A Mathematical Model for Improving Lightweight Security with Network Coding

Alireza Esfahani
Instituto de Telecomunicações
Aveiro, Portugal
Email: *alireza@av.it.pt*

Alberto Nascimento
Mathematics and Engineering Department
University of Madeira
Funchal, Portugal
Email: *ajn@uma.pt*

Du Yang
Instituto de Telecomunicações
Aveiro, Portugal
Email: *duyang@av.it.pt*

Jonathan Rodriguez
Instituto de Telecomunicações
Aveiro, Portugal
Email: *jonathan@av.it.pt*

***Abstract*—As a promising technology of efficiently utilizing resources, network coding is capable of improving many performance such as throughput, security, etc. The main purpose of this paper is joining network coding and cryptography methods. In this paper, not only we propose a new mathematical method to use in network coding but also a security framework with slightly increased complexity and overhead. We use Greatest Common Divisor and Lowest Common Multiplication instead of sending original data packets, so as to provide additional guards against eavesdropping attack.**

Keywords-Security; Network Coding; Random Linear Network Coding; Eavesdropping attack.

I. INTRODUCTION

As wireless communication has become a part of people's daily life in modern society, providing efficiency, convenience and security in communications network is becoming a very crucial aspect of modern digital infrastructure. There are many aspects to security, addressing a range of natural adversaries and malicious threats. Nevertheless, one ultimate goal of security is to remain data confidential impervious to malicious or accidental eavesdropping.

Network coding was introduced for the first time by Ahlswede et al. [1] and they showed that data throughput and network robustness can be considerably improved by allowing the intermediate nodes in a network to mix different data flows through algebraic combinations of multiple datagrams. In [2], Random linear Network Coding (RLNC) was studied as a fully distributed method for performing network coding. They mentioned there is a possibility that each node in the network independently and randomly selects a set of coefficients and uses them to make linear combinations of the data symbols. In other words, RLNC allows each node in the network to make local decisions [3]. The original data packet represented as X is divided into n blocks $X_1, X_2, ..., X_n$ and each node computes and forwards linear combined packet $P = \sum_{i=1}^{n} C_i X_i$ with randomly chosen coefficients $C = (C_1, C_2, ..., C_n)$. When sufficient number of linearly are received, a node would be able to decode the original X.

Security for network coding has also been an active research area recently. The problem of secure network coding was first studied by Cai and Yeung [4]. They proposed a secure network coding scheme based on a given decodable linear network code over a sufficiently large field.

Although there are a lot of works, which have been done in security via network coding recently [5-9], network coding is still vulnerable to eavesdropping attacks. Lima et al. [10] have made cryptographic mechanism with maintaining the NC properties. They have used locked and unlocked coefficients through multiple paths from source to sinks. They have showed a noticeable reduction in computational overhead which needs for performing encryption. Motivated by gaining supplementary security approach, in this paper, we present a mathematical lightweight cryptography scheme. We mainly deal with eavesdroppers which have full access to the information about decoding and encoding. We will use properties of number theory and finite field against eavesdropping attacks and maintaining RLNC's properties.

The remainder of this paper is organized as follows. Section II presents the history of Network Coding, Preliminaries and Definition of Greatest Common Divisor (GCD) and Lowest Common Multiplication (LCM). Section III introduces the new security scheme. Section IV discusses the achievable performance. Section V concludes the paper.

II. PRELIMINARIES AND DEFINITION

In this section, we summarize the history of Network Coding; then, we discuss GCD and LCM. Also, we explain how to decompose a number to prime factors. Moreover, we examine the complexity of finding GCD and LCM in the end of this section.

A. Network Coding

After the first Max-Flow Min-Cut theorem [13], [14] presentation, Ahlswede et al. [1] elaborated a version of this theorem for information flow. Against to traditional and classical commodity flow, in which information is only routed or replicated, information flow can also employ coding operations at the nodes and we have known this method as Network Coding. After, that Linear Network Code (LNC) was appeared. In LNC, for all nodes (except source nodes) the outgoing packets are always linear combinations of the incoming ones. Yeung et al. [15] explained the relation between a linear dispersion and a generic Network Coding; also they defined a relation on the sizes of the base fields of the code. A different approach to Network Coding was presented in [16]. It proposed a completely algebraic framework, with the consequent possibility to apply the mathematical theorems of algebra on Network Coding. The objective of this framework is the definition of the transfer matrix M, which includes all the characteristics of the network itself. The translation of the Max-Flow Min-Cut theorem into the new framework modified the Network Coding problem into the problem of finding a point on an algebraic variety. Ho et al. [5] provided two results for solving the LNC multicast problem. After that and considering the results of the previous work, they proposed Random Linear Network Coding (RLNC) [2]. RLNC was studied as a fully distributed method for performing network coding.

B. LCM and GCD

In concept of arithmetic and number theory, we can define the least common multiple of two integers a and b that usually denoted by $LCM(a, b)$, as the smallest positive integer that is divisible by both a and b. Also greatest common divisor (GCD) of two integers is the largest positive integer that divides the numbers without a remainder. According to the fundamental theorem of arithmetic we can have a positive integer number which is the product of prime numbers [11], and it could be defined as:

$$n = 2^{n_2} * 3^{n_3} * 5^{n_5} * ... = \prod p^{n_p} \qquad (1)$$

$2, 3, 5, ...$ are prime numbers and $n_2, n_3, n_5, ...$ are the exponents of those prime number, which are non-negative integers ($n_i \geq 0$). By considering two integer numbers, e.g., $a = \prod p^{a_p}, b = \prod p^{b_p}$, we can define GCD and LCM as the following relations:

$$GCD(a, b) = \prod p^{min(ap,bp)} \qquad (2)$$

$$LCM(a, b) = \prod p^{max(ap,bp)} \qquad (3)$$

$$LCM(a, b) = (a * b)/(GCD(a, b)) \qquad (4)$$

Crandall *et al.* [11] explained above equations in details. Finally we have to mention that prime factorization is the decomposition of a composite number into smaller non-trivial divisors, whose multiplied result equals the original integer. We need to decompose the LCM and GCD at the end of the process which we will explain in the next section.

C. Time Complexity of LCM and GCD

In the following of this section, we present two pseudo codes for calculating GCD and also for finding prime factors of two integer numbers. LCM can be obtained from Equation (4) using a, b and $GCM(a, b)$. Regarding these two algorithms which we want to use in source and sink nodes, we can calculate the time complexity of our method.

Homer and Selman [17] have shown a lot of methods for calculating time complexity in different algorithms. According to their approach, we have one loop in both algorithms (Algorithm 1 and Algorithm 2), and all the steps of each loop can be calculated according to logarithmic-scale. Considering to the time complexity of Gaussian Elimination which is exponential-scale; it is clear that the time complexity of both algorithms which have been used in source and sink nodes, are less than exponential-scale. So our noticeable result is that the time complexity is close to the logarithmic-scale behaviour, and we can consider $O(logn)$ as the time complexity of our approach.

Algorithm 1 Algorithm for calculating Greatest Common Divisor

```
GCD(x,y)
if $y$ == 0 then
    return x;
else
    return GCD( y, x MOD y )
end if
```

Algorithm 2 Algorithm for finding prime factors

```
Input(A)
for all $i = 2$ to $i = A$ do
    if A MOD I == 0 then
        write A
        A = A / i
    else
        Continue
    end if
end for
```

Algorithm 3 Algorithm for recovering original numbers

```
Input(G,L,S)
if (G==L) then
    return (G,L)
else
    X= The difference prime numbers
    for all $i = 1$ to $i = X$ do
        Choosing P1, P2
        if S == P1+P2 then
            return P1,P2
            exit
        else
            Continue
        end if
    end for
end if
```

D. Finding unique result

As recovering the original data, the two integer a and b, through the GCD , LCM does not have a unique result, so we need to use addition information for having the unique result. Thus we want to send the summation of the original numbers before calculating GCD , LCM. So, by having three parameters, i.e., GCD, LCM, and summation of a and b, we could recover original numbers.

III. Our Scheme: M*athematical Secure* N*etwork* C*oding* (M*e*N*o*)

It is about twelve years after the emergence of the first example of network coding and specifically the butterfly network coding; a lot is already known about network in particular for the case of network multicast. Network multicast refers to simultaneously transmitting the same information to multiple receivers in the network [1][12]. The Secure Practical Network Coding (SPOC) which has been proposed in [10] is a secure algorithm with keeping the RLNC properties. The authors have tried to achieve confidentiality by protecting the locked coefficients without impairing any of the operations of practical network coding protocols. They could reduce the complexity of their algorithm in comparing to the traditional end to end encryption approaches. They have used two kinds of coefficients (locked and unlocked), but as we want to maintain the RLNC properties. It's clear whenever one node acts as an eavesdropper and hears all or part of the information which send to the sink so there is possibility that he discovers the original packet. We show the packet format scheme in Fig. 1. For instance, we suppose the basic scheme which has been used in Fig. 2. If node 4 appears as an eavesdropper so he has this chance that finds original data.

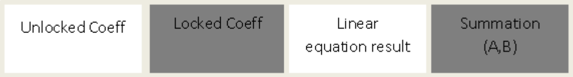

Fig. 1. Proposed packet format

In this work, not only we show that our scheme does not bring more complexity but also we create more security framework to ensure that data are sent completely in secure condition. We propose Mathematical Secure Network Coding (MeNo), a more secure framework for improving security mechanism. As we considered that eavesdrop attacker is a person who can hear information and also he has full access to all the information about coding and decoding, in this case eavesdrop attacker knows that source node has used RLNC properties and has made a linear combination of original packets, so there is possibility that this node decodes original packets. In our work, we want to use GCD and LCM instead of sending original packets. For instance if we want to send two data, e.g., $a = 4$ and $b = 10$, the first step is calculating the GCD and LCM and for this, we need using the algorithm was explained in Section II-B. In fact, we can calculate $GCD = 2$ and $LCM = 20$.

These values are all sent to the receiver to guarantee correctly recovering the original data. Having the knowledge of $GCD(a, b)$ and $LCM(a, b)$, we could have two possible recovering results $(a = 4, b = 10)$, and $(a = 2, b = 20)$. Aided with the additional knowledge of sum of a and b, it is evident that the original value of (a, b) can only be 4 and 10. Now we can use RLNC to create random linear equations and also use locked and unlocked coefficients. For recovering unique results over GCD and LCM we have to send the summation of original numbers to the sink. But, if attacker wants to discover data, by knowing (2,20), in fact he knows nothing or in the other word he realizes meaningless data, and it is the advantage of our algorithm. So, we can protect the native information from eavesdrop attackers and we can guarantee we will have completely security method without noticeable overhead. We summarize our algorithm in table I.

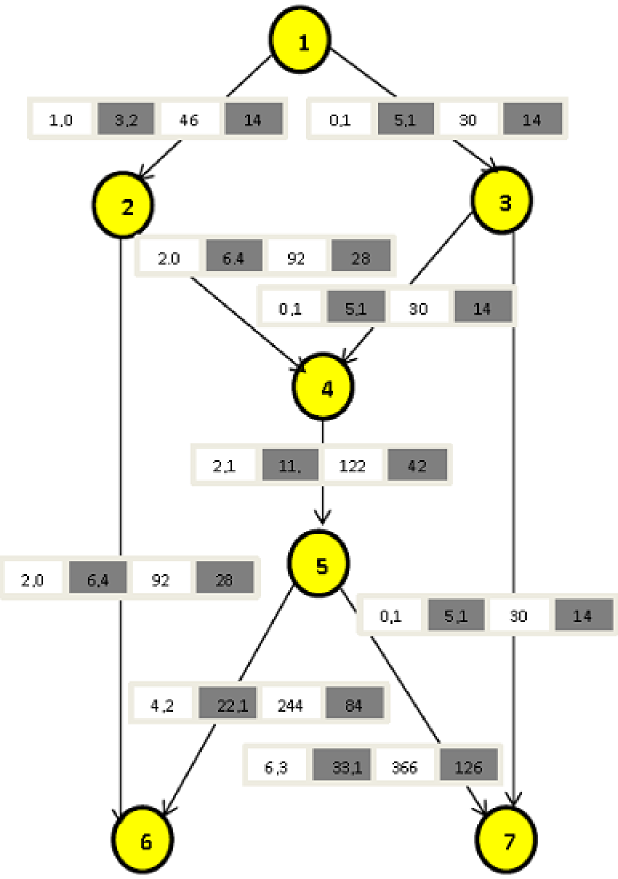

Fig. 2. Basic scheme (Encrypted information are in Gray color)

TABLE I
Summary of MeNo

Node	Operation
Source	Calculation GCD , LCM and generating RLNC
Intermediate	Combination by maintaining RLNC properties
Sink	Gaussian Elimination ,Recovering and Finding original data

According to Vilela et al. [10], the stream of packets divides into some generations of size h, and just those packets which belong to the same generation number could be combined with each other. Before moving to next section and presenting performance evaluation, we have to mention again that the kind of attack which we have considered in our approach is an eavesdropping attack which has the ability of hearing to all the

ISBN: 978-1-61208-289-9

TABLE II
TIME COMPLEXITY OF MENO

Node	Operation	Detailed cost	Total Cost
Source	Generation	$O(h^2)$	$O(h^2)$
	GCD and LCM	$O(Log(h))$	
Intermediate	Combination	$O(nh)$	$O(nh)$
Sink	Gaussian Elimination	$O(n^3)$	$O(n^3)$
	Recovering	$O(Log(h))$	
	Original data	Negligible	

information and it has full access to information of coding and decoding. On the other hand, considering to RLNC properties, they know source node has generated random coefficients, but as if they can recover these coefficients, solve the equations and recover packets, but they do not know about the process which we need to find original packets through GCD, LCM and summation of original data. We show the process for recovering original data for instance for node 6 as follows:

$$\left|\begin{array}{cc|cc} 2 & 0 & 6 & 4 \\ 4 & 2 & 22 & 10 \end{array}\right|$$

Then by implementing Gaussian elimination:

$$\left|\begin{array}{cc|cc} 1 & 0 & 3 & 2 \\ 0 & 1 & 5 & 1 \end{array}\right|$$

$\Longrightarrow (3,2),(5,1)$

$$2(3G + 2L) = 92 \tag{5}$$

$$4(3G + 2L) + 2(5G + L) = 244 \tag{6}$$

Equation (5) and (6), we have $G = 2, L = 20$. G represents the GCD value, and L represents the LCM value. In addition with the knowledge of sum value S, we could recover the original data $a = 4, b = 10$.

IV. PERFORMANCE EVALUATION

Although the proposed algorithm needs to calculate GCD and LCN, the additional complexity is in log-scale, and is much smaller compared to Guassian Elimination in exponential-scale, Hence, the complexity of the proposed scheme is slightly higher than the one proposed in [10]. The detail information is summarized in Table II.

V. CONCLUSION AND FUTURE WORK

We proposed a new mathematical secure framework in additional to the lightweight secure network coding by utilizing GCD and LCM. The proposed scheme is capable of providing additional security against eavesdrop attack, with slightly increased computational complexity only at the source and sink nodes.

For future work, as homomorphic hash function brings more computational overhead, we are going to join our model with this approach for providing authentication and lower overhead.

ACKNOWLEDGMENT

The first author acknowledges financial support of FCT through a scholarship given by Universidade da Madeira. The research leading to these results has received funding from the Fundao para a Cincia e Tecnologia (PTDC/EEA-TEL/119228/2010 - SMARTVISION).

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, ”Network information flow”. IEEE Transactions on Information Theory, 2000, pp. 1204−1216.

[2] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, J. Shi, and B. Leong, ”A random linear network coding approach to multicast”, IEEE Transactions on Information Theory, vol. 52, no. 10, 2006, pp. 4413−4430.

[3] T. HO, M. Medard, R. Koetter, D. Karger, M. Effros, S. Jun, and B. Leong, ”A random linear network coding approach to multicast,” IEEE Transactions on Information Theory 52, 2004, pp. 4413-4430.

[4] N. Cai and R. W. Yeung, ”Secure network coding,” IEEE International Symposium on Information Theory, Lausanne, Switzerland, , July. 2002, p. 323.

[5] L. Lima, S. Gheorghiu, J. Barros, M. Mdard, and A. L. Toledo, ”Secure Network Coding for Multi-Resolution Wireless Video Streaming,” Journal of Selection Areas in Communication, vol. 28, 2010, pp. 377-388.

[6] Y. Hongyi, D. Silva, S. Jaggi, and M. Langberg, ”Network Codes Resilient to Jamming and Eavesdropping,” in Network Coding (NetCod), IEEE International Symposium on, 2010, pp. 1-6.

[7] G. Zhenzhen, Y. Yu-Han, and K. J. R. Liu, ”Anti-Eavesdropping Space-Time Network Coding for Cooperative Communications,” Wireless Communications, IEEE Transactions on, vol. 10, 2011, pp. 3898-3908.

[8] S. Jaggi and M. Langberg, ”Resilient network codes in the presence of eavesdropping Byzantine adversaries,” IEEE International Symposium on information theory, 2007, pp. 541-545.

[9] C. Gkantsidis and P. Rodriguez, ”Cooperative security for network coding file distribution,” In Proceeding of IEEE INFOCOM, Barcelona, April. 2006, pp. 1-13.

[10] J. P. Vilela, L. Lima, J. Barros, ”Lightweight Security for Network Coding,” Proceeding of the IEEE International Conference on Communications (ICC 2008), Beijing, China, May. 2008, pp. 1750-1754.

[11] R. Crandall, C. Pomerance, ”Prime Numbers: A Computational Perspective”, New York: Springer, ISBN 0-387-94777-9, 2001.

[12] Q. Li, J. C.S. Lui, and D. Chiu, ”On the Security and Efficiency of Content Distribution via Network Coding,” IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 2, March/April. 2012, pp. 211-221.

[13] P. Elias, A. Feinstein, and C. E. Shannon, ”A Note on Maximum flow through a Network,” IRE Transactions on Information Theory, vol. IT, no. 2, 1956, pp. 117-179.

[14] L. R. Ford and D. R. Fulkerson, ”Maximal flow through a network”, Canadian Journal Mathematics,vol. 8, 1956, pp. 399-404.

[15] S.Y. R. Li and R. W. Yeung, ”Linear Network Coding,” IEEE Transactions on Information Theory, vol. 49, no. 2, 2003, pp. 371-381.

[16] R. Koetter and M. Mdard, ”An algebraic approach to network coding,” IEEE/ACM Transactions on Networking, vol. 11, no. 5, 2003, pp. 782-795.

[17] S. Homer and A. Selman, ”Computability and Complexity Theory”, Springer, 2000.