International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Detection and Prevention from Pollution Attacks in Network Coding

Dr. Sonia Sharma[1], Kiranbir Kaur[2]

[1]HOD, Department of CSE, MIMIT, Malout, Punjab, India.
[2]Student of CSE, MIMIT, Malout, Punjab, India.

Abstract - *When packets are transmitted from source to destination in a network, they suffer from pollution attacks. Malicious nodes inject malicious packets in the network which affects the throughput of the network majorly. For this, network coding came into existence, which helps in broadcasting the coded packet rather than a simple store and forward traditional method. Network coding does not only allow source node to encode the packet but also intermediate nodes can encode the packets before forwarding to the downstream nodes and it provides methods for detection of fake data packets not only at sink nodes but also at intermediate nodes. In this paper various schemes against pollution attacks to data are discussed which helps in transmitting the data in a secured manner. These methods have greater complexities due to all the computations but the protocol discussed in this paper is very optimistic due to (a) low computational complexity; (b) the ability that intermediate nodes can also detect pollution attack and tag pollution attack; (c) high fault-tolerance ability. The existing key pre-distribution based schemes aiming at pollution detection can only achieve to some extent. So proposed scheme uses tag encoding with tracing mechanism detect pollution attacks and malicious node.*

Key Words: Network coding, pollution attacks, sink nodes, complexity, malicious nodes.

1. INTRODUCTION

Network coding is a method of optimizing the flow of digital data in network by mixing up the packets and making the optimized use of the transmission channels. In network coding the message is encoded and decoded at the source and destination nodes respectively. The received message is deduced at destination rather than simply combining the data packets in a full complete message [2]. Ahlswede et al. [1] proposed network coding which allows intermediate nodes to encode the number of packets and then forwards it to the downstream nodes. Network coding was originally proposed to achieve maximum throughput while multicasting data in a single-source multicast networks. This feature had a great impact as it helped in achieving the maximum throughput and great performance within a network.

The concept of network coding was first introduced by R. W. Yeung and Z. Zhang in 1999 as an alternative to traditional routing schemes. The outgoing data is divided into packets by the transmitting node, each of which contains some part of the message with data intact in it. It is not necessary that all the packets follow the same route but at the end all packets arrive at destination, where the receiver reassembles the packets into a single message. The main issue with this method of transmission is that when traffic is more in the network, there is possibility of bottlenecks which may result in long delays. And other nodes may remain under-utilized and other routes also might be free [2][13].

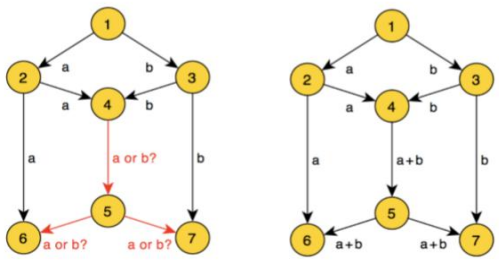

Figure 1. Canonical network coding example: node 1 multicast bits a and b to nodes 6 and 7. If node 4 did not perform a simple encoding operation on the incoming bits, the maximum network capacity of 2 could not be achieved.

Due to network coding, the network becomes more resistant to hacking, eavesdropping and other forms of attack than traditional data transmission scheme[15]. Factors like Network topology, frequency and severity of bottlenecks highly affects the throughput of network. Network coding has proved to be useful in multicast networks, wireless sensor networks, digital file distribution and peer-to-peer (P2P) file sharing [2]. Network coding not only provides maximized throughput but it has also been adopted in a wide range of applications in computer networking [3].

Although network coding has many benefits but it has some security issues due to which the performance of a network deteriorates. A small number of polluted messages can pollute the whole network because these messages are received by all the downstream nodes which affect a large proportion[16]. Therefore, the polluted messages should be detected as soon as possible. There are many existing schemes to deal with pollution attacks which are discussed in this paper.

This paper surveys on several approaches against pollution attacks. The remainder paper is organized as follows: In section 2 related work is given where all the available techniques are discussed which are used to prevent pollution attacks or to detect them. In section 3 network model and in section 5 the scheme which is proposed in this paper or methodology is discussed. In section 6 overhead analysis based on the parameters like computational complexity, storage and communication overhead is done.

2. RELATED WORK

There are several schemes[19] proposed for the authentication to detect the polluted packets at intermediate nodes rather than the sink nodes based upon cryptographic functions with computational assumptions [5].

2.1 Hash Functions - Korhn et al.[7] proposed the use of hash functions[6] which are used to generates hash values that are used to check the integrity of transmitted packet at nodes. If a packet does not pass the verification, the packet is discarded. These hash values are distributed over a pre-established secure channel. In this scheme communication overhead is reduced but it suffers high complexity if the network is large. This approach has a weakness that all corrupted packets are discarded, as we know a message is divided into number of packets and if one packet is discarded due to pollution attacks, the net transmission efficiency will be close to zero[14]. Charles et al. [8] used the cryptographic idea to prevent the message from being polluted due to polluted fragments. Some error correction based approaches [9] provide error tolerant decoding at sink nodes but it is only applicable if the corrupted blocks are limited.

2.2 Message Authentication Code (MAC) – In [10] a homomorphic MAC scheme is proposed, which ensures the authentication of the message. It introduced the concept of tags[17] which are appended to the packets to confirm its authenticity. It works same as hash function but the main difference is that it uses secret key to generate MAC value which is used for the authentication of the message[18] and sent along with the data itself which is in encrypted form. The sender and receiver share a symmetric key. It has a major drawback that the receiver can reject the packet if the tag has been polluted but the message is intact. But it does not ensure nonrepudiation.

2.3 RSA Algorithm - The scheme proposed in [11] is based on RSA Algorithm in which intermediate nodes can authenticate the packets which are being transmitted using their own key which is distributed to these nodes without knowing or sharing the secret key of the sender which increases the security of the packets. These intermediate nodes generates verifiable signature and for this task one key pair is required which verifies the correctness of a packet. Key Generation, Encryption and Decryption are the main steps. It provides high security because it is difficult to factor the large numbers but very complex due to large computations.

2.4 Signature Scheme / Digital Signature – The scheme proposed in [12] utilizes a homomorphic signature function where the source node uses its private key to sign the message while source's public key is used to verify the received message. These are public-key primitives of message authentication. It binds an entity to the data and the data is signed by the signer using their secret key which verifies that the message belong to the sender and if the digital signature does not match, the packet is discarded assuming it is polluted same as MAC.

Table -1: Comparison of various schemes based on some key factors

	Computational complexity	Communication overhead	Storage Overhead	Security
Homomorphic Hash Functions	High if network is large.	Low as polluted packet is discarded immediately.	Hash values are stored in a simple array like data structure.	Has provable security against collision attacks.
Message Authentication Code	Only MAC values are computed using secret key.	Keys are shared beforehand.	MAC values are stored which are sent with the message.	Only provides the authenticity of message using MAC.
RSA Algorithm	Computationally expensive and time consuming to generate key pairs.	Easier to send public key.	Storage is high because keys can be 1024 or 2048 bits long.	More secure as it is difficult to factor the large numbers.
Signature Scheme/Digital Signature	Signature computation using signer's private key	It is low because signature is generated for hash of data rather than data itself.	More storage as signature is for given hash value and signature key.	It provides non-repudiation, authentication and data integrity.

3. DISCUSSION

In the previous section all the important approaches which provide defence against pollution attacks have been discussed. But all these approaches have some limitations or other. Like Hash functions can take longer time to generate hash values for large networks which results in high delay. MAC Scheme is only applicable if users have shared secret key prior to the use of MAC and it does not provide a non-repudiation service as the user can deny the message sending. RSA Algorithm even though addresses the key distribution issue as it uses asymmetric key but it requires a third party to verify the reliability of public keys and the middleman can temper the public key system. These key predistribution schemes require large fields and sometimes very large data is appended to the packet. These methods suffer from clock synchronization delay and computational complexities that affects the performance of the network. So to overcome all these limitations a scheme has been generated which is based on tag encoding in which pollution attack and tag pollution attack is detected at nodes and a tracing procedure is followed which notifies about the contaminated link and that malicious node is identified and removed from the network. The main feature of this scheme is that the computational complexity is very low and it is more secure because keys are distributed before the transmission of data. Instead of large data fields it uses small sized tags.

4. PRELIMINARIES

4.1 Network Coding Model

In traditional packet transmission the encryption is done only by the source node and then it is forwarded to the downstream nodes whereas in network coding intermediate nodes are also involved in encoding. Packet tagging and buffering are key for network coding. In practical network coding if the data which is to be transmitted is very large then it is divided in number of fragments, which are called as generations or groups. Each group is further divided into data blocks each containing number of h packets respectively. The packets related to kth block belong to that generation and then coding is performed on each block separately. Packets which belong to a generation are synchronized by buffering to perform network coding at intermediate nodes. So KEPTE can be performed on each generation as a separate file.

4.2 Adversary Model

To implement this scheme some assumptions are made. Like the source node is considered to be trustworthy and it is assumed that the pollution attacks are done during the transmission. The attacks are mainly – Pollution attacks and Tag pollution attack. The pollution attacks are those in which a malicious node injects fake data packets. And the pollution attacks refers to the modification of tag which is used to ensure the integrity, authentication, non-repudiation of the message which is sent using various schemes like RSA Algorithm, MAC Scheme, Digital Signature etc. In tag pollution attacks, even if the tag is polluted but data is intact, still the packet will be discarded due to the failed verification. Due to both pollution attacks and tag pollution attacks bandwidth is wasted.

Table -2: Security Goals

	Hash Function	MAC	Digital Signature
Integrity	✓	✓	✓
Authentication	✖	✓	✓
Non-repudiation	✖	✖	✓

5. METHODOLOGY

5.1 KEPTE (Key Predistribution tag encoding) Scheme – As we have discussed, the above described schemes suffer from tag pollution attacks which leads to numerous correct data packets to be discarded. Since it is a key predistribution scheme the computational complexity is very low. A Key Distribution Center (KDC) is responsible for distributing the N secret vectors to the source node, which distributes a pair of these keys to all the nodes participating in data transmission. The source node also generates tags for all the data packets using Message Digest algorithm. And then these packets are transmitted along with their corresponding tags which are further coded for transmission which acts as encapsulation. Even though KEPTE prevents tag pollution attack and provides verification at nodes, but if at some intermediate node verification fails, an alternative and shortest path from the node before the failed/malicious node is found out, for which we will be using trackback mechanism which will notify about that malicious node and we can remove that node from the network for future convenience. This technique follows the basic idea as:

A Source node 'S', set of sink nodes denoted as 'R' and let 'g_i' be the intermediate nodes to be considered in the system. The source node distributes tags for each data packets using N secret vector. Let (Zg,Vg) be the pair of keys which are distributed to intermediate nodes for encoding and decoding purpose. When the intermediate node receives W packets with N tags, it uses Zg key to encode the packet and a new tag is generated, which encapsulates all the packets received as W. And then for the verification of the received packet W is done with Vg and tag t. KEPTE works as follows:

5.1.1. Setup: This is the initialization phase in which the Key Distribution Center sends the N secret vectors to the source as X1,X2,…XN. The data transmission is based on the shortest path from a specific node to its neighboring nodes. After receiving the secret vectors the source node distributes two keys (Zg, Vg) to each of the node participating in data transmission. The pair of these secret vectors is selected on basis of the condition described below:

$$Vg = Zg(X1, X2 \ldots XN)$$

5.1.2. Tag Generation: In tag generation, source node generates tags for all the incoming data packets which is used as a unique identity of that packet. Tag generation can be done with any *signing* algorithm. For example, we can use Message Digest (MD5) algorithm, which produces 128-bit hash value using homomorphic hash functions which is expressed in 32-bit Hexadecimal number or a 160-bit Elliptic Curve DSA. When the tags are generated for all the data packets, these are transmitted to the intermediate nodes.

5.1.3. Encoding: As the intermediate nodes receive the data packets along with their tags, these perform network coding on these packets and *combine* all the data packets into a single new encoded packet with N tags, received with all the packets, which forms a double layer protection/encapsulation for the data packets being transferred.

5.1.4. Verification: The main advantage of this scheme is that it not only checks the correctness of data packet at sink nodes but also at intermediate nodes. Each intermediate node *verifies* the data packet by comparing it with received secret vectors (Zg, Vg). If the verification fails which means the output is 0, the packet is discarded considered as fake data packet at the intermediate node itself and it is not further transmitted so it saves the bandwidth from wastage. Otherwise it is transmitted to further nodes as it passed the verification. The verification is based on the given condition:

$$Zg.\ (T_{w,1}, T_{w,2,} \ldots T_{w,n}) = W.\ Vg^{T}$$

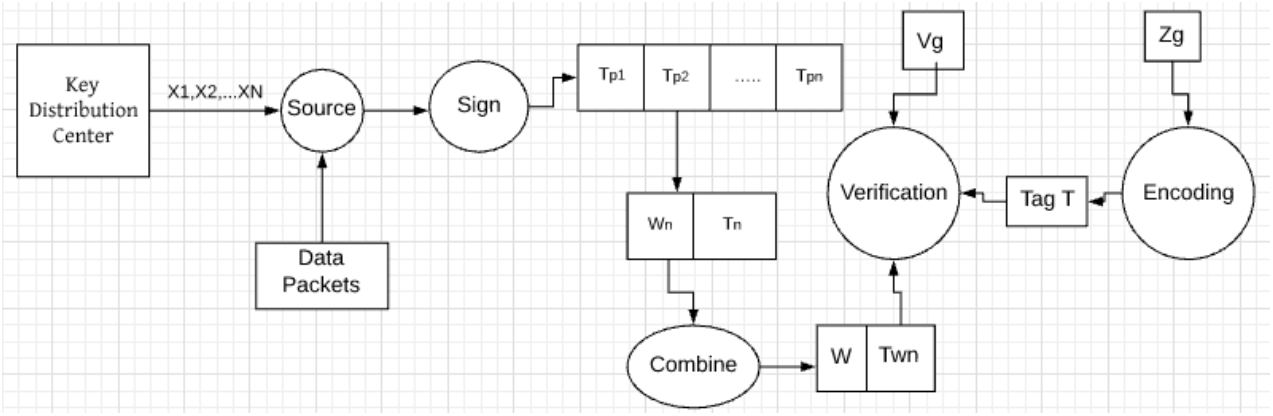

Fig -2 : Working of scheme

5.2 Tracing Procedure: In addition to providing protection against pollution and tag pollution attacks, this procedure attempts to locate malicious node. This procedure requires MAC tags to be sent with each transmitted packet. If any inconsistency is identified, an alert message is sent to the controller. Then controller asks nodes the information about the packets received from upstream nodes and is able to identify the polluting nodes.

In this proposed tracing procedure, nodes are using tagging scheme to verify the data packets and identify a pollution attack. Whenever at a node the packets fails the verification, the node sends a bad link notification to the source (malicious packet is included) of that link and transmission on that link is avoided in future. After this the source node initiates a traceback procedure to identify the pollution attacker.

5.2.1 Algorithm:

Input: A polluted packet P` and a node reporting pollution, say R.
Output: Identification of Attacker Node.

1. Let **P** be the correct packet.
2. **If** P == P`, both are identical // Pollution Notification is fake
3. **Return** Node R.
4. Set P = R // P is the node which has to prove its correctness.
5. Source node finds the incorrect bit in P` say $b_{u.}$ // u is the bit position.
6. **Loop**
7. Query **P** for tuples for bit u in input packets.
8. Compare Σb_i and $b_{u.}$ // b_i is the value of bit u in input packet.
9. **If** $\Sigma b_i == b_{u.}$ **//P** codes correctly.
10. **For each** tuple Query n_i for cross-examination
11. **If** n_i returns "no" then query P for signed packet p_i
12. **If** p_i is from n_i and bit u is b_i

13. **Return** node n_i
14. **Else Return** node **P // P** is exonerated.
15. Set **P** = n_i and $b_u = b_i$ and repeat the traceback procedure.

6. OVERHEAD ANALYSIS

6.1 Computational Complexity: This scheme has very low computational complexity because of tag generation which are generated for once and there is no additional computations at intermediate nodes while encoding or verifying the messages. At the *setup* phase computational complexity is very low as there is only transfer of secret vectors. In the *tag generation* and *verification* phase computations are done to generate tags and verification checking, so there is some computational complexity but very low in comparison to other previous schemes. In other words, the key & tag generation at source and key generation & verification of correctness of each packet at each node g comprises the computational complexity of KEPTE. For example if using the 160-bit Elliptic curve DSA signature, this signature generation/verification can be performed in 1ms.

6.2 Storage Overhead: Storage overhead is also very less as the secret vectors are of very small size. The source node stores the secret vectors which contributes about 54.7kB of total size and these are only stored for the completion of second phase of this scheme. Once the tags are generated from these secret vectors, these can be removed, so the storage overhead is also low. Storage units involve only the tag size and keys which are distributed to intermediate nodes for en/decoding purpose which constitutes a very small size memory. Each encoding and decoding node is required to store the received packets for verification if traceback is performed, but these packets can be stored in secondary memory. For example, for storage of 1 GB, with a packet size of 1KB and packet rate of 1000, a packet will be available for over 1000 seconds.

6.3 Communication Overhead: The communication overhead comprises of keys and tags attached. Communication overhead is low as secret vectors are of very small size. In *tag generation* communication overhead is reduced due to the small size of tags. The communication involves tags which are appended to each data packet for transmission and the key pair distribution to all the nodes. And this type of communication does not involve much overhead. The size of 160-bit ECDSA signature is normally 320 bits which for an unsigned coded packet is 2.7% and 5.4% for a signed coded packet of total bandwidth.

7. PERFORMANCE EVALUATION

7.1 Corrupt Packet Identification

A packet is considered to be corrupted if it contains invalid signature as all nodes perform verification for each received packet. And if a packet is found polluted it is dropped immediately at the first neighbor of attacker and the link is also identified as polluted.

7.2 Traceback Procedure Invocation

Whenever a polluted packet is transmitted with a correct signature by a pollution attacker and results in incorrect decoding at the receiving end, a notification is sent by the reporting node to the source/sending node which invokes traceback procedure.

7.3 Attacker Node Identification

With each traceback invocation one attacker node is identified as it performs breadth-first search to check if any intermediate node is polluted and eventually reaches at destination node and checks if any inconsistent data is received.

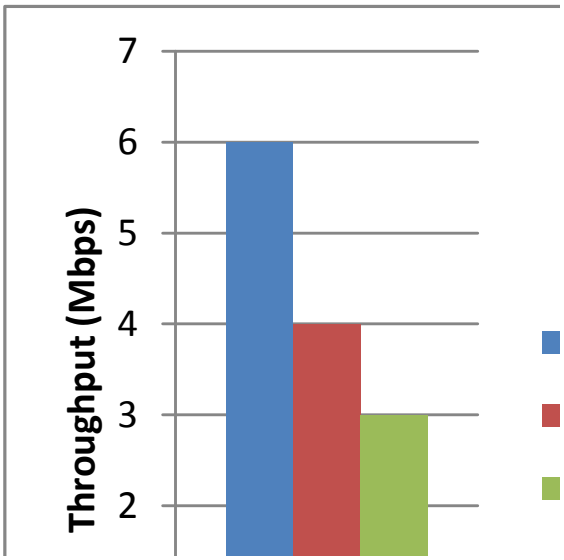

Chart -1: Throughput comparison

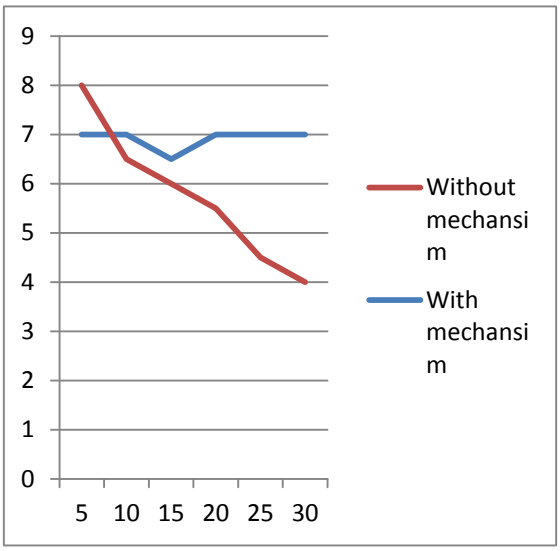

Chart -2: Throughput with & without mechanism

8. CONCLUSION

The pollution attacks are a very critical problem which leads to the wastage of network resources and it affects the performance of network. The solution to this problem was to append a code or tag which ensures the integrity of the data packets during transmission but then malicious nodes started polluting these tags which became a major problem as if the tag is polluted but the message is intact, it was presented as forged packet so it will be discarded which affected the transmission efficiency highly. To overcome these problems a new scheme was proposed which is used to provide the security against pollution and tag pollution attacks and also help in identifying at least one malicious node in the network. With added security, this scheme includes fault tolerance capability.

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 2000.

[2] Rouse, Margaret. 'What Is Network Coding? - Definition From Whatis.Com'. 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] Frédérique Oggier and Hanane Fathi An Authentication Code Against Pollution Attacks in Network Coding, IEEE/ACM Transaction on Networking 2011.

[6] C. Gkantsidis and P. Rodriguez, Cooperative security for network coding file distribution, in Proc. IEEE INFOCOM, 2006, pp. 1–13.

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

[9] 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.

[10] Mangesh N. Bhandare, Ravindra C. Thool, Gurdeep Singh Wahi, Avoiding Pollution Attacks in Network Coding using Authentication Code, IJCSMC, Vol. 2, Issue. 6, June 2013, pg.239 – 243.

[11] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, An efficient signature based scheme for securing network coding against pollution attacks, in Proc. IEEE INFOCOM, 2008, pp. 1409–1417.

[12] D. Boneh, D. Freeman, J. Katz, and B. Waters, Signing a Linear Subspace: Signature Schemes for Network Coding. Springer, Mar. 2009.

[13] Mayank Kumar Goyal, Satya Prakash Ghrera, Jai Prakash Gupta, Network Tomography Integrated Probe Tested Network Coding in Wireless Networks, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-5, January 2019.

[14] Khanzada, Mukhtiar, Naz Bushra, Talpur, Faisal, Evaluation and Analysis of Network Coding at Network Layer, IEEE,2017.

[15] Jing Dong, Reza Curtmola, Member, IEEE, Cristina Nita-Rotaru, Senior Member, IEEE, and David K. Y. Yau, Member, IEEE, Pollution Attacks and Defenses in Wireless Inter-flow Network Coding Systems.

[16] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Defenses against pollution attacks in wireless network coding," ACM Trans. Inf. Syst. Secur., vol. 14, pp. 7:1–7:31, June 2011.

[17] Vysagh M, Reeja R Rajan, Ambikadevi Amma T, Securing Network against Pollution Attack using Tagging Scheme, IJIRST – International Journal for Innovative Research in Science & Technology| Volume 1 | Issue 11 | April 2015 ISSN (online): 2349-6010.

[18] Mangesh N. Bhandare, Ravindra C. Thool, Gurdeep Singh Wahi, Avoiding Pollution Attacks in Network Coding using Authentication Code, IJCSMC, Vol. 2, Issue. 6, June 2013, pg.239 – 243.

[19] Neha V. Mamidwar, Ms. Deepali Gothawal, "Schemes against Pollution Attack in Network Coding: A Survey", International Journal of Computer Science and Information Technologies, Vol. 6 (6) , 2015, 5085-5089.