International Journal of Security and Its Applications
Vol. 10, No. 3 (2016), pp.161-172
http://dx.doi.org/10.14257/ijsia.2016.10.3.15

A Light Weight Security Scheme for Network Coding Based on a Mobius Transformation

M. H. Tadayon

Iran Telecommunications Research Center (ITRC), Tehran, Iran.
tadayon@itrc.ac.ir

Abstract

Network coding is a new method for forwarding network throughput in digital communication systems. In this paper, we introduce an efficient scheme for protecting the source data against wiretapper in linear network coding. The proposed scheme is implemented on the generated data packets in source node. We employ a well-known permutation function called Mobius transformation to transfer the existing data packet to an appropriate interchanged data packet. Then the new data packet can be sent to the intermediate nodes via output links of the source node in network securely. Indeed, the security of the proposed scheme against wiretapper is provided by employing Mobius transformation and interleaver operation on the generated data packet in source node. In the absence of cryptography systems, the proposed method is a light weight security scheme for network coding that can provide a security level easily.

Keywords*: Network Coding, Mobius transformation, Interleaver operation, light weight security, wiretapper*

1. Introduction

Network Coding has introduced as a new message forwarding technique that allows one to combining multiple input-packets together to form an output-packet. Network Coding in comparison with conventional communication method – routing – has more potential advantages, however, the nature of packet-combining makes the communication system highly sensitive to error propagation and wiretapper. Due to complexity of data transmitting and processing in Network coding graph, security and privacy are big challenges that have not had acceptable solutions yet [1], [2].

Indeed, the nature of packet-combining in Network Coding is the feature which allows each of the intermediate nodes combining algebraically the incoming data packets and making a new data packet to forward to next nodes. However, in the conventional communication network such as Internet, data packet delivery is performed by store-and-forward or routing. Data packets received from an input links of an intermediate node are stored and their copies are forwarded to the next node via an output link. In the case where an intermediate node is on the transmission paths toward multiple destinations, it sends one copy of the data packets into each output link that leads to at least one of the destinations [3]. Thus, Network coding is generalized routing model by assuming that intermediate nodes (routers) are allowed to modify the packets before forwarding them. Moreover, we expect Internet of Things (IOT) uses Network Coding concepts in data passing, so it needs the new schemes for its security challenges, i.e. security, privacy and trust.

In this paper, we employ permutation function and interleaver operation for presenting a good low cost scheme to protect the generated data packet against wiretapper. The security of the proposed scheme depends on cardinality of the finite field that used by structural permutation polynomials and interlevers.

ISSN: 1738-9976 IJSIA

The main portion of secure network coding is devoted to the problem of controlling and protecting it against the malicious eavesdroppers. The problem is how to prevent information from leaking to adversaries. Traditional approaches require end-to-end or hop-by- hop encryption. However, for the end-to-end approach, if a large amount of the encrypted information is obtained by adversaries, they will conduct known-ciphertext attack to learn the real contents of the messages. Although the confidential data is transmitted as the ciphertexts by using cryptosystems, but in general case cryptosystems cannot be always applied to the public digital communications because of the cost, complexity and the delay created by them in [1]. So in some cases, we would like to have a fast technique for security or privacy preserving in a short time. We present a lightweight cryptographic scheme to ensure confidentiality in network coding, which leverages the inherent security to reduce the overhead in comparison to end-to-end encryption of the entire data flow.

A network is secure if a wiretapper, who may access to any subset of the wiretap channels in network, can obtain no information about the data packets. The wiretappers can be just as the intermediate nodes; all destination nodes in the network are as legal users which can decode the secure data packets with zero error. There are several alternative models for secure network coding. Among them, Jain in [4] focused on the relation between security and network topology and the trade-off between security and the cost of network coding was independently studied by Tan and Médard in [5].

It is expected that the Network Coding as a method of digital communication network has the least initial security against the wiretappers, hence, we have proposed a new scheme that provide the initial security without complex computations and with the low cost and delay. Indeed, in our proposed scheme, the initial security is implemented by the functions which have mathematical structure e.g. permutation functions (polynomials) and interleaver operations.

The rest of the paper is organized as follows: in the next section, we provide necessary background definitions and notations related to Network Coding. Section 3 is devoted to the definition of permutation functions and interleaver operations. A scheme is given in section 4 and the conclusion is presented in section 5.

2. Background on Network Coding

A simple communication network is often described by a finite directed graph G = (V, E), where V is the set of nodes and E the set of edges which represent the point-to-point channels. In Network Coding, we assume that every edge is a discrete memory less and noiseless channel with the capacity of one data unit per unit time and consider G is acyclic, i.e., it does not contain a directed cycle.

2.1. Notations and Definitions

Network Coding consists of three layers (the first layer includes source node(s), where data packet is generated in it, the second layer includes intermediate nodes, and the third layer includes destination node(s)). The network used in this paper is single-source multicasting. In such network, there is a single source node, which produces a data packet, and some multiple destination nodes. In the multicast case, let n denote the minimum number of min-cuts from source node to all destination nodes [6]. Indeed, n is the multicast capacity of the network or maximum number of edge-disjoint Steiner trees from source to destination nodes.

Ahlswede et al. in [7] were proposed Network Coding as a way to increase the average rate of data packet dissemination from a source node to multiple destination nodes. Also Li et al. in [8] and Koetter and Médard in [9] proved that linear Network Coding is sufficient to achieve the multicast capacity. Linear Network Coding means that the data

packets can be considered symbols of elements from a finite field, and the functions performed at the nodes can be simple linear combinations over this finite field. Furthermore, all decoding at the receivers can be performed using linear operations. Jaggi et al. in [10] provided a polynomial time algorithm for finding encoding and decoding coefficients in directed acyclic network. Then Erez et al. extended this approach to directed networks with cycles [11].

Let M to be an arbitrary set of alphabet with cardinality m that is entered by imaginary channels to source node, F_q be a finite field with $q \geq 2$ elements; n be the number of data packets, $F_q^{n \times m}$ denote the set of all $n \times m$ matrices over F_q. Then, source node generates data packets X_i for $1 \leq i \leq n$ from the messages (i.e. X_i is a vector of m symbols over a finite field F_q, or $X_i \in F_{q^m}$). In other words, the data packets which are transmitted by the source node are the rows of a matrix $X = [X_1, X_2, ..., X_n]^T$. The matrix X is an element of the finite field $F_{q^m}^n$. Because there is a bijection function that maps every $X \in F_{q^m}^n$ to $X \in F_q^{n \times m}$, X is assumed to be an element of the finite field $F_q^{n \times m}$. For the sake of uniformity of notation, let the symbols on outgoing edges of source node be equal to $X_1, X_2, ..., X_n$. Then, it is clear that the packet $Y(e)$ on any edge $e \in E$ in the network can be computed as a linear combination of the source data packets $X_1, X_2, ..., X_n$, namely, $Y(e) = \sum_{i=1}^{n} g_i(e) X_i$. The coefficients of this linear combination form a vector $g(e) = [g_1(e), ..., g_n(e)]$ where $g_i(e) \in F_q$, known as the global encoding vector [12], [13] on edge e (Fig. 1).

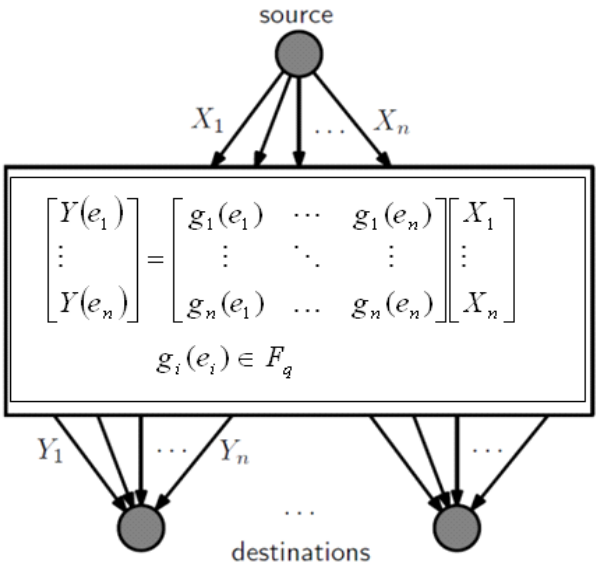

Figure 1. Linear Network Coding [13]

The global encoding vector represents the packet $Y(e)$ in terms of the source data packets $X_1, X_2, ..., X_n$. It is easy to see that the global encoding vectors themselves can be computed recursively as $\mathbf{g}(e) = \sum_{e'} k_{e'}(e) \mathbf{g}(e')$ using the coefficients of the local

encoding vectors $k(e)=[k_{e'}(e)]$ [12]. Suppose that a destination node t receives data packets $\{Y(e_1),...,Y(e_n)\}$ on edges $e_1,...,e_n$. The received packets can be expressed in terms of the source data packets in (1):

$$\begin{bmatrix} Y(e_1) \\ \vdots \\ Y(e_n) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_1(e_n) \\ \vdots & \ddots & \vdots \\ g_n(e_1) & \cdots & g_n(e_n) \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = G_t X, \quad (1)$$

where the i^{th} row of the matrix G_t is the global encoding vector associated with the edge e_i entering the destination t. If the matrix G_t is an invertible matrix then the destination node t can recover all n source data packets by multiplying the inverse of G_t by received packets.

2.2 Wiretapper

We now consider an acyclic multicast network G = (V, E) with unit capacity edges and the value of the min-cut to each destination node is equal to *n*. The goal is to send maximum data packets with the constraint of revealing no information about the data packets to the adversary (eavesdropper) that can access data packets on any μ edges.

We assume that the adversary knows the implemented network code, i.e. all the coefficients of the linear combinations that determine the data packets on each edge. Moreover, we assume that there is no shared randomness between the source node and destination nodes. In general case, in wiretapping attack the adversaries are able to wiretap or eavesdrop on a subset of the edges in a network coding system and gain access to the information transmitted on these edges.

Cai et al. in [14] proposed a model, called the wiretap network that incorporates information security with Network Coding. Their model includes secret sharing in classical cryptography as a special case. They presented a construction of secure linear network codes provided that a certain graph-theoretic condition is satisfied. In their model, the message M is randomly chosen from F_q^{n-r} (not necessarily uniformly distributed), while the independent random key K is uniformly chosen from F_q^r. Then X= (M, K) is sent from source node to other nodes and they showed how n, r and the linear network code can be chosen to make the code admissible, i.e., decodable and secure.

3. Permutation and Interleaver

3.1 Permutation Function

A permutation of a set is a bijective function from the set onto itself. There are many permutation functions that some of them are introduced by Carlitz in [15]. In this section, we introduce a well-known permutation functions such as Mobius nonlinear transformation. This type of permutation polynomial motivates us to investigate its application in Network Coding. The complexity of this permutation polynomial is linear. Mobius transformation is defined as follows:

$$T(x)=\begin{cases} \dfrac{ax+b}{cx+d} & x\neq\dfrac{-d}{c} \\ \dfrac{a}{c} & x=\dfrac{-d}{c} \end{cases}, \tag{2}$$

where $x\in F_q,\ a,b,c,d\in F_q,\ c\neq 0$ and $ad-bc\neq 0$.

Its inverse is

$$T^{-1}(x)=\begin{cases} \dfrac{dx-b}{-cx+a} & x\neq\dfrac{a}{c} \\ -\dfrac{d}{c} & x=\dfrac{a}{c} \end{cases}. \tag{3}$$

Therefore, the Mobius function transforms every element to another element in F_q. In general permutation function operates on a vector over finite field F_q while every data packet generated by the source node is a matrix form. Hence, we need to transform every data matrix into a vector.

3.2 Interleaver Operation

An interleaver rearranges input data such that consecutive data are spaced apart. At the receiver end, the interleaved data is arranged back into the original sequence by the deinterleaver. In other words, interleaver can transform a matrix into a vector. There are several well-known interleaver operations for such transformation, such as helical interleaver, Even-Odd interleaver. In general case, such interleavers operate like a permutation function and transform a sequence of length n information bits or a data matrix of size $n\times m$ to a vector of length norm, respectively.

For example, consider the data matrix as follows then we implement the helical Interleaver on the data matrix.

$$X=\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}. \tag{4}$$

For helical interleaver, first draw the oblique lines or diagonal lines from left to right in Fig.2.

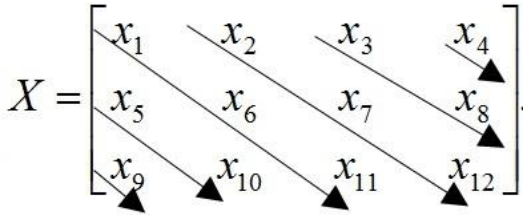

Figure 2. Helical Interleaver

Then, the entries on the oblique lines, from the left to right line, are put in the vector X'. Thus, the helical interleaver transform the matrix X to the vector X' as follows:

$$\boldsymbol{X}' = \begin{bmatrix} \boldsymbol{x}_9 & \boldsymbol{x}_5 & \boldsymbol{x}_{10} & \boldsymbol{x}_1 & \boldsymbol{x}_6 & \boldsymbol{x}_{11} & \boldsymbol{x}_2 & \boldsymbol{x}_7 & \boldsymbol{x}_{12} & \boldsymbol{x}_3 & \boldsymbol{x}_8 & \boldsymbol{x}_4 \end{bmatrix}. \quad (5)$$

In general case, an interleaver substitutes the n bits of sequence so that the size of the substituted sequence equals the size of the original sequence. In other words, let $L = \{0,1,...,n-1\}$ be all indices of the sequence, X, then the interleaver $\prod$ is defined as one-to-one function on to L.

$$\prod : L \to L \quad (6)$$

For every $i \in L$, there exists $j \in L$ such that: $\prod(i) = \ln \alpha^j = j$, where $\ln$ is the discrete logarithm of base α which is a primitive element of the finite field F_q and $\ln(0) = 0.$

Assume that α is a primitive element of F_q. Let T be a permutation function over F_q. Then, we define an interleaver $\prod_T : Z_4 \to Z_4$ by $\prod_T(i) = \ln\left(T(\alpha^i)\right)$.

We illustrate this definition in the following example.

Example 1: Mobius

Let $q = 2^m$, m= 2, $a = d = b = \alpha^2$, and $c = \alpha$ where α is a primitive element of the finite field F_q with $\alpha^2 + \alpha + 1 = 0$, Then we have

$$T(x) = \begin{cases} \dfrac{\alpha^2 x + \alpha^2}{\alpha x + \alpha^2} & x \neq \alpha \\ \alpha & x = \alpha \end{cases}, \quad \begin{aligned} &T(0) = \frac{\alpha^2}{\alpha^2} = 1 = \alpha^3, \quad T(\alpha^2) = \frac{1}{\alpha} = \alpha^{-1} = \alpha^2, \\ &T(\alpha) = \alpha, \qquad\qquad T(\alpha^3) = \frac{0}{1} = 0. \end{aligned}$$

It is clear that T is a permutation function over F_{2^2} and the Mobius transformation is defined as follows:

$$\prod_T : Z_4 \to Z_4$$
$$\prod_T(i) = \ln\left(T(\alpha^i)\right)$$
$$\prod_T(0) = \ln(T(0)) = 3, \quad \prod_T(2) = \ln(T(\alpha^2)) = 2,$$
$$\prod_T(1) = \ln(T(\alpha)) = 1, \quad \prod_T(3) = \ln(T(\alpha^3)) = 0.$$

For every $x \in F_{2^2}$ such that $x = \alpha^i, i \in Z_4$ we have $T(x) = T(\alpha^i) = \alpha^j, \prod_T(i) = j.$

In fact we have used x^2+x+1 as the irreducible polynomial of degree 2 over F_2. Let P be a permutation polynomial over $F_{q^m}=\{0,\alpha^1,...,\alpha^{q^m-2},\alpha^{q^m-1}\}$, where α is a primitive element of the finite field F_{q^m}, then for every *i*, $1\le i\le q^m-1$, there exists *j*, $1\le j\le q^m-1$, such that $P\left(\alpha^i\right)=\alpha^j$. It is clear that the inverse of the permutation function equals to $P^{-1}\left(\alpha^j\right)=\alpha^i$. In other words, function P operates as a rearranging function for powers of α. We want to implement permutation polynomial and interleaver operation over the elements of the data packets (matrix or vector) transmitted by the source node before they are entered to intermediate nodes. To do this, we show that data packets can be considered as a vector in $F_{q^m}^n$ and also we can consider it as a matrix in $F_q^{n\times m}$. In the next section, we show how the permutation polynomial (function) and interleaver operation can be implemented on source data matrix.

4. The Proposed Scheme

The source node generates data packet matrix X from matrix message M such that each of rows of matrix shown by X_i for $i\in\{1,...,n\}$ is sent out via every outgoing channel. By our scheme, the matrix X is permuted or interleaved and then every permutated rows (interleaved rows) of the matrix X, $X_i^{'}$ for $i\in\{1,...,n\}$, is sent to one of the outgoing channels (Fig. 3). Note that the destination nodes know the inverse of both permutation polynomial and interleaver operation.

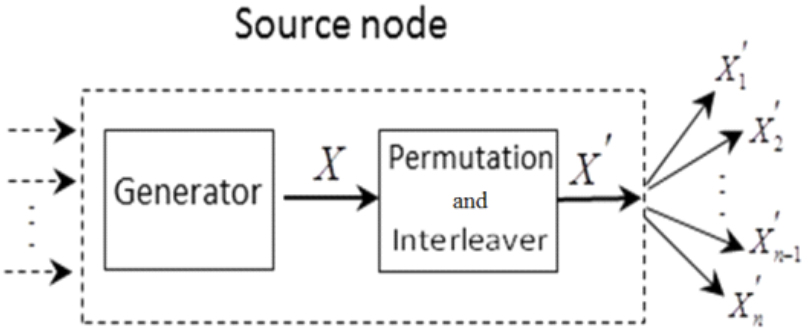

Figure 3. Proposed Scheme for Source Node

In Fig.3, input messages are transformed to data packet matrix X by generator and then matrix X is transformed to a new data packet matrix X' by the permutation function and interleaver operation. We illustrate the method in the following example.

Example 2: Let $q=2$, m= 3 and n=3 then there is a matrix data $X\in F_8^3=F_2^{3\times3}$ which must be sent to intermediate nodes. Suppose that $X=\begin{bmatrix}X_1\\X_2\\X_3\end{bmatrix}$ where every X_i for $i\in\{1,2,3\}$ is an element of the finite field F_{2^3} .It is obvious that every X_i can be

presented as a vector which has length 3, i.e. $X_i = \left(x_{i,1}, x_{i,2}, x_{i,3}\right)$. We use $\alpha^3 + \alpha + 1 = 0$, where α primitive element of the finite field is F_{2^3}. Then

$$F_8 = F_2^{1\times 3} = \{\alpha = 2 = (0,1,0),\ \alpha^2 = 4 = (0,0,1),\ \alpha^3 = \alpha + 1 = 3 = (1,1,0),\ \alpha^4 = \alpha^2 + \alpha = 6 = (0,1,1),$$
$$\alpha^5 = \alpha^2 + \alpha + 1 = 7 = (1,1,1),\ \alpha^6 = \alpha^2 + 1 = 5 = (1,0,1),\ \alpha^7 = 1 = (1,0,0)\}.$$

Assume that data matrix is:

$$X = \begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

When, we use permutation polynomial both coefficients of permutation polynomial and entries of the data matrix should belong to the same finite filed. These coefficients and entries of data matrix are powers of a primitive element.

To implement the permutation polynomial, we need to arrange the rows of the permuted matrix X in a row vector of length nm. The resulting permuted vector $\bar{X}$ is:

$$\bar{X} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

We define the Mobius transformation from Z_{nm} to Z_{nm}. Once an interleaver is used on the data vector the locations of the data vector entries (elements) will change and then the interleaved data vector is returned to its original matrix form.

Assume that the coefficients of Mobius transformation as follow:

$$a = \alpha^3,\ b = 1,\ c = \alpha^2,\ d = \alpha, \quad T(X_i) = \begin{cases} \dfrac{\alpha^3 X_i + 1}{\alpha^2 X_i + \alpha} & X_i \neq \alpha^{-1} = \alpha^6 = 5 \\ \alpha & X_i = \alpha^{-1} = \alpha^6 = 5. \end{cases}$$

If Mobius transformation coefficients are chosen differently for every X_i, then these coefficients must be induced as parity bits to each of $X_i's$. We have, $T(X_1) = \alpha^5 = 7$, $T(X_2) = \alpha = 2$, and $T(X_3) = \alpha^2 = 4$, for $i = \{1,2,3\}$.

Now, the new data matrix is:

$$X_{new} = \begin{bmatrix} \alpha^5 \\ \alpha \\ \alpha^2 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore, we can construct the data vector by putting every row of X_{new} in the vector $\bar{X}$ as follows:

$$\bar{X} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

If the Mobius transformation is used on $\bar{X}$, then we have the following results:

$$\prod_T : Z_{nm} \to Z_{nm}$$
$$\prod_T(i) = \ln\left(T(\alpha^i)\right)$$

$\prod_T(0) = \ln(T(\alpha^0)) = 1$, $\prod_T(1) = \ln(T(X_1 = \alpha)) = 5$, $\prod_T(2) = \ln(T(\alpha^2)) = 6$, $\prod_T(3) = \ln(T(\alpha^3)) = 3$, $\prod_T(4) = \ln(T(\alpha^4)) = 0$, $\prod_T(5) = \ln(T(\alpha^5)) = 0$, $\prod_T(6) = \ln(T(X_2 = \alpha^6)) = 1$, $\prod_T(7) = \ln(T(X_3 = \alpha^7)) = 2$, $\prod_T(8) = \ln(T(\alpha^8)) = 5$.

Now, by using inputs and outputs of the given Mobius transformation, we change $\bar{X}$ to X' as the following.

Let input and output elements of Z_{nm} in the Mobius transformation be as the corresponding position in the vector $\bar{X}$. Then by applying the Mobius interleaver, we substitute the input and output elements with each other (see Table 1and Fig. 4), e.g. if the number of input elements for the Mobius transformation is two then the second element of vector $\bar{X}$ is replaced by the sixth element of the vector $\bar{X}$ i.e. the second element is the one to be transmitted to the sixth position of the vector $\bar{X}$ and the sixth element is zero to be transmitted to the second position of the vector $\bar{X}$.

Table 1. Inputs and Outputs of the Mobius transformation for Ex. 2

Input	0	1	2	3	4	5	6	7	8
Output	1	5	6	3	0	0	1	2	5

In the Table 1, the inputs and outputs of the Mobius transformation are given. Now by using the Table 1, we transform the vector $\bar{X}$ to the matrix X' and then each of the rows of the matrix X' (i.e. X'_i) is sent to the nodes via output links (edges) of the source node.

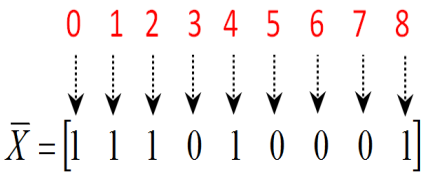

Figure 4. The Number of Positions in the Vector $\bar{X}$

The substitutions are done as follows:

From Table1, input 0 is replaced by output 1, (i.e. the element in position zero transmitted to position one in the vector $\bar{X}$ and so on) (see Fig. 5 and Fig. 6).

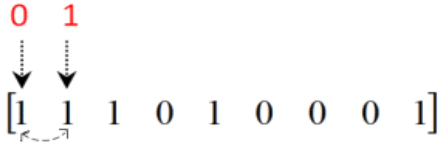

Figure 5. Position Zero is Replaced by Position One

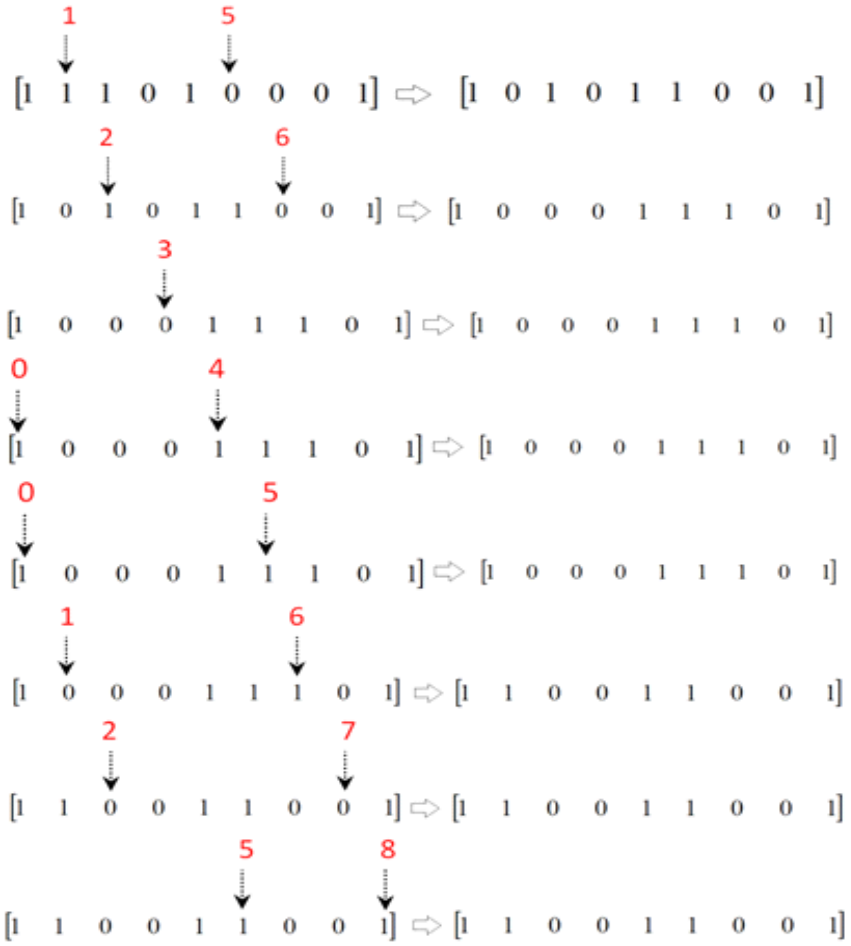

Figure 6. Steps of Replacing the Other Positions

So, the vector $\bar{X}$ is transformed to the new vector $\bar{X}'$ where:

$$\bar{X}' = [1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1].$$

Then we rearrange every of n elements from $\bar{X}'$ into one of rows of the matrix X' i.e.:

$$X' = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha^3 \\ \alpha^4 \\ \alpha^2 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 4 \end{bmatrix}.$$

Therefore, each row of X' is sent to other nodes in network via an output link of the source node.

Computational overhead:

The computational overhead of our scheme in the source node is O(n) operations because the time complexity of Mobius transform is O(n). In [16] and [17] the computational overhead is $O(n^3)$ operations. Although there exist algorithms with lower computational complexity, the resulting improvements are minor, and usually apply only to very large matrices.

5. Conclusion

We have introduced an efficient scheme for Network Coding that is based on permutation function for improving the security of the source data matrix. One of well-known permutation function called Mobius nonlinear transformation is employed. Then, we have used this transformation in our proposed scheme. The proposed scheme is capable for security to some degree in comparison with conventional methods. Thereby; our scheme can provide an initial security that protects source data against wiretappers or malicious nodes with low and linear complexity. We know that, this scheme is not a perfect security scheme, but it is a good idea for the next generation technologies like Network Coding and IOT technologies that need new security solutions. In the other words, the proposed scheme provides a light weight security level for data transmitting in the Network Coding. Our outcomes are comparable with the existence methods in this context.

References

[1] V. N. Talookia, R. Bassolia, D. E. Lucanib, J. Rodrigueza, F. H. P. Fitzekb, H. Marquesa and R. Tafazolli, "Security concerns and countermeasures in network coding based communication systems: A survey", Computer Networks, vol. 83, no. 3, **(2015)**.
[2] Y. Wei, Z. Yu and Y. Guan, "Efficient Weakly-Secure Network Coding Schemes against Wiretapping Attacks", IEEE International Symposium on Network Coding, **(2010)**.
[3] I. Woungang, S. Misra and S. Chandra Misra, "Selected Topics in Information and Coding Theory", Singapore: World Scientific Publishing Co. Pte. Ltd. 5 Toh Tuck Link, **(2010)**.
[4] K. Jain, "Security based on network topology against the wiretappingattack", IEEE Wireless Commun., vol. 11, **(2004)**, pp. 68-71.
[5] J. Tan and M. Médard, "Secure Network Coding With A Cost Criterion", Proc. Int. Symp. Modeling optim, mobile Ad Hoc Wireless netw, **(2006)**.
[6] L. R. Ford and D. R. Fulkerson, "Flows in Networks", Princeton University Press, **(1962)**.
[7] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow", IEEE Trans. Inf. Theory, vol. 46, no. 4, **(2000)** , pp. 1204 – 1216**.**
[8] S.-Y. R. Li, R. W. Yeung and N. Cai, "Linear Network Coding", IEEE Trans. Inf. Theory, vol. 49, no. 2**, (2003)**, pp. 371 - 381**.**
[9] R. Koetter and M. Médard, "An algebraic approach to Network Coding", IEEE/ACM Trans., vol. 11, no. 5, **(2003)**.
[10] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction", IEEE Trans. Info. Theory, vol. 51, no. 6**, (2005)**, pp. 1973 – 1982.
[11] E. Erez and M. Feder, "Convolutional network codes for cyclic networks", International Symposium on Information Theory, **(2005)**.
[12] R. W .Yeung, "A First Course in Information Theory", Springer, **(2002)**.
[13] D. Silva and F. R. Kschischang, "On metrics for error correction in Network Coding", IEEE Trans. Info. Theory, vol. 55, no. 12, **(2008)**, pp. 5479 – 5490**.**
[14] N. Cai and R. W. Yeung, "Secure Network Coding on a Wiretap Network", IEEE Trans. Inf. Theory, vol. 57, no. 1, **(2011)**, pp. 424 – 435**.**
[15] L. Carlitz, "A note on permutation functions over a finite field", Duke Math. J, vol. 29, no. 2, **(1962)**.
[16] J. P. Vilela, L. Lima and J. Barros, "Lightweight security for network coding", Proc. of the IEEE International Conference on Communications (ICC), **(2008)**; Beijing, China.
[17] X. Wang and W. Guo, "Novel lightweight algorithm for secure network coding", Advances in Information Sciences and Service Sciences (AISS), vol. 4, no. 20, **(2012)**.

Authors

Mohammad Hesam Tadayon, He received the B.Sc. degree in Mathematics from the University of Mazandaran, Babolsar, Iran, in 1995, the M.Sc. degree in mathematics from the University of Tarbiat Modares,Tehran, Iran, in 1997, and the Ph.D. degree in applied mathematics (coding and cryptography) from the University of Tarbiat Moallem of Tehran (Kharazmi), Tehran, Iran, in 2008. He is now an Assistant Professor at the Iran Telecommunication Research Center. His research interests include error-control coding, information theory and data security.