An Efficient Dynamic Authentication Scheme for Secure Network Coding

Ming He*, Hong Wang, Lin Chen, Zhenghu Gong, Fan Dai
College of Computer, National University of Defense Technology
Kaifu District, Changsha, China. tlp/fax +86 0731 84572062
*Corresponding author, e-mail: hemingclear@nudt.edu.cn

Abstract

Network coding based applications are notoriously susceptible to malicious pollution attacks. Packets authentication schemes have been well-recognized as the most effective approach to address this security threat. However, existing packets authentication schemes for network coding either incur high computation overhead, or are vulnerable to arbitrary collusion among attackers. In this paper, we proposed a novel dynamic authentication scheme for secure network coding by dynamically using the linear vector subspaces of network coding. The scheme can efficiently detect packets generated by pollution attackers, and effectively resist arbitrary collusion among attackers. Our work is an innovative dynamic solution to frustrate pollution attacks with arbitrary collusion among attackers, and authentication cost can be further reduced by employing batch authentication in participating nodes. In addition, our scheme provides an efficient packet authentication without requiring the existence of any extra secure channel. Security analysis and simulations results demonstrate the practicality and efficiency of our scheme.

Keywords: *network coding, pollution attack, dynamic, security, collusion*

1. Introduction

Network coding, which was first introduced by Ahlswede et al [1], is an efficient means of information dissemination. Unlike the traditional transmission mechanism which requires simply duplicating every input packet, network coding allows intermediate nodes to mix the information content in packets before forwarding them. This mixture of information has been theoretically proven to maximize network transmission capacity [2]. The source node splits the original data file into data blocks and then encodes them with appropriate coding methods, including exclusive or (XOR) operation, linear operation and so on. Intermediate nodes can also perform coding operations on linearly independent incoming packets, in addition to just forwarding them. Receivers are able to decode the original data file once they have received sufficient linearly independent coded blocks. With lots of benefits for communication network, network coding has emerged as a promising technique in many practical network applications, such as wireless sensor networks [3], mobile communication networks [4] and peer-to-peer content distribution networks [5].

However, network coding may face potential security threats due to the packet encoding at intermediate nodes. The nature of packet mixing subjects network coding systems to a severe security threat, known as pollution attack, where attackers inject corrupted packets into the network. In pollution attacks, if a single corrupted block is mixed by a intermediate node, all output packets of the intermediate node will be contaminated, and then corrupted packets may spread to all downstream nodes. As a result, a single corrupted block will pollute the whole network system and prevent receivers from decoding. Therefore, it is crucial to introduce corresponding security schemes to detect and filter corrupted packets as early as possible.

In this paper, we propose D-Auth scheme, an efficient solution for network coding against pollution attacks along with arbitrary collusion among malicious nodes. Our work is on the basis of some core technologies [6, 7]. D-Auth scheme differs from previous schemes in two main respects: (1) tolerating arbitrary collusion among malicious nodes, and (2) using the linear vector subspaces of network coding to achieve computationally efficient and strictly secure. We make two main contributions. First, we construct a dynamic authentication scheme that enables forwarders to verify the authentication of incoming blocks even with the existence of arbitrary

Received January 12, 2013; Revised March 8, 2013; Accepted March 16, 2013

collusion, and it also limits pollution attacks by preventing the propagation of corrupted blocks to downstream nodes. Second, we analysis the security of D-Auth scheme and validate the performance and overhead of our scheme with corresponding simulations.

The remainder of the paper is structured as follows. Section 2 overviews related work. Section 3 introduces our system model and threat model. Section 4 provides the detail of D-Auth scheme, while Section 5 shows simulations and analysis of D-Auth scheme. Finally, Section 6 concludes the paper.

2. Related Work

So far, a number of schemes have been proposed to defend against pollution attacks in network coding systems. These schemes can be categorized into two classes: end-to-end scheme and in-network scheme.

2.1. End-To-End Scheme

End-to-end schemes filter polluted messages only at receivers. End-to-end schemes make minimal changes to existing network coding algorithms, and only the source and receivers are involved in performing polluted messages detection [8].

Silva D et al. [9] proposed rank-metric codes to protect source message from malicious nodes. However, rank-metric codes can not resist large amount of pollution attacks. Koetter et al [10] improve network error correction coding theory for detecting and correcting corrupted packets. However, In a wireless environment where pollution attacks occur, attackers can easily overwhelm the error correcting ability and fail the decoding of receivers.

Vilela J P et al. [11] proposed a security framework, which exploits the intrinsic security of network coding. However, to achieve confidentiality, it requires either a large enough field from which coding coefficients are chosen or secure GEVs which have many columns. Both of them will result excessively high computation overhead.

Jaggi et al [12] provided a polynomial-time network coding algorithm to allow receivers to recover native packets in the presence of pollution attacks. However, if the attacker has a large bandwidth to the receiver, the useful throughput can easily degrade to zero.

In end-to-end schemes, when attackers locate their attacks at the bottleneck of the network, end-to-end schemes can lead to a worst-case view of malicious attack. Furthermore, there is a huge benefit in detecting the presence of the attacker and forbidding polluted packets to propagate in the network, due to the constrained bandwidth of the medium.

2.2. In-Network Scheme

In-network schemes filter polluted messages at all intermediate nodes. Compared to end-to-end schemes, in-network solutions can make corrupted packets dropped earlier.

Krohn et al [13] proposed a homomorphic hashing scheme that allows intermediate nodes to perform integrity verification of blocks exchanged in the network. Since each intermediate node needs to verify all incoming blocks before combining them, the performance of the network would be limited by the performing rate of its homomorphic hashes.

Kehdi et al [14] proposed null keys to allow forwarders to detect corrupted blocks using null space properties of network coding. However, null key become vulnerable under collusion attack, where multiple malicious nodes are allowed to exchange their attack messages. Zhen Yu et al. [15] proposed a scheme for securing XOR network coding. It can not resist modified packet pollution that attacker sends fake source messages by computing corresponding MACs using its own random keys. Jing Dong et al. [16] used time-based authentication to resist pollution attacks. However, because each node has to buffer incoming unverified packets before the receipt of the next checksum, malicious nodes can easily conduct a DDos attack by overflowing the unverified buffer during this period of time.

In-network solutions use cryptographic approaches by which forwarders detect and drop polluted packets. However, in-network schemes are various from each other in efficiency and security, and there is no practical in-network schemes so far. Previous solutions to the problem will be worse when facing arbitrary collusion among malicious nodes.

3. System and Threat Model

In this section, we briefly introduce the system model and the threat model, which are the foundation of D-Auth scheme.

3.1. System Model

We model the network by an acyclic directed graph $Gr=(V,E)$, where V is the set of nodes, and E is the set of edges with unit capacity. Edges are denoted as $e=(v,v')\in E$, where $v=head(e)$ and $v'=tail(e)$. The set of edges that originating at $v\in V$ is defined as $\Gamma O(v)=\{e\in E: head(e)=v\}$ while the set of edges end at v is defined as $\Gamma I(v)=\{e\in E: tail(e)=v\}$. Assume that each symbol is an element of a finite field F_q, where q is a prime number. We consider a single source $S\in V$, a set of receivers $R\subset V, R=\{R_1,R_2,...,R_k\}$, and multiple forwarders for packets. Receivers may also act as forwarders. Assume that S attempts to send some information blocks to receivers in R, and S divides this sequence of blocks into generations. We focus on the coding and transmission of a single generation, and only messages from same generation are encoded. Each generation consists of n blocks, and each block $\vec{p}_i$ can be viewed as an element in an m-dimensional vector space over the field F_q:

$$\vec{p}_i=(p_{i1},p_{i2},...,p_{im}),\quad p_{ij}\in F_q, 1<j<m. \tag{1}$$

A generation G consisting of n packets can be viewed as a matrix, with each packet in the generation as a column in the matrix:

$$G=[\vec{p}_1,\vec{p}_2,...,\vec{p}_n]^T. \tag{2}$$

S forms random linear combinations of native blocks $\vec{e}=\vec{c}G=\sum_{i=1}^{n}c_i\vec{p}_i$, where $\vec{c}=(c_1,c_2,\cdots,c_n)$ and $c_i(1\le i\le n)$ is a random element in F_q. S then forwards packets consisting of $(\vec{c},\vec{e})$ in the network. We refer to $(\vec{c},\vec{e})$ as coded packets, and $\vec{c}$ as the global encoding vector. A forwarder node also forms new coded packets by computing linear combinations of linearly independent coded packets it has received and forwards them to the network. When a receiver has obtained n linearly independent coded packets, it can decode them to recover native packets by solving a system of n linear equations. We denote blocks which S is about to disseminate to receivers as X, a $n\times(m+n)$ matrix whose i^{th} row is $\vec{x}_i$ $(1\le i\le n)$.

The dimension of orthogonal space of Π_X is equal to m [14]. We denote the orthogonal space of the matrix X as $\Pi_X^{\perp}$, which is the set of all vectors $\vec{u}$ matching $X\cdot\vec{u}^T=0$. $\Pi_X^{\perp}$ is spanned by basis vectors $U=\{\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_m\}$, and we denote nullity vectors U as a $m\times(m+n)$ matrix whose i^{th} row is $\vec{u}_i(1\le i\le m)$.

We assume that clocks in the network are loosely synchronized. Each node knows the upper bound on the clock difference between itself and the source, which is denoted as Δ.

3.2. Threat Model

We assume that the source is always trusted, while both forwarders and receivers may not be trusted since they may be either spurious nodes disguised by the attacker or legitimate but compromised nodes.

(1) Pollution attacks

Malicious nodes launch pollution attacks either by injecting spurious packets or by modifying their output packets to contain corrupted data. We say a packet $(\vec{c},\vec{e})$is a polluted packet if the vector $\vec{e}$ is not equal to the product of the original generation of packets $\{\vec{p}_1,\vec{p}_2,...,\vec{p}_n\}$ and the global encoding vector $\vec{c}$. i.e., the following in equation:

$$\vec{e} \neq \sum_{i=1}^{n} c_i \vec{p}_i \qquad (3)$$

(2) Collusion attacks

Collusion attack means that malicious nodes collude to cheat more payment from source node. The main goal of collusion attacks is to destroy the cryptographic function of security scheme in network coding system. They attempt to compromise as many intermediate nodes as possible to obtain the confidential encryption information. If a node is compromised, attackers can read its memory and monitor all incoming and outgoing communications. Attackers can also collaborate to launch collusion attacks, making the security scheme easy to be destroyed.

4. D-Auth scheme

Definition 1(collusion-resistant): A security scheme is said to be collusion-resistant if attack information exchange during adversaries has neglectable impact on packets verification.

In this paper, we propose D-Auth scheme, a collusion-resistant authentication scheme for secure network coding by dynamically using the linear vector subspaces of network coding.

4.1. Scheme Description

We present our scheme incrementally. Firstly, we describe the parameter initialization phase of our scheme. We then show the source setup phase and packet authentication phase.

(1) Parameter initialization

In this phase, we present initialization of some parameters. We denote blocks to be distributed as X, a $n \times (m+n)$ matrix whose i^{th} row is $\vec{x}_i$ $(1 \le i \le n)$ in the finite field F_q.

ϕ: ϕ is a large prime number, and q is a divisor of $\phi - 1$. The technique which applied in Digital Signature Algorithm (DSA) can be used to to find such ϕ.

F_ϕ: F_ϕ is a finite field.

F_ϕ^*: F_ϕ^* is the multiplicative group of field F_ϕ, and the order of F_ϕ^* is $\phi - 1$.

Ω: Ω is a subgroup with order q in F_ϕ^*. Since the order of F_ϕ^* is $\phi - 1$, which is a multiple of q, we can always find such Ω.

g: g is a generator of Ω. i.e., $g \in \Omega$ and $\Omega = \{g^j, j \in Z\}$.

(2) Source setup

The source need not change the way to distribute native data packets. In our scheme, security against pollution attacks is based on a set of verification vectors dynamically generated by the source. Those verification vectors are referred to as D-Auth vectors, and they enable participating nodes to verify incoming packets. The source protects the authenticity of D-Auth vectors by digitally signature scheme [17]. The detail of source setup is as follows.

The source use Gaussian elimination to find a vector $\vec{u}$ matching $X \cdot \vec{u}^T = 0$. We denote that $\vec{u} = \{u_1, u_2, \ldots, u_{m+n}\}$. The source generates a set of nonzero random elements $B = \{b_1, b_2, \ldots, b_{m+n}\}$. Multiplicative inverse of B is $B^{-1} = \{b_1^{-1}, b_2^{-1}, \ldots, b_{m+n}^{-1}\}$,$b_i \cdot b_i^{-1} = 1 (1 \le i \le m+n)$. Both B and B^{-1} are only kept by the source.

The source computes validation vector $A = \{a_i = g^{b_i}\}$ $(1 \le i \le m+n)$, and new orthogonal vector, $D = \{d_i = u_i \cdot b_i^{-1}\}$ $(1 \le i \le m+n)$. The source forms a D-Auth vector $K_{D-Auth} = (A, D, c\_time)$, and c_time is the timestamp at the source when K_{D-Auth} is created. The source signs K_{D-Auth} with digitally signature scheme and broadcasts K_{D-Auth}.

(3) Packet verification

Each participating node maintains two packet buffers, verified-buffer and unverified-buffer, which buffer verified and unverified packets, respectively. Each node combines only packets in the verified-buffer to form new packets and forwards such packets. On receiving a new coded packet, a node buffers the packet into unverified-buffer and records the corresponding received time. We denote the data form of both verified-buffer and unverified-buffer as $(buf\_flag, rec\_time, data)$.

Once receiving a D-Auth vector K_{D-Auth}, a participating node first verifies that K_{D-Auth} is signed by the source. If K_{D-Auth} is authentic, the node publishes it to its neighbors.

The node uses $K_{D-Auth} = (A, D, c\_time)$ to verify those packets in unverified-buffer that were received before K_{D-Auth} was created at the source. i.e., $rec\_time \le c\_time - \Delta$. Δ is the maximum clock difference between the node and the source. The node verifies the integrity of corresponding data block ψ by checking if it holds the following equation:

$$\prod_{i=1}^{m+n} a_i^{d_i \cdot \psi_i} = 1 \tag{4}$$

This is simple computation that allows participating nodes to rapidly perform the packet authentication. And equation (4) holds for any valid ψ, we have:

$$\begin{aligned}\prod_{i=1}^{m+n} a_i^{d_i \cdot \psi_i} &= \prod_{i=1}^{m+n} \left(g^{b_i}\right)^{\left(u_i \cdot b_i^{-1}\right) \cdot \psi_i} \\ &= \prod_{i=1}^{m+n} g^{u_i \cdot \psi_i} \\ &= g^{\sum_{i=1}^{m+n} (u_i \cdot \psi_i)} \\ &= g^0 \\ &= 1\end{aligned}$$

Valid data blocks are transferred from unverified-buffer to verified-buffer. Data blocks that do not pass the verification are discarded. D-Auth vectors are not required to be delivered reliably. If a node fails to receive a current version D-Auth vector, it can verify its buffered packets upon the receipt of the next version D-Auth vector. When a receiver receives enough linear independent coded packets that have passed the packet verification, it decodes these coded packets to recover native packets. It verifies native packets using an end-to-end authentication scheme such as traditional digital signature or message authentication code before passing packets to the upper layer protocol. The additional end-to-end authentication is used for addressing the extremely rare occasion when some polluted packet pass our verification at the receiver, which would otherwise cause incorrect packets to be delivered to the upper layer.

4.2. Batch Authentication

The above individual packet authentication can be extended to efficiently batch authentication which verify a set of coded packets at one time.

We assume that $E = \{(\vec{c}_1, \vec{e}_1), \ldots, (\vec{c}_l, \vec{e}_l)\}$ be a set of unverified coded packets that are received before current version D-Auth vector was created in node v. v computes a random linear combination of E. i.e., $(\vec{c}, \vec{e}) = \left(\sum_{i=1}^{l} \lambda_i \vec{c}_i, \sum_{i=1}^{l} \lambda_i \vec{e}_i\right)$, where $\lambda_i (1 \le i \le l)$ is a random element in F_q. v verifies the combined packet $\psi = (\vec{c}, \vec{e})$ with the individual packet authentication described above. The false negative probability of the verification can be further reduced by repeating this batch authentication with different random coefficients.

If ψ passes the batch verification, then all coded packets in E can be regarded as valid. Otherwise, the invalid packets in E can be located efficiently using a technique similar to binary search.

5. Results and Analysis

In this section, the results of our scheme is explained, and at the same time comprehensive discussion is given.

Theorem 1: D-Auth scheme is collusion-resistant.

Proof: In collusion attack, attackers collaborate to obtain more D-Auth vectors from source node. Assume that attackers know all the D-Auth vectors which source node has disseminated in the network, and attackers are able to generate corrupted packets that match its known D-Auth vectors. However, attackers cannot convince other nodes to accept those corrupted packets, as they will not be verified with D-Auth vectors known to the attackers, but with another new version D-Auth vector dynamically generated by the source at a time after those packets are received. For the timeliness of D-Auth vectors, it is very hard for malicious nodes to know the content of its neighbor, and in that case, it is hard to find a corrupted block that can pass the verification process. With the path diversity and the timeliness of D-Auth vectors, the attacker can only randomly guess the upcoming new version D-Auth vectors in intermediate nodes, thus only having a small chance of success, and D-Auth scheme is collusion-resistant.

We compare our scheme with cooperative security scheme introduced by Gkantsidis et al [18] and null keys scheme introduced by Kehdi et al [14]. We use the percentage of corrupted nodes as a metric to measure the efficiency of security schemes with network coding. In the simulations, the network consists of 1000 nodes, and the block size is set to 128. The topology is a directed random graph with one source node. Each pair of nodes is connected with a probability p. The simulator is round-based, where in each round a node can upload and download blocks. In each round, the malicious node send one polluted message on each of their outgoing links. We randomly choose malicious nodes from the population, and all the results are averaged over several runs.

Figure 1 shows the pollution spread without D-Auth scheme. In all cases of probability p, the increase of malicious nodes percentage will definitely cause the raise of corrupted nodes percentage, and when the number of malicious nodes grows, participating nodes become more susceptible to pollution attacks. As the probability p increases, the effect of malicious nodes becomes stronger. That is because malicious nodes can easily inject a greater number of bogus blocks to participating nodes since the connection with malicious nodes grows. We can see that the security situation is terrible when facing multiple attackers, and it will be worse when facing arbitrary collusion among multiple attackers.

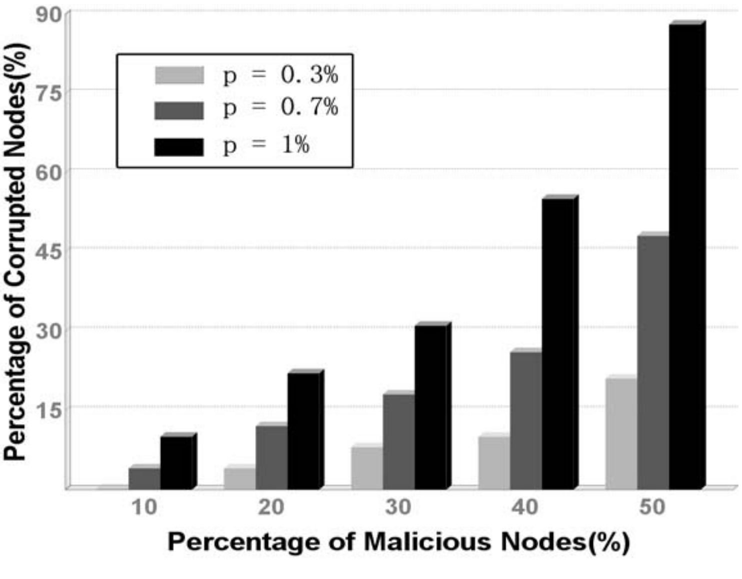

Figure 1. Pollution spread without D-Auth scheme

Figure 2 shows that our scheme succeeds in improving the protection against pollution attacks. We should notice that the corruption decreases as we increase the checking probability in cooperative security, but the network performance degrades due to higher computational complexity. A checking probability of 30% will impose a significant computation overhead in cooperative security. In homomorphic hashing model, a node stops using unsecured blocks when an alert message is received. This is another drawback that decreases the network performance, since non-corrupted blocks can be part of these unsecured blocks. In null keys scheme, we mentioned that increasing the percentage of malicious nodes obviously expands the percentage of corrupted nodes, which shows that null keys scheme become vulnerable when facing multiple attackers. In our scheme, we see that across different percentages of malicious nodes, the percentage of corrupted nodes only increases slightly compared to other solutions. Our scheme can effectively defend against pollution attack with arbitrary collusion among multiple attackers.

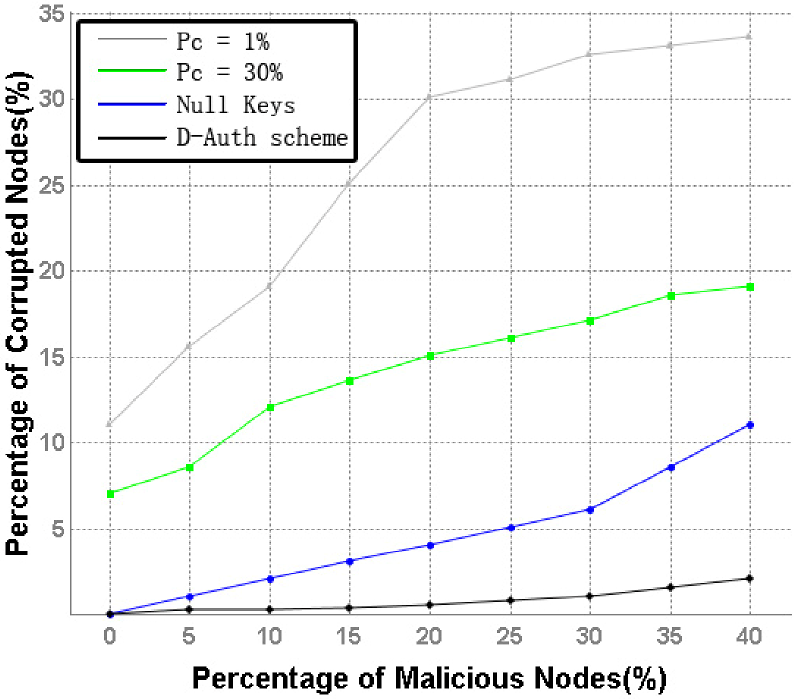

Figure 2. Comparison between our scheme and previous solutions.

6. Conclusion

In this paper, we present a collusion-resistant authentication scheme for secure network coding by dynamically using the linear vector subspaces of network coding. Our scheme nicely makes use of the linearity property of random linear network coding, and enables participating nodes to check the integrity of packets without the requirement for a secure channel. Packets authentication is based on D-Auth vectors dynamically generated by the source, and the security of our scheme is based on the hardness of the discrete logarithm problem. The protection against pollution attacks with collusion is based on the timeliness of D-Auth vectors. D-Auth vectors are signed using digitally signature scheme and the impact of signature can be optimized by adjust the time interval of releasing verification packets.

Our work is an innovative dynamic solution to frustrate pollution attacks with arbitrary collusion among attackers, and authentication cost can be further reduced by employing batch authentication in participating nodes. Our simulations show that our scheme effectively limits the pollution and isolates malicious nodes. The stable corruption spread, helps in identifying the locations of the malicious nodes even facing pollution attacks with collusion.

Acknowledgments

This work is supported by Program for Changjiang Scholars and Innovative Research Team in University(No.IRT1012), "Network technology" Aid program for Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province, Hunan Provincial Natural Science Foundation of China(11JJ7003), "Reconfigurable Network Emulation Testbed for Basic Network Communication" Program for National Basic Research Program of China (973 Program).

References

[1] R Ahlswede, N Cai, SR Li, RW Yeung. Network Information Flow. *IEEE Transactions on Information Theory*. 2000; 46(4): 1204-1216.
[2] SR Li, RW Yeung, N Cai. Linear network coding. *IEEE Transactions on Information Theory*. 2003; 49(2): 371-381.
[3] HY Shwe, F Adachi. *Power efficient adaptive network coding in wireless sensor networks*. Proceedings of IEEE Int Communications (ICC) Conf. 2011; 1–5.
[4] H Wu, J Zheng. Efficient network coding-based multicast retransmission mechanism for mobile communication networks. *IET Communications*. 2012; 6(2): 187-193.
[5] Jinbiao Xu, Xin Wang, Jin Zhao, Azman Lim. Iswifter: Improving chunked network coding for peer-to-peer content distribution. *Peer-to-Peer Networking and Applications*. 2012; 5: 30-39.
[6] Ali M Fadhil, Haider M Al Sabbagh. Performance Analysis for Bit Error Rate of DS CDMA Sensor Network Systems with Source Coding. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 1(1): 165-170.
[7] H Omranpour, M Ebadzadeh, S Shiry, S Barzegar. Dynamic Particle Swarm Optimization for Multimodal Function. *IAES International Journal of Artificial Intelligence (IJ-AI)*. 2012; 1(1): 1-10.
[8] Yaping Li, Hongyi Yao, Minghua Chen, Sidharth Jaggi, Alon Rosen. RIPPLE Authentication for Network Coding. *Infocom*. 2010; 2258-2266.
[9] Silva D, Kschischang FR. Universal weakly secure network coding. *IEEE Transactions on Information Theory*. 2009; 281-285.
[10] R Koetter, FR Kschischang. Coding for errors and erasures in random network coding. *IEEE Transactions on Information Theory*. 2008; 54: 3579-3591.
[11] Vilela JP, Lima L, Barros J. Lightweight Security for Network Coding. *IEEE International Conference on Communications*. 2008; 1750-1754.
[12] S Jaggi, M Langberg, S Katti, T Ho, D Katabi, M Medard. Resilient network coding in the presence of byzantine adversaries. *Infocom*. 2007; 616-624.
[13] MN Krohn, MJ Freedman, D Mazieres. On-the-fly Verification of Rateless Erasure Codes for Efficient Content Distribution. *IEEE Symposium on Security and Privacy*. 2004; 226-240.
[14] Elias Kehdi, Baochun Li. Null Keys: Limiting Malicious Attacks Via Null Space Properties of Network Coding. *Infocom*. 2009; 1224-1232.
[15] Zhen Yu, Yawen Wei, Bhuvaneswari Ramkumar, Yong Guan. Null Keys: An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks. *Infocom*. 2009; 406-414.
[16] Jing Dong, Reza Curtmola, Cristina Nita-Rotaru. Practical Defenses against Pollution Attacks in Intra-Flow Network Coding for Wireless Mesh Networks. *The Second ACM Conference on Wireless Network Security*. 2009; 111-122.
[17] F Zhao, T Kalker, M Medard, KJ Han. Signatures for content distribution with network coding. *IEEE Int.Symp.Information Theory*. 2007; 556-560.
[18] C Gkantsidis, P Rodriguez Rodriguez. Cooperative security for network coding file distribution. *Infocom*. 2006; 1-13.