Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.017771
Article

Tech Science Press

A Hybrid Scheme for Secure Wireless Communications in IoT

Muhammad Irshad Nazeer[1,2,*], Ghulam Ali Mallah[1] and Raheel Ahmed Memon[2]

[1]Department of Computer Science, Shah Abdul Latif University, Khairpur, 66111, Pakistan
[2]Deparment of Computer Science, Sukkur IBA University, Sukkur, 65200, Pakistan
[*]Corresponding Author: Muhammad Irshad Nazeer. Email: irshad.nazeer@iba-suk.edu.pk
Received: 10 February 2021; Accepted: 13 April 2021

Abstract: Network Coding is a potential technology for the future wireless communications and Internet of Things (IoT) as it reduces the number of transmissions and offers energy efficiency. It is vulnerable to threat and attack that can harm intermediate nodes. Indeed, it exhibits an ability to incorporate security of transmitted data, yet a lot of work needs to be done to provide a safeguard from threats. The purpose of this study is to strengthen the existing Network Coding scheme with a set of generic requirements for Network Coding Protocols by adopting system models and a Genetic Algorithm based cryptosystem. A hybrid approach is being adopted by combining a key-distribution scheme and a code encryption approach in an experimental setup which simulated it using MATLAB and KODO libraries. The proposed hybrid approach showed good results by strengthening security along with better key management while retaining the additional benefits. Further, simulations show good performance of this cryptosystem for Network Coding enabled communications. In terms of a generic frame, to the best of our knowledge, this is the first work which takes generic security requirements into consideration, combines key-distribution and Genetic Algorithm based encryption and takes a system approach. The proposed approach is named as a Genetic Crypto Outer Code Secure Network Coding (GCOCSNC).

Keywords: Secure network coding; Genetic algorithms; Security in ad-hoc networks

1 Introduction

Ad-hoc networks are a significant part of the emerging Internet of Things. In ad-hoc networks security provision is a proving to be a challenge. In addition to the security requirements in standard networks, the latter's nodes also require confidentiality of location, absence of traffic diversion and fairness of cooperation among themselves. This situation becomes more challenging as there are very limited resources in terms of power, processing and memory. Network Coding is a technique to exploit operations at the intermediate nodes' level, providing a net advantage over Routing [1], particularly in wireless broadcast settings. There are many ways and areas to explore more of network coding's benefits, particularly in today's Internet of Things environment.

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The initial work on Network Coding contributed a lot to the fundamental theoretical aspects where many practical elements were not considered, like the assumption of prior knowledge about the topology of a given flow network. However, the subsequent work progressed towards more practical aspects. There are many research studies that supported this concept while some others shed many criticism. Wang et al. [1] provided a good discussion on the practical aspects of Network Coding. An important one is related to the security of transmitted data. The coding and decoding information is supplied to the intermediate nodes so that they can further encode data if the message is not intended for them. In this way, there are risks that the intermediate nodes may inject false data, modify the received data, or perform a malicious activity. As a consequence, the intermediate nodes can be compromised or data passing through the corresponding link can be eavesdropped. Security issues of Network Coding received a special attention from the researchers with the introduction of Secure Network Coding problem. The new problem formation resulted into a good number of Secure Network Coding schemes so far. Evaluation of the solution for a particular optimization problem is a beauty of the scientific research. The solution to the Secure Network Coding problem has also evolves over the time with its merits, demerits and limitations.

Several Network Coding schemes and protocols have been suggested in literature review [2–5]. Most of these are vulnerable to threats (e.g., wiretapping, pollution attacks, etc.) [6,7]. Generally, Network Coding schemes are classified either as state-aware and state-less [8] or as intra-flow and inter-flow Network Coding [9]. Both of these classifications are on the basis of the level of complexity. There are certain Network Coding protocols that do not take into account the network state information (for example RLNC [2]) and provide some built-in security against impersonation attacks, however, these protocols are not effective against the other types of attacks. Another class of the Network Coding protocols that do take care of network state information, exploits the opportunistic coding at the intermediate nodes to provide better performance. In terms of flow, some protocols mix packets of a single flow stream while other mix packets from multiple flows. These protocols are known as intra-flow and inter-flow Network Coding protocols, respectively. Detailed studies of these protocols and schemes lead towards the understanding that these protocols may deviate slightly from the security requirements. The analysis and comparisons of these protocols are available in the literature review [7]. To the best of our knowledge, none of these schemes is completely secured against the all security threats reported in literature so far. There are a few studies which provide the insight to the type of attacks, effect, consequences and proposed solution on the basis of partial set of the security requirements [3,10]. Apparently, there is a lack of a generic set of the security requirements for designing the optimistic protocols.

In the recent past, Computational Intelligence techniques demonstrated the optimal solutions to many computing and communication problems. Computational Intelligence refers to the models which have ability of learning and adaptation according to some rules or guidance. Some famous techniques of Computational Intelligence include Neural Networks, Fuzzy Logic and Evolutionary Algorithms. Among Evolutionary Algorithms, the most common are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Simulated Annealing (SA). Furthermore, the security and the surveillance problems tackled by these algorithms demonstrate better performance as compared to the traditional techniques. Particularly, the common intrusion attacks can be mitigated through the Genetic Algorithms. Cryptosystems designed on these lines demonstrate the adequate security strength and exhibits a strong avalanche effect [11].

The security in Network Coding based transmissions is usually provided by two means, either using the Inner Codes or using the Outer Codes. Generally, the process of implementation of security in any system involves the use of the same transmission stream for key distribution as well as the actual data. In a typical Network Coding based transmissions, data is transmitted from a source to a destination via the intermediate nodes. These intermediate nodes provide the coding opportunities and redundancy for complete decoding of the transmitted data at the destination.

In contrary to the above situation, we propose to separate key distribution from actual data transmissions. This key distribution process should use a light-weight Networking Coding scheme in order to reduce the overall complexity of the system. To do this, we propose to start the boot-strapping process first and then distribute the keys using the scheme suggested in Oliveira et al. [12]. By using the term "Boot-strapping" we refer to a process through which a node gets familiarization with the neighbor nodes [13]. The approach is simplified as: "once the bootstrapping is done, use the outer coding scheme for enhanced security" [14]. The sequence of these steps is shown in Fig. 1.

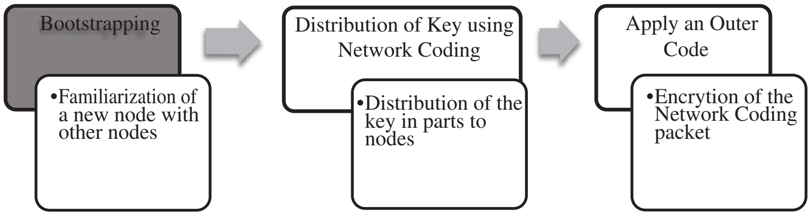

Bootstrapping
•Familiarization of a new node with other nodes
Distribution of Key using Network Coding
•Distribution of the key in parts to nodes
Apply an Outer Code
•Encrytion of the Network Coding packet

Figure 1: A hybrid model combining key distribution and outer codes

To make the solution further viable and practical, there is a need to describe a generic set of the security requirements for Network Coding protocols. This serves two purposes, first, as a reference material for the analysis of the existing protocols; second, as a guideline for designing extensions or a new, more secure protocols. The proposed requirements presented in Section 2.2 are aligned with the recommendations for routing protocols by Internet Engineering Task Force (IETF).

These are set with respect to a system's model perspective. A system's model which incorporates Secure Network Coding in existing networks (with reference to the OSI model) is presented in Section 3.3.

As the Outer Codes approach is adopted, a strong and efficient cryptosystem named Genetic Cryptosystem [11] is selected for the encryption. The way how we get benefit of this is presented in Section 3.4. To test the overall approach named GCOCSNC in an ad-hoc wireless environment, we have designed a few experiments and simulated these. The details of these experiments and simulations are described in Section 3.1. The results, discussions and conclusions are mentioned in the Sections 3 and 4 respectively.

2 Materials and Methods

The literature surveys [15] and simulation research methods both are used to explore the potential applications Computational Intelligence in cryptography and Secure Network Coding. Genetic Algorithm turned out to be a good point to start with and lead to develop a solution with better performance.

The following literature surveys helped significantly:

i. A comprehensive survey of contemporary Network Coding schemes,
ii. A comprehensive survey of Contemporary Secure Network Coding schemes, and
iii. A comprehensive survey of Computational Intelligence techniques and its applications in data security

For carrying out the following work, deduction and analogies are used.

i. Deduction of set of security requirements on the basis of literature survey
ii. Development of system model supporting security requirement

Furthermore, computer simulation methods are used to test the suggested approach. This is already established fact that to investigate existing models or before proposing a new one, practical evaluation is very helpful in identifying the gaps. We used empirical research methods for evaluation of research work due to following reasons:

a) Standards and many details of Network Coding schemes are not final,
b) Implementing real test-bed costs more,
c) Executing proposed model on real test-bed can be difficult and time consuming,
d) Repetition of experiments is challenging, and
e) Experimenting in real time is difficult to pivot on one aspect and observe the others.

We selected the appropriate simulation tools to test the proposed approach by fine tuning parameters and reconfiguring the model.

The limitations of simulation methods are directly related to the limitations of a simulation tool and developed model. The major limitation is the validity of the simulation tool. Other limitations include managing upper layers and dealing with scalability.

2.1 Experiment Setup

For the sake of experiment, Network Coding operations are applied at application and network layers. Coding operations at these layers will introduce less overhead as compared to other upper layers. Actually, Network Coding is not designed keeping in view the OSI model. To incorporate this in the OSI model either above the physical layer or somewhere at other higher layers, a shim layer is introduced so that data can be encoded or decoded soon after the symbols are detected.

A multi-hop wireless communication network topology called a butterfly network which is the most referred network topology in the literature is adopted [16] for evaluating the performance. The is the base model discussed in literature [17]. Two scenarios are selected for s. The first scenario-A deals with "single hop" while the second scenario-B tests the "multiple hops". In each scenario, the data is sent from a source device to a sink device reliably but through a lossy wireless link (characteristics are described in the relevant section). The details of these scenarios are described in Tab. 1.

Table 1: Experiment test Scenario A and B

	Scenario-A: Single hop	Scenario-B: Multiple hops
Mode	Multicast	Multicast
Payload	The sender sends a 32-byte data in ASCII format (stored in a file) to nearby nodes.	The sender sends a 32-byte data in ASCII format (stored in a file) to sink nodes.
Service Type	Best effort multicast service for multimedia The sender sends a butterfly image to receiver nodes.	Best effort multicast service for multimedia The sender sends a butterfly image to receiver nodes over the multi hop butterfly network.

Specific network services like neighbor discovery, coding and decoding services are assumed to be available with the simulation environment. The performance metrics include *throughput* and *Bit Error Rate*. The parameters that can control performance are *Packet loss rate*, *Network Coding rate*, *Generation size*, *Galois field size* and *Number of transmitted packets*. These parameters are set using a configuration file in the simulator.

2.2 Simulation Tools

There are many computer tools to simulate experiments in an ad-hoc communications network and Internet of Things. For the sake of simplicity and understanding, there is no any technical difference between a software, computer tool and simulation tool except the level of specificity and nature of their uses. A network is viewed as an abstract model for the broadcast in an ad-hoc wireless network. For selection of the appropriate simulation tools for our experiments, a list of available simulation tools along with their current version, license type and implementation language is given in Tab. 2. These tools are evaluated by means of their specific purpose, the supported features, the easiness of modelling, and the limitations. This comparison justified the selection of tools for our work.

Table 2: List of available simulation tools and their specifications

Tool	Version	License Type	Programming Language
GloMoSim	2.03	Open source	Parsec
J-Sim	2.17	Non-commercial	Java
Kodo	N/A	Open source	C++/C
MATLAB	8	Commercial	MATLAB Programming Language
Neco	Neco 2.0	GNU Public License 2.0	Python
NetScale	12.2	Commercial	Xml
NetSim	10.2	Proprietary	C, Java
NS2	3.27	Open-source	C++, Python
OMNet++	OMNet++ 5.3 Released	Open-source	C++
Opnet	17.5	Commercial	C++/Java
QualNet	QualNet 8.1 and EXata 6.1. T	Commercial	Java
WiNeSim	N/A	Academic	Graphical

The factors to select a computer tool are availability, easiness of uses, adaptability by the research community and efforts in terms of learning and implementation time. Among the tools which are listed in Tab. 2, NS-2 is an old and mature but a complex tool. For the sake of implementation of our experimental test bed, we select a MATLAB based simulator with Kodo library. The MATLAB based code is executed on Octave [18] under GNU while an academic license is obtained from Kodo to use their libraries.

2.3 A System Model for Incorporating Secure Network Coding in Existing Networks

The data rates in Network Coding based communications can achieve the maximum flow of the network. So, Network Coding protocols provide better congestion control and reliability as compared to traditional routing protocols. These protocols are responsible for distributing information to destinations attached to the network using the unicast or broadcast nature of the channel [19]. The operating principle of Network Coding [20] is as simple as a coding or mixing the packets and then unicast or broadcast them to the next hop node(s) instead of routing them to a predefined node as it happens in case of routing. This process begins at a sender node where packets from same flow or multiple flows are coded. At the intermediate or the forwarder node(s), these packets are decoded and then encoded. This process is repeated until the transmitted packets reach the destination or the sink node(s). The decoding information either travels with the coded packets or this is already available to all nodes by some predefined mechanism. The identification of the threats that can affect and limit Network Coding performance is very important. For example, if a node drops the packets, it received from predecessor node(s) results in terms of data loss equivalent to half of the throughput [3]. We consider all these operations with respect to a system model.

2.3.1 The Need for a System Model

Throughout this study, Network Coding is being discussed with respect to the routing. Routing being a well-established and well implemented field has lots of standardizations by the Internet Engineering Task Force (IETF). Many Requests for Comments (RFCs) are available to play a role in the standardization of routing and other mechanisms in the traditional store-and-forward networks. To the best of knowledge of the authors, no work has been done on these notions in Network Coding. We need a systematic perspective for Network Coding based transmissions which demands a set of generic security requirements to streamline the security aspects of Network Coding. The development of a system model as on lines of IETF can provide a generic outline of threats to Network Coding enabled transmissions and a generic set of security requirements to cope with the known security threats.

2.3.2 Definitions, Assumptions and Protocol Model

To get formalize the generic security requirements, the underlying definitions, assumptions and system model are described as below:

Our first assumption is about correct implementation of the system i.e. only those protocols or schemes are considered which are correctly implemented and do not consider the problems and actions due to poor implementation. In fact, poor implementation makes the protocol or the scheme more vulnerable.

Our second assumption is about the threat model and threat sources i.e. the threat source can be any of the components of the system model. At each component, there can be an adversary (legitimate or illegitimate) with following characteristics or abilities:

- Malfunctioning of the coding or decoding information,
- Disturbing or killing the neighboring nodes or nodes in its range,
- Getting something useful for itself
- Inserting a false packet,
- Wasting the computing power,
- Falsifying the data and
- Taking control of a node.

Following definitions are also applicable:

i. Network: A well-connected set of nodes G (V, E), where E is the set of links and V is the set nodes.
ii. Path: A path in the Network Coding based transmissions is a list of successive intermediate nodes through which the destination can be reached insuring the maximum flow.
iii. Path Properties: Path properties are obtained from the network information given as G (V, E). In Network Coding the data dissemination is not path specific so these can be the properties related to the intermediate nodes. This can be done with help of the trust model by declaring a value of trust level.
iv. Trusted Node: A node is said to be trusted if it belongs to same control plane or has some agreement of trust.
v. Evaluated Node: A node is said to be an evaluated if trust of that node is evaluated on the basis of path properties.

vi. Expected Node: A node is said to be an expected node if it provides path properties but with no certainty.
vii. Hazardous Node: A node is hazardous if path properties are not correct.
viii. Encoding/Decoding Information: This is the information in any form which is useful for encoding and decoding of other information.
ix. Eavesdropping: It is the accessing flow on all links or its subset to find some coding or decoding information, data etc. This can be an internal or an external eavesdropping.
x. Byzantine Attacks: All attacks caused by an adversary node that mimic a legitimate node. Arbitrary sort of behavior is expected from such a node.

In the next sub section, we present System Model on the basis of these assumptions and definitions.

2.3.3 Network Coding protocol: A System Model

The proposed model can support intra-flow, inter-flow, state aware and stateless protocols. This has three major components; Network Information maintenance component, Coding/Decoding Information component and Layer Coordination component; as shown in Fig. 2.

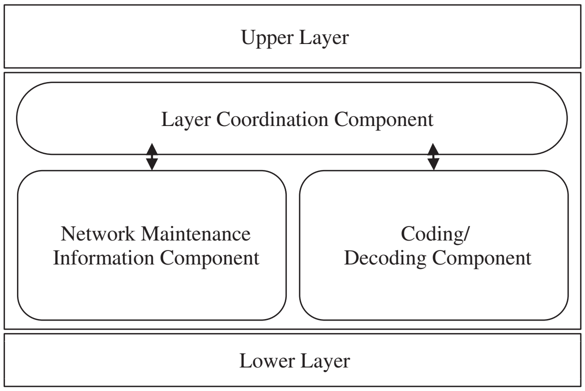

Upper Layer
Layer Coordination Component
Network Maintenance Information Component
Coding/ Decoding Component
Lower Layer

Figure 2: A system model for Secure Network Coding

As shown in Fig. 2, we incorporated Network Coding as a shim layer in the existing OSI reference model e.g. between network and transport layer. The protocols at upper layer could be the TCP or the UDP while the protocol at lower layer could be the IP. The role of each component is described as:

i. Network Information Maintenance component: This component keeps updates and maintains information related to network nodes in the range.
ii. Coding/Decoding Information component: This component takes care of the information necessary for decoding and coding of packets at a node.
iii. Layer Coordination component: This component coordinates with the lower or the upper layers to deliver, send, receive and forward data.

In case of stateless protocol, the Network Information Maintenance component is treated as a NULL.

The proposed system model caters two major categories of threats; first Eavesdropping and second Byzantine attacks. If micro perspective is taken, then following threats are expected to each component:

The threats to Network Information Maintenance component are Network Information Falsification, Network Information Modification and Wormhole attacks. Through, the Network Information Falsification attack, an adversary node provides false information about links. Through, the Network

Information Modification attack, an adversary forwarding node provides modified link state information. While, through the Wormhole attack, an adversary node, upon compromise shows a link as persistent but actually non-persistent.

Threats to the Coding/Decoding Information component can be Falsification, Modification and Dropping of such information.

Threats to the Layer Coordination component can be in the form of Acknowledgement (ACK) injection, Dropping, Modification and Delay.

There are some other threats that occur when a node is totally compromised. These are Pollution attacks, Packet Dropping attacks and Blackhole attacks. In the Packet Pollution attacks, an attacker node injects corrupted packets. In the Packet Dropping attacks, a compromised node does not forward some of packets being received by it. In Blackhole attack, a compromised node denies forwarding the received packets at all.

Taking into account the cost of threats, there shall be some consequences of these threats. These consequences are well known in literature and are named as Usurpation, Deception, Disruption and Disclosure in the order of their importance.

All the falsification attacks result in usurpation. Any sort of pollution, dropping or modifying of a packet or ACK and wormhole attack would cause deception. Injecting false packets will overload the node and hence cause disruption. All eavesdropping attacks result into data disclosure.

A set of the security requirements should deal with prevention against the conditions of consequences. This prevention may be against the existence of threat sources to detect if there is any attack.

2.3.4 Generic Security Requirements

On the basis of assumption and definitions already described in Sections 2.3.2 and 2.3.3, a set of generic security requirements for Network Coding is listed below. These are described in a manner that a main requirement (MR) follows none, one or more sub requirements. The words in capital letters and security terminologies follow conventions and interpretations of the RFC 2119 [21] and RFC 2828 [22] respectively.

The First main requirement is:

MR (1) Correct coding or decoding information SHOULD be made available for the encoding or decoding OR forwarding at a node. This process MUST be completed in time-efficient manner.

The sub requirements for MR (1) are:

MR (1.a) When the encoding or decoding information is unavailable or incorrect, the first main requirement means that a correct decoding information SHOULD be made available, either through the use of another protocol or it SHOULD be discarded.

MR (1.b) When decoding information is available and correct, the first main requirement means that misuse of the encoding or decoding information SHOULD NOT jeopardize decoding information availability or correctness, as this would also compromise the correct forwarding.

The second main requirement is:

MR (2) The intermediate node MUST ignore the packets irrelevant to it.

The third main requirement is:

MR (3) The Network Information SHOULD be available in time-efficient manner to provide opportunistic coding.

2.4 Implementing Secure Network Coding using Computational Intelligence

This section describes implication of GAs to provide security and the way we adapt this to make Network Coding secure.

2.4.1 Genetic Crypto

The key idea of Genetic algorithms is to imitate the randomness of nature. On the basis of a natural selection process and behavior of the population, the individuals adapt to the surroundings. The survival and reproduction of the individuals are supported by exclusion of the least-fit individuals. The population is generated in such a way that the individual with the highest fitness value is most likely to be replicated and the least-fit individual is discarded on the basis of a threshold which is set by an iterative application of stochastic genetic operators [23]. The performance of this sort of solution is measured in terms of the key strength. The key strength is categorized by key search space size (i.e., how many alternative keys can be tried to break the cipher) and the attack scenario (i.e., how much time is required by the eavesdropper to attack). Encryption and decryption operations are also performed by implementing this algorithm. The key strength is evaluated in terms of the attack time using a system called GRC[1] which is in fact an interactive brute force key "Search Space" calculator. A cryptosystem implemented as a result of this approach known as a Genetic Crypto which is strong enough and exhibits strong avalanche effect [11] is incorporated in our proposed GCOCSNC approach.

2.4.2 Implementing Genetic Crypto to Secure Network Coding

This section describes an end to end system for implementation of the proposed approach. The block diagram of the process from the source to the destination is given in Fig. 3.

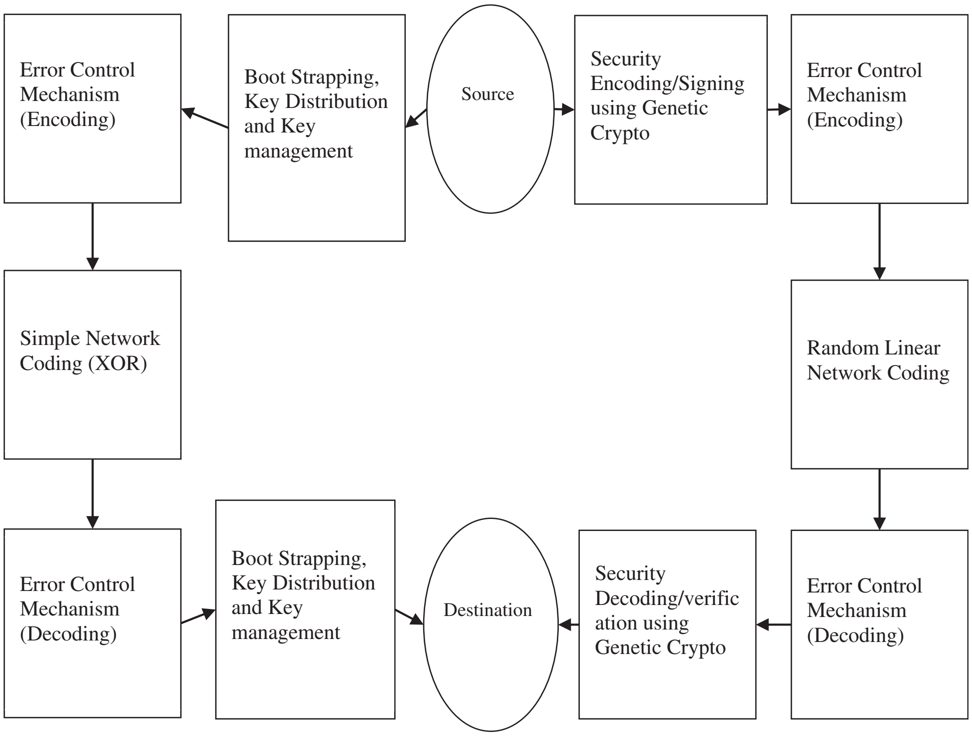

Error Control Mechanism (Encoding)
Boot Strapping, Key Distribution and Key management
Source
Security Encoding/Signing using Genetic Crypto
Error Control Mechanism (Encoding)
Simple Network Coding (XOR)
Random Linear Network Coding
Error Control Mechanism (Decoding)
Boot Strapping, Key Distribution and Key management
Destination
Security Decoding/verific ation using Genetic Crypto
Error Control Mechanism (Decoding)

Figure 3: Block diagram of proposed Secure Network Coding model

[1] *GRC's* Interactive Brute Force Password "Search Space" *Calculator* available at https://www.grc.com/haystack.htm.

The blocks to the left of source and destination show the key distribution activities, which use the routine error control and separate use of a simple Network Coding scheme (XOR). The blocks to the left of source and destination show the separate encoding and decoding of data stream for which we intend to use the Genetic Crypto. The source and the intermediate nodes know identifiers of the sinks and source, and the sink nodes share symmetric keys to encrypt data when needed. The source node can establish secure connection with the intermediate node or any intermediate node can establish a secure connection with any other intermediate node. Fig. 4 shows one example of this process. Here, the node T and X establish a secure connection through an intermediate node W.

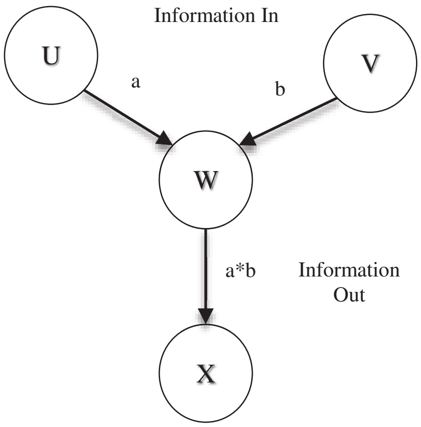


Figure 4: Coding at intermediate nodes

The initial phase of the process is boot-strapping. After boot-strapping, key distribution process is carried out among the participating nodes. Random Linear Codes are applied to get the benefits offered by Network Coding. The encoding coefficients used in the Random Linear Codes are encrypted using the Genetic Crypto. The decoding is performed at the destination node subject to the availability of sufficient number of packets which is most probable because of the redundancy. Fig. 5 shows flow of the overall process.

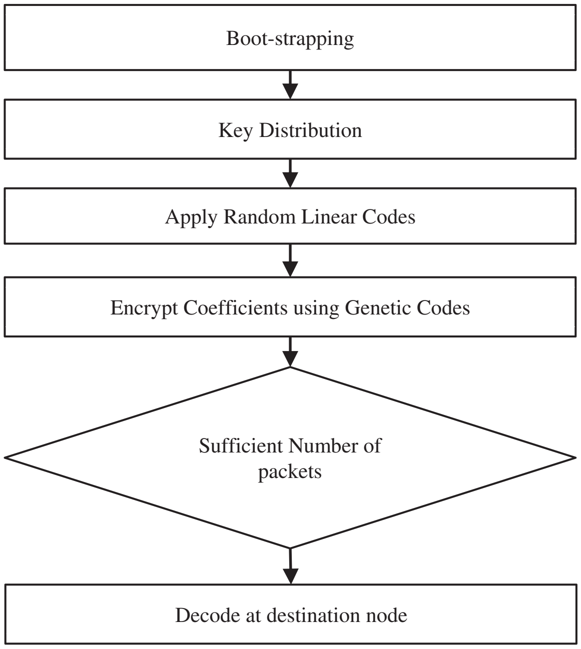

Boot-strapping
Key Distribution
Apply Random Linear Codes
Encrypt Coefficients using Genetic Codes
Sufficient Number of packets
Decode at destination node

Figure 5: Flow diagram of the overall process

We, now, explain each part of our proposed scheme, GCOCSNC, one by one. The bootstrapping process is shown in Fig. 6.

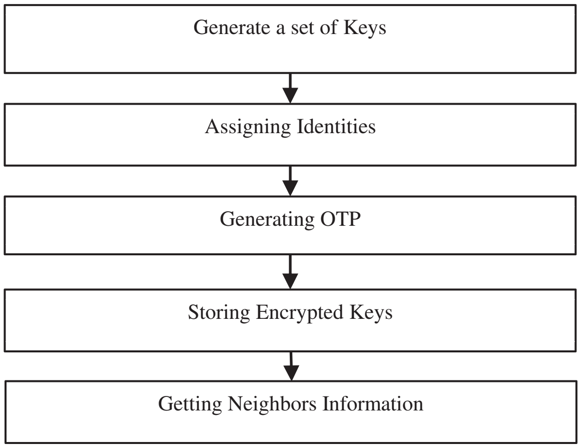


Figure 6: Flow diagram of Boot-strapping process

Following steps are taking place for Boot-strapping.

Step E-1: Key Generation: Generate a set of keys for each participating node. These keys must be statistically independent.

Step E-2: Assignment of Identities: Assign an identity to each node. The identity had two types, the local and the global. The global identity is made by using the node identity and the local identity as per following format: *global_key_identity* (32 bits) =*node_identity* (24 bits) || *local_key_identity* (8 bits)

Step E-3: OTP Generation: Generate One Time Pad (OTP). The size of OTP must be equal to the size of the *Key* made in step E-1. The OTP follows the Bernoulli distribution with probability equals to 0.5

Step E-4: Storing the Encrypted Keys: The global key is encrypted. The tuple which is stored here is: (*global_key_identifier, E (global_key XOR OTP*))

Step E-5: Getting Neighbors' Information: Save the number of keys which is |*edges-out*| in each node. Each node is aware of its own *node_identity* and *local_key_ identity.*

The next part of the process is the Key Distribution. Key distribution process is shown in Fig. 7.

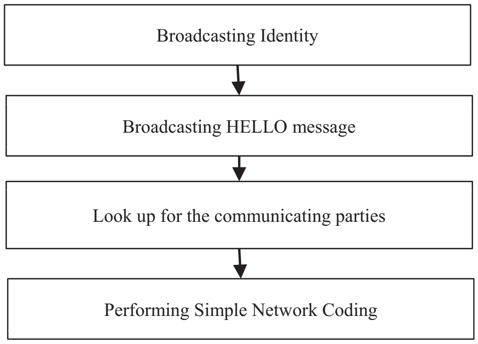

Broadcasting Identity
Broadcasting HELLO message
Look up for the communicating parties
Performing Simple Network Coding

Figure 7: Flow diagram of Key distribution process

Following are the steps of Key Distribution phase:

Step K-1: Identification: Each node broadcasts its *identity* through broadcast messages and updates its *list_of_neighbors*

Step K-2: Broadcasting HELLO message: The intermediate node broadcast HELLO messages. In return each node sends the *number_of_neighbors* and the *list_of_neighbors* as a tuple: (*number_of_neighbors*, *list_of_neighbors*)

Step K-3: Look up for the communicating parties: The source looks as if the destination is in the *list_of_neighbors* of any node. If so, the intermediate node sends *global_key_identity* of the source || *global_key_identity* of the destination node using Network Coding operations.

Step K-4: Performing Simple Network Coding: The neighboring nodes perform XOR of the received information with their own *Key* to get the *Key* of each other.

Boot-strapping and Key distribution processes guarantee that the attacker cannot see keys from the XOR messages. Any sort of injection can be detected by the legitimate node by rejecting an invalid *Key*.

The last phase of the proposed scheme is Applying Outer code. This is in fact encrypting the coefficients using Genetic Crypto and sending these through the Random Linear Network codes. The steps of this process are different depending upon the type of nodes. In our network, a node can be a source, an intermediate and a destination node. The process is shown in Fig. 8.

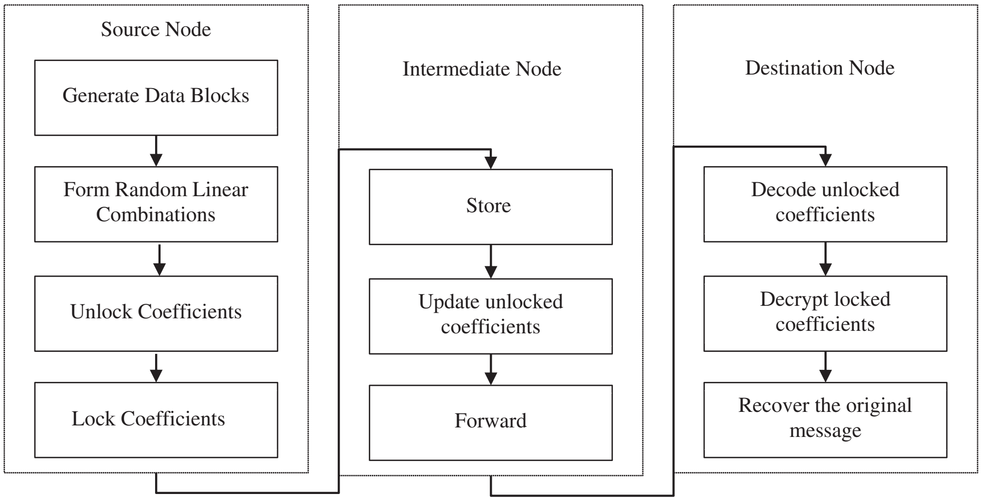

Source Node
Generate Data Blocks
Form Random Linear Combinations
Unlock Coefficients
Lock Coefficients
Intermediate Node
Store
Update unlocked coefficients
Forward
Destination Node
Decode unlocked coefficients
Decrypt locked coefficients
Recover the original message

Figure 8: Flow diagram of applying Outer code process

The steps that take place at Source Node are:

Step OCA-1: Generation of the data blocks: The source node has n block of data

Step OCA-2: Formation of Random Linear combinations: Here a random linear combination is formed from the n packets (the current generation identity is set in a packet to be sent)

Step OCA-3: Making Unlocked coefficients: Unlocked coefficients or the *local_encoding_vectors* are generated in this step.

Step OCA-4: Making Locked coefficients: The *global_encoding_vectors* are made using the Genetic Crypto in this step.

The steps that take place at Intermediate Node are:

Step OCI-1: Storing: The received packets from upper nodes are stored.

Step OCI-2: Updating Unlocked coefficients: Update the unlocked and locked coefficients for further processing.

Step OCI-3: Forward: Forward the updated packets to the next nodes.

The steps that take place at Sink Node are:

Step OCS-1: Decoding Unlocked coefficients: Decode the received message using unlocked coefficients to get access to the locked coefficients.

Step OCS-2: Decrypting locked coefficients: Decrypt the locked coefficients using the *Key*

Step OCS-3: Recover the original message

3 Results and Discussions

As far as the performance of the Genetic Crypto is concerned, it was compared with the DES and the AES symmetric key cryptosystems in terms of the encryption time, the decryption time and the key strength [13]. The performance improvement is recorded in terms of the encryption time, the size of the key search space and the attack time. Encryption time of DES and AES recorded higher than the Genetic Crypto. As described in Section 2, our focus for proposing GAOCSNC has been to reduce the number of encryption and the encryption time as compared to traditional systems. In order to see performance of our proposed solution GCOCSNC, we used the KODO simulator as described in Section 2.2. The parameters are set in the configuring file to values as shown in Tab. 3.

Table 3: Parameters setting in the configuration file

Parameter	Value	Parameter	Value	Parameter	Value
Packet loss rate	0.05	Network Coding rate	0.9	Generation size	1300-1400 bytes
Galois field size	GF(2)	Number of transmitted packets	100		

Following quantities are plotted against number of nodes:

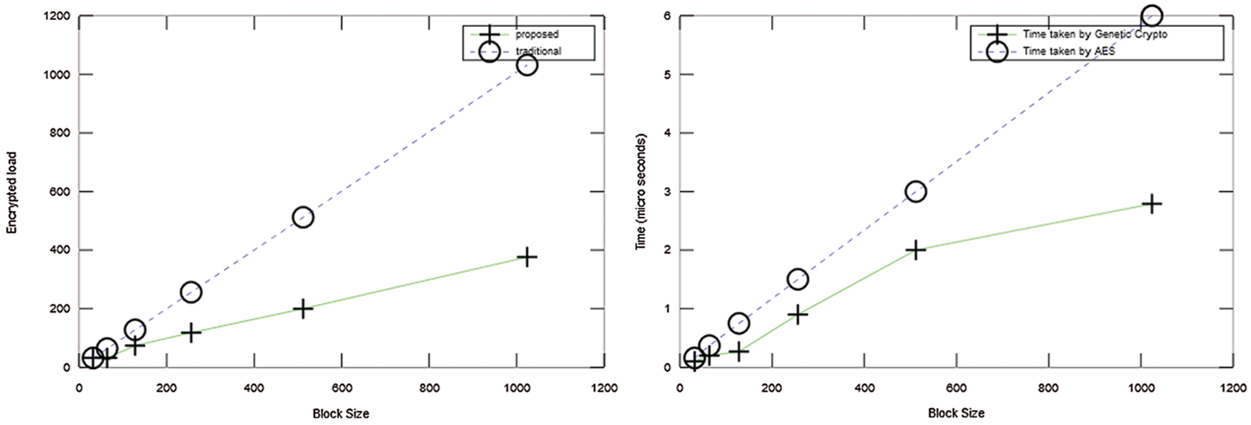
a) the amount of data to be encrypted as compared to tradition encryption, and

b) the time taken by the encryption and decryption process.

The amount of data to be encrypted as compared to the traditional cryptosystems is vital for the success of our proposed solution. To verify this, we simulated the experiment as per our experiment design and test-bed defined in Section 2.1 with varying block size (8, 16, 32, 64 and 128 KB) and recorded the amount of encrypted data. The tests have been performed with the Genetic Crypto and with the traditional block cipher (AES). The amount of encrypted data is recorded for both cryptosystems with varying block size. Fig. 9a shows clearly that our solution demonstrated an edge over the traditional encryption technique.

The time taken by the encryption and decryption process is also important for the success of our proposed solution. To verify this, we simulated the experiment as per our experiment design and noted the time to encrypt data. The amount of encrypted data is recorded for both cryptosystems with varying block size. Fig. 9b shows clearly that our solution demonstrated an edge over the traditional encryption techniques.

This work can further lead to improvements in many areas: Efficient hardware implementation of a cryptosystem is always helpful in provision of fast and efficient communications. As a Genetic Crypto is composed of simple computation operations and offers good security strength. So it is a suitable candidate for hardware implementation. Therefore, this model also seems a viable candidate to experiment the hardware implementations as a specific device that could be commercialized later as a product.



(a) (b)

Figure 9: a) Amount of data to be encrypted as compared to traditional encryption b) Time taken by encryption and decryption process

As many networks of this era are heterogeneous and their all-the-time availability is not assured, so there is a need to test such systems in the Delay and Disruption Tolerant Networks (DTNs). For the performance evaluation of the proposed model in such an environment could be an interesting project. In future, the researchers can work to improve our algorithm for multimedia encryption like images, videos and audios. From the evaluation point of view, the Genetic Crypto can be compared with other cryptographic algorithms. Also, one can use more statistical techniques for evaluation of the key randomness.

There are many situations where deployment of isolated networks are required beyond the need of an ad-hoc network or any other infrastructure-less environment. Such environments or situations are needed in many classified projects, defence organizations and law enforcement agencies. The proposed work in article could be adopted and extended for private networks which are isolated from the Internet or usual store-and-forward networks. We recommend to adopt this model along with the use of Wi-Fi-direct technology [23]. There are certain proposals to use Network Coding as a network service [24]. The proposed system model also has an ability to be implemented by Software Defined Network.

4 Conclusions

The literature showed a lot of concentration on linear codes specially the Random Linear Network Codes as a significant work on Network Coding. There are many polynomial time approximate solutions or algorithms that can solve the problem of Secure Network Coding problem. The current state-of-the-art Network Coding has been investigated and found that Random Linear Network codes are the most successful codes so far from the perspective of implementation. The proposed approach tested a few simulation scenarios. For the future investigations, testing the proposed approach in a wide variety of network conditions, for example with different network topologies, considering node mobility, and varying the number of total nodes would help in understating the further robustness of the proposed model.

Many security issues have been focused. Problem formulation, proposed approach and methodology suggested a Secure Network Coding scheme that meet the generic security requirements. This article described the way to achieve these requirements by incorporating Computational Intelligence techniques particularly Genetic Algorithms. Some of the existing schemes combined in an appropriate order offered some good results. The model proposed performed well in terms of security strength and offered better key management. Security aspects of Network Coding in the context of system aspects of a protocol make this model to support intra-flow, inter-flow, state aware and stateless communications. A generic set of the security requirements for Network Coding protocols and systems can be used as a reference material for current Network Coding protocols and systems analysis. These requirements can also serve as guidelines for extension design and new, more secure, Network Coding protocols and systems.

Acknowledgement: This work is the extension of our unpublished work presented in the Ph D thesis submitted at Shah Abdul Latif University, Khairpur, Sindh, Pakistan.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Wang and B. Li, “How practical is network coding?,” in *Proc. IWQoS*, New Haven, CT, USA, pp. 274–278, 2006.

[2] L. Lima, M. Medard and J. Barros, “Random Linear Network Coding: A free cipher?,” in *Proc. IEEE ISIT*, Nice, France, pp. 546–550, 2007, 2007.

[3] J. Dong, R. Curtmola, R. Sethi and C. Nita-Rotaru, “Toward secure network coding in wireless networks: Threats and challenges,” in *Proc.4th WSNP*, Orlando, FL, USA, pp. 33–38, 2008.

[4] P. A. Chou and Y. Wu, “Network coding for the internet and wireless networks,” *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77–85, 2007.

[5] P. A. Chou, Y. Wu and K. Jain, “Practical Network Coding,” in *41st AACC*. Monticello, IL, USA, pp. 1–10, 2003.

[6] T. M. J. Feldman, C. Stein and R. A. Servedio, “Secure network coding via filtered secret sharing,” in *42nd AACC*. Monticello, IL, USA, 30–39, 2004.

[7] L. Lu, Y. Liu, L. Hu, J. Huai, L. M. Ni *et al.,* “Pseudo trust: Zero-knowledge authentication in anonymous P2Ps,” *IEEE Transactions on Parallel and Distriuted Systtems*, vol. 19, no. 10, pp. 1325–1337, 2008.

[8] J. Tan and M. Medard, “Secure Network Coding with a cost criterion,” in *Proc. 4th WiOpt*, Boston, MA, USA, 2006, pp. 1–6, 2006.

[9] P. Garrido, D. Gómez, R. Agüero and J. Serrat, “Combination of intra-flow Network Coding and opportunistic routing: Reliable communications over wireless mesh networks,” in *Proc. 8th ICSTT*, Athens Greece, pp. 191–199, 2015.

[10] V. N. Talooki, R. Bassoli, D. E. Lucani, J. Rodriguez, F. H. P. Fitzek *et al.,* “Security concerns and counter measures in network coding based communications systems: A survey,” *Computer Networks*, vol. 83, no. 4, pp. 422–445, 2015.

[11] M. I. Nazeer, G. A. Mallah, N. A. Shaikh, R. Bhatra, R. A. Memon *et al.,* “Implication of genetic algorithm in cryptography to enhance security, International Jornal of Advnced,” *Computer Science and Appllications*, vol. 9, no. 6, pp. 375–379, 2018.

[12] P. F. Oliveira and J. Barros, “A Network Coding Approach to secret key distribution,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 414–423, 2008.

[13] J. R. Vacca, *Computer and Information Security Handbook*. Elsevier Science, pp. 15–45, 2012.

[14] J. P. Vilela, L. Lima and J. Barros, “Lightweight security for network coding,” in *IEEE ICC*. Beijing, China, pp. 1750–1754, 2008.

[15] M. I. Nazeer and M. S. Shaikh, “Secure Network Coding schemes: Comparisons and broader perspective, Sindh University Research,” *Journal (Sci. Ser.)*, vol. 43, no. 1A, pp. 85–90, 2011.

[16] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard *et al.,* "XORs in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

[17] D. Wang, Q. Z. Q. Zhang and J. L. J. Liu, "Partial Network Coding: Theory and Application for Continuous Sensor Data Collection," in *14th IEEE IWQOS*, Haven, CT, USA, pp. 93–101, 2006.

[18] J. W. Eaton, "GNU Octave and reproducible research," *Journal of. Process Control*, vol. 22, no. 8, pp. 1433–1438, 2012.

[19] M. Alotaibi, "Security to wireless sensor networks against malicious attacks using Hamming residue method," *Journal on Wireless Communication and Networking*, vol. 8, no. 1, pp. 1–7, 2019.

[20] R. W. Yeung, S. Y. R. Li, N. Cai and Z. Zhang, "Network Coding Theory Part I: Single Sources," *Foundions and Trends in Communication and Information Theory*, vol. 2, no. 4, pp. 241–329, 2005.

[21] S. Bradner and H. University, "2119-Key words for use in RFCs to Indicate Requirement Levels Status," *Network Working Group,* 1997. https://www.rfc-editor.org/rfc/pdfrfc/rfc2119.txt.pdf.

[22] R. Shirey, "RFC 2828-Internet security glossary," 2000. http//www. faqs. org/rfcs/rfc2828. html.

[23] D. Camps-Mur, A. Garcia-Saavedra and P. Serrano, "Device-to-device communications with WiFi direct: Overview and experimentation," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, 2013.

[24] D. Szabó, A. Csoma, P. Megyesi, A. Gulyás and F. H. P. Fitzek, "Network Coding as a service," *Infocommunications Journal*, vol. 7, no. 4, pp. 2–11, 2015.