is known that, in packetized network, a large packet is divided into small fragments to transmit. As long as malicious node can corrupt one fragment in the whole packet, this fragment will be discarded and in this way net transmission efficiency can be close to zero.

In [17], cai and yeung proposed a technique to correct error at sink nodes using error correcting network coding. They derived the hamming bound and the Gilbert vershmov bound. Charles et al. [18] use the cryptographic idea to discard the corrupted packets. Error correction based approaches [5] provide error tolerant decoding at sink nodes. Neverthless, as a passive defense, error correction is applicable only when there are a limited number of corrupted blocks in the network and achievable own rate is determined by the number of contaminated links.

B. Malicious Node Localization base Scheme

Some schemes [11], [12], [13] are proposed to locate malicious nodes and make those nodes unable to further inject polluted data packets. In [11], Anh Le, and Athina Markopoulou propose a novel homomorphic message authentication code (MAC) scheme for expanding spaces called SpaceMac. SpaceMac allows a node to verify if it's received packets belong to a specific subspace even if the subspace is expanding over time. Then designed a novel, cooperative defense system which includes both a detection scheme and a locating scheme using SpaceMac as their building block. The detection scheme relies on SpaceMac to force intermediate nodes to send only linear combinations of packets that they actually receive from their parents. Parents and children of any intermediate node cooperate to detect corrupted packets sent by the intermediate node. The locating scheme uses SpaceMac to force nodes in the network to truthfully cooperate with a central controller so that the controller can exactly locate the pollution attackers. Finally, by leveraging multiple generations scheme is able to deal with an arbitrary number of colluding attackers.

In [12], a malicious node Identification Scheme (MIS) that identifies and isolates malicious nodes, so that the pollution attack can cause harm to the network for a short period of time only and the subsequent streaming will no longer be influenced. MIS is block-based in that a malicious node can be identified rapidly as long as it injects a single bogus block. To unambiguously identify malicious nodes, a novel and light-weight non-repudiation transmission protocol are designed to ensure that any node that has injected a bogus block cannot deny its behaviour and any malicious node cannot disparage any innocent node. MIS can fully satisfy the requirements of live streaming. In MIS each node only needs to perform a small number of hash computations for an incoming/outgoing block, incurring computational latency in the range of several microseconds, which is significantly smaller than most previous schemes. Besides, each block only carries a 20-byte evidence code, introducing much smaller communication overheads than any existing schemes. The verification information given to each node is independent of the streaming content and thus does not need to be redistributed. Furthermore, MIS is scalable to large networks and is effective even in the presence of a large number of malicious nodes.

In [13], a novel is proposed, complete defense system for network coding-based P2P systems that can quickly detect corrupted blocks, precisely identify the attackers, thereby eliminating them from the network, resist arbitrary collusion, and work with unknown, dynamic topologies, as it is the case in P2P systems. This scheme uses and builds on two key ingredients: homomorphic message authentication codes and time asymmetry (as in TESLA) to provide source authentication for the detection scheme and

non-repudiation for the identification scheme. This mechanisms introduce significantly less communication and computation overhead than other comparable state-of-the-art schemes for P2P systems.

The limitation of this schemes is that, it either assume that there exists a powerful controller that knows the entire topology of a network [11], [12] or assume a clock synchronization of all nodes in a network [13]. Those schemes are of limited practicality when multiple malicious nodes exist.

C. Pollution Attack Detection Scheme

The existing schemes of pollution detection, which are mainly based on key delay distribution, public key cryptography (PKC), and key predistribution, focus on detecting and filtering fake or polluted data packets at intermediate nodes or sinks directly and can prevent pollution propagation efficiently. The schemes in [14], [15] based on key delay distribution require a clock synchronization of all the nodes in a network. So it is difficult to implement them in an adversarial distributed environment. The implementation of the schemes based on PKC or key predistribution are relatively simple.

In [16], an efficient signature-based scheme against pollution attacks on linear network coding systems has been proposed. In this scheme, the source signs its messages using its private key, while other nodes verify the received messages using the source's public key. This scheme utilizes a novel homomorphic signature function, which allows forwarders to compose the signatures for their output messages from those of input messages using the similar way that the output messages are composed from the input messages. Since each node appends the signatures to its output messages, its downstream nodes can verify the received messages effectively and discard the polluted or forged ones. It has been proved that finding a hash collision message in our scheme is equivalent to solving a hard discrete logarithm problem. Experimental results show that this scheme is ten times faster than some existing one. In addition, an alternate lightweight scheme based on a much simpler linear signature function. This alternate scheme further improves computation efficiency and is more suitable for resource-constrained networks such as wireless sensor networks. However, it introduces a trade-off between efficiency and security. This scheme allows the source to delegate its signing authority to the forwarders. That is, the forwarders can generate the signatures for their output messages without contacting the source, but they cannot create the valid signatures for polluted or forged messages. It does not need any extra secure channels, and can provide source authentication and batch verification. Most importantly, it is much more efficient than existing ones.

In [17], an efficient dynamic-identity based signature scheme for secure network coding has been proposed, which features the notable properties like Efficiency, security, Scalability. This signature scheme can support fast identity based batch verification, and rapid signature generation for the output packets. By employing two optimized verification techniques, packet-based and generation-based batch verification methods, a node can quickly verify multiple received packets in batch such that the total verification cost can be dramatically reduced. Hence this scheme effectively eliminates the performance bottleneck due to the greatly reduced computational overhead at forwarders. Moreover, with identity-based signature, both certificate management cost and the transmission overhead can be significantly reduced. To address the security and robustness of this scheme, a Multi-level Binary Authentication Tree (M-BAT) approach is proposed for detecting pollution attacks. In addition, with the one-way dynamic-identity based signature function, the scheme can efficiently thwart random forgery attack, which exists in most of reported homomorphic signature schemes for network coding. This scheme also does not need any extra secure channel, and provides source authentication via one-way identity hash-chain. In this scheme, the signature keys can be updated with one-way pseudo-identity refreshing in a natural way, while the public keys keep invariant. Therefore, it is more efficient for transmitting live data or distributing multiple files with the same public keys. However the limitation of such scheme is that, it requires a large field, which implies that the computational complexities of PKC-based schemes are very high.

There are some scheme [18] which are based on key predistribution. In [18], the first scheme proposed for securing XOR network coding systems against pollution attacks. This scheme allows the polluted messages to be filtered at the forwarders, and it works not only for XOR network coding, but also for for normal network coding. This scheme exploits probabilistic key pre-distribution and message authentication codes (MACs). In this scheme, the source produce multiple MACs for each message using its secret keys, where each MAC can authenticate only a part of the message and the parts authenticated by different MACs are overlapped. Every encoded message is attached with the MACs of the source messages from which it is constructed. Therefore, multiple downstream forwarders can collaboratively verify different parts of the encoded message using the MACs and their own shared keys. By carefully controlling the overlapping between the parts authenticate by different MACs, this scheme can filter polluted messages in a few hops with a high probability. The computational complexities of the schemes in [18] are low, but they experience tag pollution attack that leads to numerous correct data packets being discarded. With the scheme in [18], the correctness of a data packet will be verified after several hops.

III. DISCUSSION

In the previous section, all the approaches against pollution attack has been discussed. The approaches are categorized into three classes i.e., Error Correction based approach, Malicious node localization, and Pollution detection. All these approaches has some limitations which are discussed above. Like Error correction based approach can be used only when small portion of packet is polluted and correction can be done only at sink node. Malicious node localization scheme, only applicable when there are limited number of malicious nodes. And other schemes which are based on PKC, key predistribution require large field and in

some cases very large number of packets needed to append to each packet. So to overcome all these limitation a new scheme has been generated based on tag encoding. The detection polluted data packets, a Key Predistribution-based tag encoding (KEPTE) schemes has been proposed.

Before one can dive into key predistribution based tag encoding scheme there are certain building blocks which are require to implement the KEPTE scheme. The preliminaries are discussed in the next subsection.

Preliminaries

A. Network Coding Model

In traditional packet forwarding, only source node allowed to encode the message or packets before it gets forwarded to its downstream nodes. Whereas, in network coding the intermediate nodes are allowed to perform computation on input messages, which gives the output messages that are the mixture of input ones. Packet tagging and buffering are key for network coding to be practical.

In practical network coding, if the file is too large is generally divided into its subfiles, which are called as generations or groups. Each generation is further divided into some data blocks with *h* packets in each blocks. All coded packets related to the *k*th block belong to generation *k* and random coding is only performed among the packets in the same block. Packets within a generation need to be synchronized by buffering for the purpose of network coding in intermediate nodes. So KEPTE can be perform in each generation as a separate file.

B. ADVERSARY MODEL

There are some assumptions which are taken while implementing the KEPTE scheme. Here the source node is assumed as trustworthy. An adversary may compromise some intermediate nodes or sink nodes and secret information are held by those compromised nodes. The attack can be of two types viz, Pollution attack and tag Pollution attack. The Pollution attack are the attacks in which malicious attack inject fake data packets into the output its links. And tag pollution attacks are the attack where a compromise node can modify the tag of correct data packet and inject correct data packet with modified tags to its output links. The objective of the pollution attack is to make intermediate or sink nodes to detect error data packets, which not only leads to incorrect decoding at sinks but also makes polluted data packets be transmitted in a network, leading to bandwidth waste. The objective of the tag pollution attack is to get correct data packets be judged as wrong and be discarded by intermediate nodes or sinks, which wastes bandwidth greatly.

Key Predistribution-based tag encoding (KEPTE) scheme

The basic idea of key predistribution based tag encoding scheme is as follows**:** Let *S* be the source, *R* be the set of sink nodes and g_i be the intermediate nodes. For each data packet there will be a tags given by source node using N keys. All the intermediate node g except source node holds two keys, (Z_g, V_g) which will be used for encoding and decoding purpose and Z_g, V_g and N held by S satisfies certain relation. The intermediate node g receives data packets W with N tags, it uses Z_g to encode data packet and generates new tag t that can be viewed as tag generated according to W and Z_g. The correctness of the packet W is verified by the g with V_g and t

As compare to the existing key predistribution based scheme, KEPTE scheme gives good performance. As it does not require a large field, it is computationally efficient. All the intermediate nodes and sink nodes are able to detect pollution attack and tag pollution attack. Also it has high fault tolerance ability. In KEPTE, the key distribution center is used, which is a common tool for the key distribution. Public key cryptography (PKC) can be used instead of KDS, if it is not available. By PKC, the source distributes secret information, which is also secret keys, to each node *g* except the source in a network.

The process of KEPTE include some steps which are as follows:

1. *Setup*: In this step the KDC distributes N secret vectors to the source, and distributes two secret vectors to each of the intermediate node.
2. *Tag Generation*: For each incoming packet, the source uses the algorithm *Sign* to generate N tags. The Sign algorithm computes N tags for each of the n data packets, where N is a security Parameter.
3. *Encoding*: Assume that intermediate node receives h correct data packets each with N tags. For an output link an intermediate node randomly selects h constants and perform the algorithm *Combine* to generate a new encoded data packet with N tags as the output of this link. The Combine algorithm produce N tags for linear combination of multiple data packet.
4. *Verification*: Upon receiving a data packet with its N tags an intermediate node checks the correctness of data packet with algorithm Verify and its secret vectors. If its output is 1, intermediate node judges being correct: otherwise intermediate node judges data packet is fake or polluted and discard. The Verify algorithm checks the correctness of the data packet with its N tags.
5.

IV. CONCLUSION

This paper presented the literature survey on the schemes against Pollution attack in Network Coding. Network coding serves many benefits however when there is a malicious node in the network it pollutes complete network. It inject fake data packets onto the network and due to which network gets polluted and it is necessary to have solution for such problems. To deal with such attacks there are some schemes against the pollution attack. In this paper the various schemes for the pollution attack has been discussed along with limitations also the key predistribution tag encoding scheme against pollution attack has been discussed. The main advantages of KEPTE are, it is computationally efficient, all the intermediate nodes and sink nodes are able to detect pollution attack and tag pollution attack. These system can be implemented in P2P system where any node could upload and download data packet as sink node or intermediate node.

ACKNOWLEDGMENT

The authors would like to thank the publishers, researchers for making their resources available and teachers for their guidance. We also thank the college authority for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network Information Flow," IEEE Trans. Information Theory, vol. 46, no. 4, pp. 1204-1216, July 2000J.

[2] Rouse, Margaret. 'What Is Network Coding? - Definition From Whatis.Com'. Search Networking. Np., 2015. Web. 26 Oct. 2015.Survey paper: network coding and its application

[3] Matsuda, Takahiro, Taku Noguchi, and Tetsuya Takine. "Survey of network coding and its applications." IEICE transactions on communications 94.3 (2011): 698-717.

[4] Wu, Xiaohu, et al. "A tag encoding scheme against pollution attack to linear network coding." *Parallel and Distributed Systems, IEEE Transactions on* 25.1 (2014): 33-42.

[5] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," IEEE Trans. Information Theory, vol. 54, no. 6, pp. 2596-2603, June 2008.

[6] E. Kehdi and B. Li, Null keys: Limiting malicious attacks via null space properties of network coding, in IEEE INFOCOM 2009, pp. 1224 1232, Apr. 2009.

[7] M. Krohn, M. Freedman, and D. Mazieres, On-the-y veri_cation of rateless erasure codes for e_cient content distribution, in IEEE Symposium on Security and Privacy 2004, pp. 226240, May 2004.

[8] T. C. Ho, B. Leong, R. Koetter, M. M´edard, M. Effros, and D. R. Karger. Byzantine modification detection in multicast networks using randomized network coding. In International Symposium on Information Theory, Chicago, USA, June 2004.

[9] Qiao, Wenbo, Jian Li, and Jian Ren. "An efficient Error-Detection and Error-Correction (EDEC) scheme for network coding." *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011.

[10] A. Le and A. Markopoulou, "Cooperative Defense against Pollution Attacks in Network Coding Using SpaceMac," IEEE J. Selected Areas in Comm. on Cooperative Networking Challenges and Applications, vol. 30, no. 2, pp. 442-449, Feb. 2012.

[11] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, "Identifying Malicious Node in Network-Coding-Based Peer-to-Peer Streaming Networks," in *Proc. IEEE Mini INFOCOM 2010* San Diego, CA, USA, pp. 1-5

[12] A. Le and A. Markopoulou, "TESLA-Based Defense against Pollution Attacks in P2P Systems with Network Coding," Proc. IEEE Int'l Symp. Network Coding (NetCod), July 2011.

[13] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical Defenses Against Pollution Attacks in Wireless Network Coding," ACM Trans. Information and System Security, vol. 14, no. 1, article 7, May 2011.

[14] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE Authentication for Network Coding," Proc. IEEE INFOCOM, Mar. 2010.

[15] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Signature-Based Scheme for Securing Network Coding against Pollution Attacks," Proc. IEEE INFOCOM, Apr. 2008.

[16] Y. Jiang, H. Zhu, M. Shi, X. Shen, and C. Lin, "An Efficient Dynamic-Identity Based Signature Scheme for Secure Network Coding," Computer Networks: The Int'l J. Computer and Telecomm. Networking, vol. 54, no. 1, pp. 28-40, Jan. 2010.

[17] W. Stallings, Cryptography and Network Security, fifth ed. Prentice Hall, 2011.

[18] D. Charles, K. Jain, and K. Lauter, Signatures for network coding, in Proc. of CISS06, pp. 857863, 2000