Hindawi
Journal of Electrical and Computer Engineering
Volume 2017, Article ID 4657831, 10 pages
https://doi.org/10.1155/2017/4657831

Research Article

Energy Efficient Partial Permutation Encryption on Network Coded MANETs

Ali Khan,[1] Qifu Tyler Sun,[1] Zahid Mahmood,[1] and Ata Ullah Ghafoor[2]

[1] *School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China*
[2] *Department of Computer Science, National University of Modern Languages, Islamabad, Pakistan*

Correspondence should be addressed to Qifu Tyler Sun; qfsun@ustb.edu.cn

Received 15 January 2017; Accepted 20 March 2017; Published 6 April 2017

Academic Editor: Xiong Li

Mobile Ad Hoc Networks (MANETs) are composed of a large number of devices that act as dynamic nodes with limited processing capabilities that can share data among each other. Energy efficient security is the major issue in MANETs where data encryption and decryption operations should be optimized to consume less energy. In this regard, we have focused on network coding which is a lightweight mechanism that can also be used for data confidentiality. In this paper, we have further reduced the cost of network coding mechanism by reducing the size of data used for permutation. The basic idea is that source permutes only global encoding vectors (GEVs) without permuting the whole message symbols which significantly reduces the complexity and transmission cost over the network. We have also proposed an algorithm for key generation and random permutation confusion key calculation. The proposed scheme achieves better performance in throughput, encryption time, and energy consumption as compared to previous schemes.

1. Introduction

Mobile Ad Hoc Networks (MANETs) have dynamic topology, which means, on requirement, devices act as nodes and establish a network for communication (Figure 1). All nodes are battery powered [1]. As MANETs do not need any infrastructure, nodes move freely in arbitrary direction, so nodes easily enter and leave the network at any moment. There is a possibility when a node cannot send information to another node directly within its communication range. To overcome this issue some intermediate nodes are used for routing the information from one node to another by multiple hops [2].

MANETs were thought as one of the most innovative and challenging wireless networking paradigms [3] at its evolution. Potentials of MANETs made ad hoc networking a key area for building Forth-Generation (4G) wireless networks and hence MANETs gained thrust and resulted in remarkable innovation for mobile network paradigm [4]. With advances in research, MANET becomes essential communication technology in military tactical environment to help in military deployment among soldiers, vehicles, and operational command centers [5]. MANET applications can also be used in law enforcement, other security sensitive environments, emergency relief scenarios, public meetings, and virtual class rooms.

Network encoding is used to transmit the maximum flow of data in a message by using encoding mechanism. It achieves transmission and reduction in communication overhead and better throughput instead of just storing and forwarding as in traditional routing. The traditional routing where the routers typically store and forward the information cannot be overlaid. Network coding, as introduced by the pioneers Ahlswede et al. [6], shows how information is encoded at intermediate or source nodes for efficient transmissions. The basic concept of network coding can be easily understood by butterfly network as explored in Figure 2. In this butterfly network topology a source (l) wants to transmit information to two nodes acting as destinations m and n. Each edge is represented as error-free channel which has the ability to deliver a single bit per channel use. The source sends two bits a and b, but instead of routing one and blocking the other, node x transmits their XOR. Node m receives a and $a \oplus b$. Since $a \oplus (a \oplus b) = b$, node m can recover

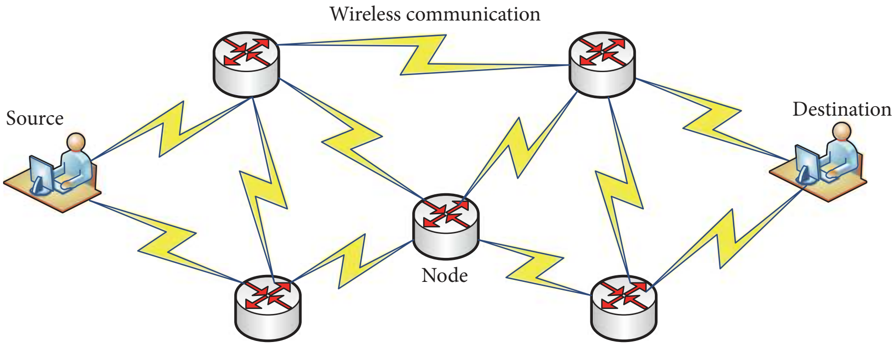


Figure 1: MANET topology.

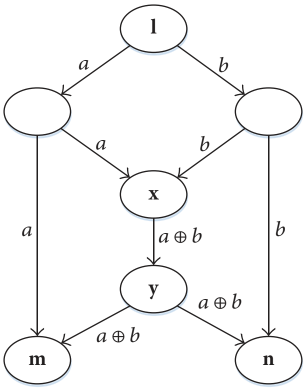


Figure 2: Network coding (butterfly network).

both a and b. By using coding operation at bottleneck node and then decoding at sink nodes, the multicast throughput has been enhanced to 2 bits, beyond what can be done in conventional routing. Network coding has advantages like enhancing better throughput [7], network robustness [8], and reducing network congestion [9]. Network coding has applications in many areas such as Ad Hoc Network [10], delay tolerant networks [11], P2P network [12], wireless sensor network [13], and content distribution network [14].

The main problem in the conventional routing approach is highlighted when a message is transmitted through a number of intermediate nodes. At one point as depicted in the butterfly network, the intermediate node x is the bottleneck as it receives multiple packets but has a single channel to forward the packets. It results in network congestion. Secondly nodes also receive redundant data from different other nodes which can be removed by using network coding.

This paper presents an encryption scheme for MANETs which fully takes advantage of the security property of network coding. As global encoding vectors (GEVs) are essential for decoding so reordering the components in a GEV randomly makes significant confusion for the eavesdropper to get meaningful information. In the paper, we studied energy efficient encryption by merely permuting the components of GEVs, without manipulating the entire messages during encryption. We refer to the proposed scheme as partial permutation based encryption scheme. A similar idea has been considered in a scheme in [15], which randomly permutes the whole message of length n with $n!$ possibilities and uses AES for encryption. Instead of permuting the whole message of length n, we permuted only the components in GEVs of length h which gives much lower computational complexity as $h \ll n$. It reduces the complexity of the scheme and energy consumption and is thus a more efficient encryption scheme. To further enhance the security of the message, we proposed an algorithm for dynamic random permutation key generation of the key.

The paper is organized as follows: Section 2 discusses the related work; Section 3 explores the proposed partial permutation based encryption scheme on network coded MANETs. Section 4 evaluates the performance and Section 5 concludes the work.

2. Related Work

The transmission energy reduction in network coding applications has received a lot of attention. We have explored different schemes in this context to identify the network coding mechanisms that claim to achieve efficient energy consumption. In this regard, Wu et al. [16] explores the solution to finding minimum energy of multicast tree expressed as linear program that is solved in polynomial time encoding at intermediate nodes. This is in contrast to the fact that the same problem is NP-complete in the case of traditional routing as investigated by Čagalj et al. [17]. Fragouli et al. [18] investigated efficient broadcasting problem in MANET and proposed probabilistic algorithms. Energy saving is done by lowering number of transmissions, as illustrated in Figure 3, where nodes are allowed to do the encoding of packets. Suppose there are six nodes where every node communicates only with both sides of its neighbors. Every node has to broadcast a message to other nodes.

It shows that without using network coding (NC), every message will need four broadcasts but with NC transmissions per message has decreased to three. So in this way 25% energy is saved in transmission only. The same problem is considered by Li et al. [19], where authors proposed deterministic approach based on PDP (Partial Dominant Pruning). The algorithm depends on two hop neighbors and opportunistic listening to encode packets.

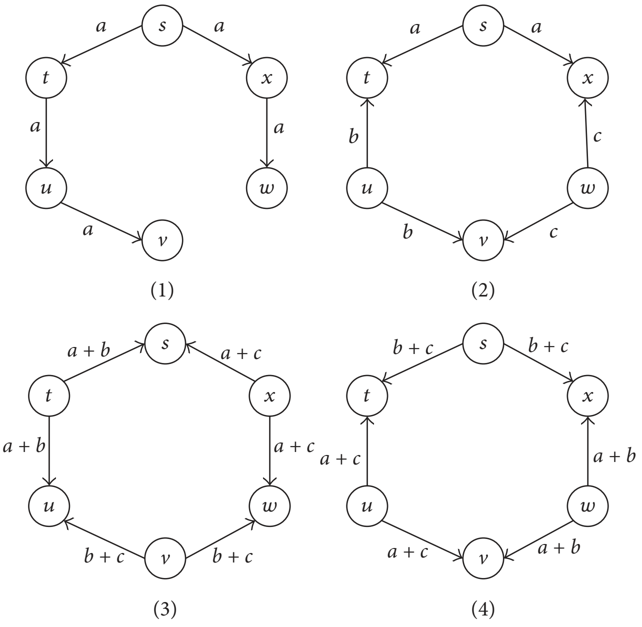


Figure 3: Reduction of transmission in MANETs.

Other than minimizing energy consumption during transmission in MANETs, network coding also has security properties as follows. Bhattad and Narayanan [20] introduced weak security, which means that a system is secured if adversary cannot get any meaningful information from the adversarial attack. Authors showed that random linear network coding is weakly secured with high probability when coding is applied on a large finite field. Lima et al. [21], after considering the threats of the intermediate nodes, developed a security criterion to access intrinsic security provided by network coding. Authors observed that security is directly dependent on the network topology as well by deriving the relationship between security level and field size. Based on this weak security research model, Wang et al. [22] designed a polynomial time deterministic code to secure linear network coding. They showed that optimal throughput between a single source and a paired destination for multiple streams is achieved by using this algorithm.

Many secure NC based cryptographic schemes have been proposed. A signature based scheme proposed by Yu et al. [23] detects and filters out the polluted messages. It used homomorphic signature function by source to delegate signing authorities to the forwarders that means intermediate nodes can generate signature for their output messages without contacting the source. SPOC (Secure Practical Network Coding), an end-to-end lightweight security scheme, is proposed by Vilela et al. [24], in which source encrypts GEVs of every message after performing random linear coding with an additional set of GEVs for network coding. Receiver recovers source messages by using decode-decrypt-decode steps. Fan et al. [25] proposed another similar scheme using HEF (Homomorphic Encryption Function) as introduced by Benaloh [26]. In this proposed scheme, the coding coefficients used HEF approach for encryption. Linear combinations can be performed directly on these encrypted coefficients because of the homomorphic nature of HEF. This result does not need additional coding coefficients by SPOC.

Wei et al. [27] proposed an efficient encryption scheme that used permutation function that randomizes the message vectors to make confusion for the adversary. Zhang et al. [15] introduced a permutation coding scheme, called P-Coding, which is a lightweight encryption above network coding in MANETs. Their scheme significantly reduced energy consumption because of minimizing the security cost. Their scheme also exploits intrinsic property of security in network coding by using simple permutation encryption. In P-Coding scheme GEVs and message symbols are permuted together. Its complexity is significantly large as compared to the proposed scheme which only permutes components of GEVs. As data needs to be protected at every node in MANETs, energy efficiency can be achieved by the more efficient encryption and decryption processes in the proposed scheme. The conventional approach of encrypting the information is to use a symmetric key algorithm. We proposed a dynamic random permutation key that uses symmetric parameters.

Researchers have given considerable attention towards the energy efficient schemes. Energy efficient scheme, which is also termed as green computing, is one of the most important areas of research these days. Dutta and Culler [28] proposed a mechanism to reduce the energy utilization of mobile wireless nodes by reducing idle listening time of the nodes. Energy efficiency has also been the focal point of routing protocols. An OSLR based routing algorithm is proposed by Tan et al. [29] in which during node selection process different trust levels are used. Venkanna et al. [30] proposed a route splitting algorithm in which a solution to battery faults is provided by the persistent performance adapting the nodes. Another exclusive method for power consumption is proposed by Takeuchi et al. [31] which provides high performance in the dynamic environment based on creating assurance network. The energy consumption has made critical requirement to adopt effective green computing in wireless communication. Our work contributes to green computing from two facets: it saves energy in wireless environment during secure data transmission and introduces energy efficient partial permutation based encryption.

3. Preliminaries

We adopt the similar system model discussed by Zhang et al. in [15]. Let π be a sequence containing each element of set $(1, \ldots, h)$ once and only once as a permutation with length h. Let $\pi(i)$ be the ith element of π. The product of two permutations π_1 and π_2, defined by $\pi_1 \times \pi_2$ or $\pi_1\pi_2$, is calculated using $\pi_1\pi_2(i) = \pi_1(\pi_2(i))$. Denote by π^{-1} the inverse of π; that is, $\pi^{-1}\pi(i) = i$.

Definition 1. Consider a sequence $\mathbf{a} = (a_1, a_2, \ldots, a_h)$ over a finite field $\mathbb{F}_q$ and a permutation k on $(1, \ldots, h)$. The

Permutation Encryption Function (PEF) using key k on $\mathbf{a}$ is defined in

$$E_k(\mathbf{a}) = \left[a_{k(1)}, a_{k(2)}, \ldots, a_{k(h)}\right]. \quad (1)$$

In the same way, we define permutation decryption function on a ciphertext c using key k as $D_k(c)$, satisfying $D_k(E_k(\mathbf{a})) = \mathbf{a}$.

There is a Key Distribution Center (KDC), responsible for establishing symmetric key so that source and destinations share a PEF key at the initial stage of the scheme. For effectiveness of PEF, the encryption key should be generated randomly. In the existing scheme [15], the generated session key sharing process uses AES encryption key management technique which is not resource efficient in MANETs. Any node, in a MANET consisting of N nodes, can act as a source. MANET can be modeled as an acyclic directed graph represented by $G(V, E)$ where V denotes the set of nodes and E the set of links of unit capacity, that is, transmitting one packet per link use. Let $\Gamma^-(v)$ be the set of terminating links at v whereas let $\Gamma^+(v)$ be the set of originating links from v. We assume that each link $e \in E$ has the capacity of one packet per unit time and $\mathbf{y}(e)$ is the packet carried on it. Here a packet is defined as a row vector of n elements from finite field $\mathbb{F}_q$. A unique source s sends a series of packets $\mathbf{x}_\mathbf{i}, \ldots, \mathbf{x}_h$ to a set of sinks $T \subset V$. Every source packet $\mathbf{x}_i$ is a row vector of length n over a finite field $\mathbb{F}_q$ and can be divided into two parts $[\mathbf{a}_i \ \ \mathbf{m}_i]$: the header vector $\mathbf{a}_i$ consists of h elements and the message vector consists of $n - h$ elements. Initially, the header vector $\mathbf{a}_i$ in the packet $\mathbf{x}_i$ is just the unit vector $\mathbf{u}_i$ of length h.

The vector matrix of source packet is defined as $\mathbf{X} = (\mathbf{x}_i^T, \ldots, \mathbf{x}_h^T)^T$, where $\mathbf{X}$ has all packets of the source as its rows. Let $\Gamma^-(s)$ denote the set consisting of h imaginary links $\widetilde{e}_1, \ldots, \widetilde{e}_h$ with $\mathbf{y}(\widetilde{e}_i) = \mathbf{x}_\mathbf{i}$. For any $e \in \Gamma^+(v)$, $v \notin T$, linearly combining the incoming packets of v, and $y(e)$ is calculated as illustrated in

$$\mathbf{y}(e) = \sum_{e' \in \Gamma^-(v)} \beta_{e'}(e)\, y(e') = \beta(e)\left[\mathbf{y}^T(e')\right]^T_{e' \in \Gamma^-(v)}. \quad (2)$$

In this equation, $\beta_{e'}$ are from $\mathbb{F}_q$ and $\beta(e) = [\beta_{e'}]_{e' \in \Gamma^-(v)}$ which is called LEV (Local Encoding Vector) of the link e. By induction, $\mathbf{y}(e)$ is represented as the linear combination of source packets.

$$\mathbf{y}(e) = \sum_{i=1}^{n} g_i(e)\, x_i = \mathbf{g}(e)\,\mathbf{X}. \quad (3)$$

Equation (3) elucidates that $\mathbf{g}(e) = [g_1(e), \ldots, g_m(e)]$ that can be recursively calculated using (2). This is named as GEV (global encoding vector) of link e.

Assume that h packets $\mathbf{y}(e_1), \ldots, \mathbf{y}(e_h)$ are received by a sink node v from links $e_1, \ldots, e_h$ incoming to v. Then by (3) we have the packets $\mathbf{Y}$ received by v as follows:

$$\mathbf{Y} = \begin{bmatrix} y(e_1) \\ \vdots \\ y(e_m) \end{bmatrix} = \begin{bmatrix} g(e_1) \\ \vdots \\ g(e_m) \end{bmatrix} \mathbf{X} = \mathbf{GX}, \quad (4)$$

where $\mathbf{G}$ is referred to as global encoding matrix (GEM) of v. When $\mathbf{G}$ is invertible, then the source packets $\mathbf{X}$ can be reconstructed by using $\mathbf{X} = \mathbf{G}^{-1}\mathbf{Y}$.

Table 1: Notations for PPE.

Symbol	Explanation
$\mathbf{x}_i$	Source packets
$\mathbf{X}$	Vector matrix of source packets
h	Number of messages, permutation length
$\mathbf{y}(e)$	Coded packet carried on link e
$\beta(e)$	LEV of link e
$\mathbf{g}(e)$	GEV of link e
$\mathbf{G}$	Global Encoding Matrix
$\mathbf{a}$	Sequence of GEVs
k	PEF key
c	Ciphertext
D	Data generations
$f(h)$	Function used for confusion key
k'	Confusion key for each data generation

4. Partial Permutation Based Encryption (PPE) Scheme

The main idea of the proposed scheme, called PPE, is that only GEVs are permuted instead of the whole packets at the source. This makes sufficient confusion for an adversary to locate GEVs in order to get meaningful information. As we are permuting only GEVs, this significantly reduces the complexity and making the scheme more efficient. On the other hand, as the header length may not be long enough for permutation to achieve sufficient security, we additionally propose using random encryption key on message to further increase the security of the proposed scheme against adversarial attacks. A list of notations to be used to introduce the PPE scheme is provided in Table 1.

Figure 4 briefly illustrates the PPE scheme. Initially, the source has h original messages $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_h$ of length $n - h$, each of which is padded with a unit vector $\mathbf{u}_j$ as the header. Then, it performs linear combinations on the packets to generate h linearly independent packets. Subsequently, the PPE scheme will conduct encryption on the h packets. First, GEVs are permuted based on some permutation key k and then the message is encrypted using dynamic random key encryption based on a key k', whose generation will be discussed in detail in the following subsections.

A typical MANET scenario involves a source node, intermediate nodes, and sink nodes. Figure 5 depicts the data transmission in a MANET based on the proposed PPE scheme and network coding and the PPE scheme is only performed at the source nodes for encryption and at the sink nodes for decryption, and the intermediate nodes perform recoding of the message packets.

4.1. Dynamic Random Key Generating Algorithm. In a scenario where source needs to transmit huge data volume

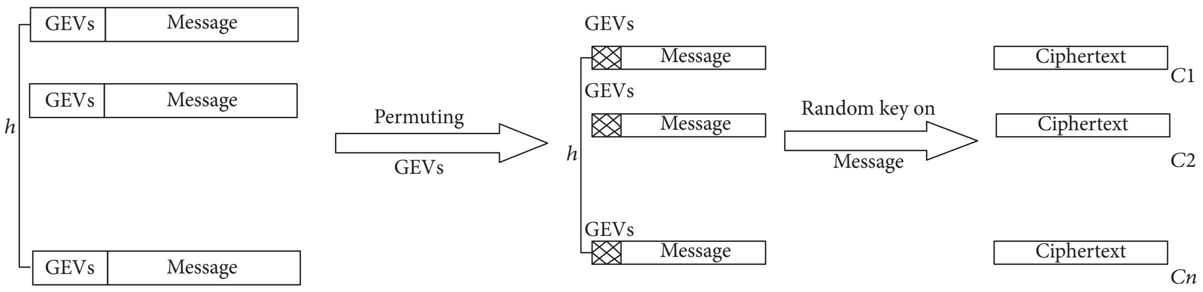


Figure 4: PPE scheme on coded message.

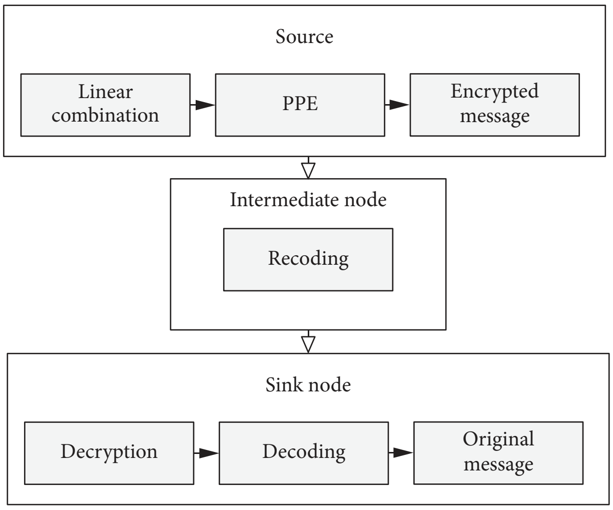


Figure 5: Stages of data transmission in a MANET based on the PPE scheme and network coding.

from one node to other nodes, source should first divide data into generations and use perturbing key on each data generation. If a single perturbing key is used in the course of transmission then there is a chance that this key would be disclosed, which will result in compromising security of the whole data volume. This is known as single generation failure. Assume the following steps are performed by source on ith data generation D_{Gi}, and key k_i is generated.

(a) Randomly choose h positions among data generation D_{Gi} which are known as perturbing key.

(b) Corresponding to data generation D_{Gi}, key k_i is calculated according to Algorithm 2.

(c) Encrypt D_{Gi} based on k_i and the encrypted data generation is sent from source node to all participant nodes that can update key.

Algorithm 1 explains the perturbing function for GEVs and is also used to generate dynamic random key.

```
(1)  Array Key[] /* size m */
(2)  Function Key_Gen (integer m)
(3)  Initialization(m)
(4)     For i ← 1 to m − 1
(5)         ψ ← rand() /* between i to m */
(6)         Key[i] ← perm(ψ)
(7)     End for
(8)  Function Initialization (integer q)
(9)  For a ← 1 to q
(10) Key[a] ← a
(11) End for
```

Algorithm 1: Key generation.

Step (1)–(3). Key size m is declared. Function *key_Gen* gets the key size as its argument. Initialization function is used to call key size m.

Step (4)–(7). A random value between i and m is stored in ψ. The permutation function takes this random value as a seed to generate a value as $Key[i]$. The loop ends on $m-1$.

Step (8)–(11). Initialization function has values from a to q where a is stored as $key[a]$ and loop ends at q.

In Algorithm 2, we use symmetric encryption for secured transmission of the secret key from source node to sink. We propose partial permutation data generation key. In this scenario, instead of using the same generation data stream, some data elements remain at their original position. In the proposed scheme, source randomly selects the length of data packet and makes a generation. On the basis of shared random value for partial permutation and confusion key, both ends have the ability to generate common session key on a distributed manner. In this scenario, we assume that secret parameter has been distributed by a trusted authority between source and sink as illustrated in Figure 6. To guard against replay attacks, we have used dynamic random key generation. Dynamic keys are the cryptographic keys that are not the same for the whole network lifetime. Instead, it is established either periodically or on demand. This helps increase the network survivability by revoking keys of the compromised nodes in the process of rekeying.

```
(1) Set $s = \text{rand}$, $h = 0$, $\gamma = D/\omega$, $d = (0, h! - 1)$
(2) $D = \{D_{G_1}, D_{G_2}, D_{G_3}, \ldots, D_{G_\gamma}\}$
(3) $k(i) = \{D_{k_1}, D_{k_2}, D_{k_3}, \ldots, D_{k_\gamma}\}$
(4) Loop from $i = 1$ to $\gamma$
        (i) $g(i) = d\%/(i + 1)$;
        (ii) $d = d/(i + 1)$
        (iii) $g = \omega i/h$
(5) $CK[i] = Confusion\_Key\ (g, d, i)$
(6) $Cipher(Ci) = Encr(D_{G_i}, CK[i])$
(7) Send $Ci$
(8) EndLoop
Function Confusion_Key (int $r_n$, int $d$, int $index$) BEGIN
(9) $b(index) = D - d - (f(d/h)/(index + 1))$
(10) For each $(index \in [1, h])$
(11) do $\theta(s - 1 + index) \leftrightarrow \theta(s - 1 + b(index))$
(12) $(k'(index)) \leftarrow \theta(k(index))$
(13) Return $k'(index) \leftarrow (k'(index) \parallel r_n)$
(14) END
```

Algorithm 2: Random permutation based keying.

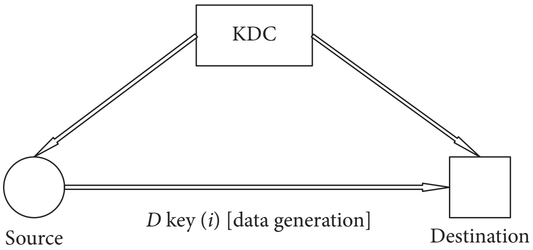


Figure 6: Dynamic key generation by a Key Distribution Center (KDC).

Dynamic Source Routing (DSR) protocol is simple and efficient routing mechanism designed typically for multihop MANETs. It makes the network completely self-configuring and self-organizing without requiring any existing infrastructure. As DSR is source initiated link state routing the nodes dynamically discover an efficient route to send the packet by multiple network hopping till the destination which makes it be the loop-free packet routing. The DSR protocol consists of two mechanisms: one is route discovery and the other is route maintenance. Based on this scenario, we adopt dynamic key for each data generation (D_G) to handle replay attack in MANET. Based on the sharing parameters at both ends, source generates dynamic key for each generation. Suppose we have key (ki) then the $D_{key(i)} = D_{key(1)}, D_{key(2)}, \ldots, D_{key(\gamma)}$ where (D_γ) represents the number of dynamic keys distributed for each generation and $D_{key(i)}$ is dynamic key.

As discussed in [15], traditional cryptographic technique like AES for end-to-end encryption cannot be used due to the limited resource capabilities of MANETs. This work contributes to generate lightweight encryption key as shown in Algorithm 2, which introduces a random number and updating key for each data generation to enhance security and reduce computation and communication overheads from source to destination in MANETs.

Steps (1)–(3). Set random number n used to generate the value of h for permutation. In this step, γ represents the number of generations and D is the total data length that is divided into generations like D_{G_1} and D_{G_2} where ω is chunk size to produce a single generation. The value of h can vary from 4 bits to 32 bits where $h \ll \omega$ and it is used for permutation. Moreover, $K(i)$ represents the set of keys where D_{k_1} represents key for the first generation. Other parameters are initialization factor $K(h)$, packet length (n), and random division of perturb-vectors ($d = (0, h! - 1)$).

Steps (4)–(8). Loop for generation from 1 to γ where $g(i)$ is the index of generation and calculated as chunk over permutation length h is used in Steps (5) and (6) for calculating confusion key and ciphertext which is generated as a function of data generation and confusion key. After the loop ends we will get ciphertext.

Steps (9)-(10). Confusion key function $b(index)$ is calculated by subtracting $f(d/h)/(index + 1)$ from random division of perturbed-vectors d and again from the data length D, where indexing is from 1 to the permutation length h.

Steps (11)–(14). Confusion key $k'(index)$ is generated for an end-to-end communication which consists of a perturbing key and a random number. The perturbing function θ is calculated by subtracting the sum of 1 and indexed confusion function, which will continue till the end of the permutation length.

5. Performance Analysis

We take into account typical cryptanalysis on permutation cipher which is a case of transposition cipher and evaluate how effectively a permutation encryption can be broken. This is based on nonuniform occurrences of *n-letter* combinations named as *n-gram*. Taking an example, the frequency of bigram "*TH*" in English is much higher as compared to bigram "*QZ*." The ability of guessing permutation π is accessed by using *n-grams* frequency statistics: first large cipher texts are decrypted by using inverse of permutation π and then evaluated on the basis of how close the statistics of *n-grams* decrypted messages are as compared to statistics of underlying languages. Then we find the permutation of other letters that have better ability by searching "*ps*" neighborhood with good ability until key k is found. Although this is rather effective in transposition ciphers breaking, we contend that for the case of our permutation encryption it does not work fine. (1) First we permuted GEVs; then we introduced dynamic random encryption key on message. This double encryption makes it much stronger against replay attack. So this means that, as compared with transposition cipher, the proposed scheme requires a lot of time to access the ability of permutation. (2) A small change of permutation, for example, change of just two positions, will give different

Table 2: Simulation parameters.

Simulation time	150 sec
Terrain area	$2000 \times 2000\,\text{m}^2$
Number of nodes	250
Node placement strategy	Random
Propagation model	Two-ray model
Mobility model	Random
Routing protocol	AODV
Traffic type	Constant bit rate (CBR)
Pause time	0 s

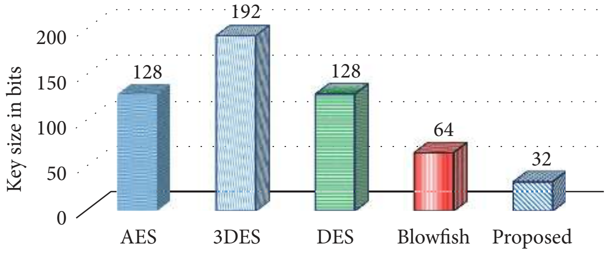


Figure 7: Key size of the storage analysis.

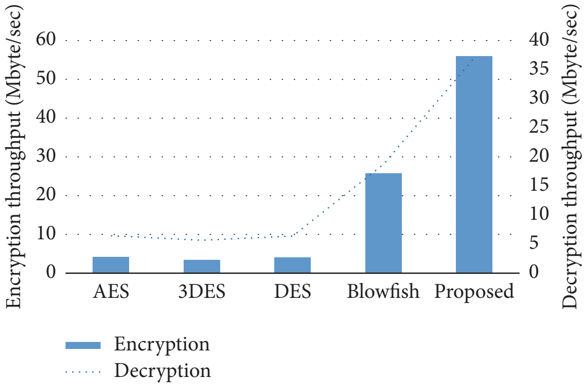


Figure 8: Encryption/decryption throughputs in Mbytes/sec.

encryption that would decode information into totally different messages. This resulted in the fact that permutation has good ability so by searching in the neighborhood of the permutation we do not expect to get permutations with better ability. As dynamic random encryption key is distributed at initial stage, permutation operation (encryption at source and decryption at destination) will give less computational overhead. So permutation encryption is lightweight in terms of computation.

The performance analysis of the proposed scheme is analyzed in built-in classes of Java. It uses managed packaging for AES, DES, and Blowfish which is available in http://java.cypto.com. The existing schemes used traditional cryptographic methods of cipher class that provide functionality of different cryptographic techniques used for encrypting and decrypting of data. It forms the core of the JCE framework. Simulation parameters are given in Table 2.

5.1. Storage Analysis. The proposed scheme is based on secret credentials to encrypt perturbing methods and acts as secure shared session key between end nodes. The pseudo random number generator (PRNG), a confusion function (f), permuting of selective data ($f(d/h)$) portion, and time stamp (T) are used as secret credentials. The memory overhead of the confusion key depends on the selective bits of data and other parameters. To overcome memory and computation overhead, the proposed scheme uses 32-bit key size which has lesser overhead from traditional cryptographic techniques like DES, AES, 3DES, and Blowfish. The application payload is 32 bits, which include 16 bits of session key and 16 bits of the random challenge. The performance analysis of Algorithms 1 and 2 shows the proposed scheme is efficient from traditional schemes and suitable for resource constrains devices. Performance comparison of different schemes is done on the basis of parameters like encryption time, encryption/decryption throughput, and energy consumption as shown in Table 3. We considered well known symmetric key encryption techniques for comparison with the proposed scheme. Data encryption standard (DES) has 128-bit key size and 64-bit block size. 3DES, which is an enhancement of DES, uses three 64-bit keys which makes 192-bit key size and has a 64-bit block size as well. AES has variable key lengths of 128, 192, or 256 bits and has a data block size of 128 bits. We consider 128-bit key. Finally Blowfish uses variable key lengths of 32 bits to 448 bits and 64-bit data block size. We used 64-bit key size.

The appraisal is intended to assess the results by using block ciphers. Consequently, the load data (plaintext) is divided into smaller data generation size as per algorithm settings given in Table 3. De Meulenaer et al. [32] evaluated energy of wireless nodes in terms of communication cost, whereas Abdul Elminaam et al. [33] evaluated various symmetric cryptography algorithms used in MANETs in terms of energy cost. The energy is calculated by using the following equation: $E = P \times t$, where E is the energy in joules, P is the nominal power in watts, and t is the time duration in seconds. Operation cost and transmission of 1 byte is 5.76 μJ, reception of 1 byte is 6.48 μJ, and AES-128 encryption of 16 bytes is 42.88 μJ. On the basis of these assumptions we have computed energy consumption at processing and transmission level of proposed scheme. In addition, we analyzed the key size as in Figure 7.

5.2. Throughput. As the encryption time decreases, more data can be processed which results in larger throughput. Encryption time has to be considered while calculating throughput of an encryption algorithm. The throughput of an algorithm is obtained by dividing plain text in kilobytes by encryption time in milliseconds. From the graphical illustration of encryption and decryption throughput in Figure 8, the proposed scheme outperforms other schemes in terms of throughput. The gap between the proposed scheme and AES, DES, and 3DES is significantly large because of their higher

Table 3: Performance comparison of schemes.

Algorithms	Key size (Bit)	Data block (bit)	Encryption time (milliseconds)	Encryption throughput Mbyte/sec	Decryption throughput (Mbyte/sec)
AES	128	128	374	4.17	6.452
3DES	192	64	452	3.45	5.665
DES	128	64	389	4.01	6.347
Blowfish	64	64	60.3	25.8	18.7
Proposed scheme	32	32	31	56	37

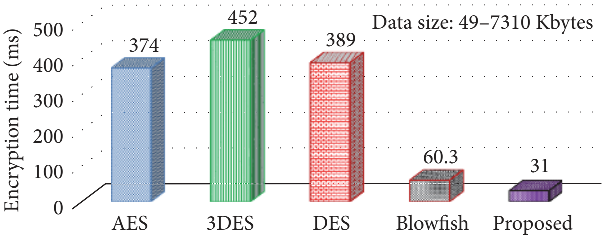


Figure 9: Data encryption time in milliseconds.

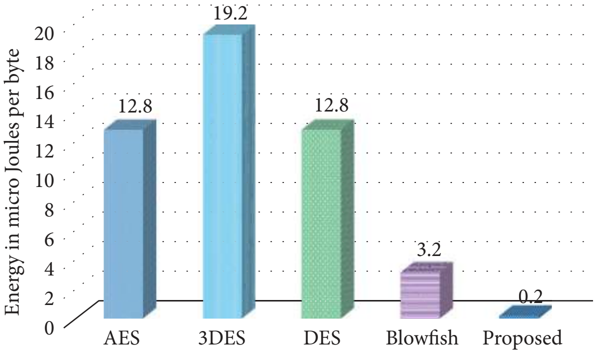


Figure 10: Energy consumption in μJ.

encryption time. Data throughput on encryption and decryption is depicted in graphical presentation. The throughput of the proposed scheme is calculated and then compared with other schemes according to the formula throughput = plain text size/encryption time. We have calculated the encryption and decryption time of various algorithms as per the throughput formula. We considered plaintext size to be the average of plain text size and the average time taken. So 1598.7 kbytes/374 ms = 4.174 Mbytes/s for AES while for our proposed scheme it is calculated as 1598.7 kbytes/31 ms = 56 Mbytes/s. In the same way, the decryption throughput is calculated via dividing the plain text size by decryption time.

5.3. Encryption Time. Encryption time is calculated as time taken by any device in executing the encryption algorithm. Encryption time is calculated by taking into account the size of plain text, key size, and block size. We consider the algorithm key sizes as given in Table 3. From Figure 9 we can clearly see that the proposed scheme has very less encryption time as compared with Blowfish algorithm which is a light weight encryption scheme. From Table 3 we can clearly see that when the key size increases, time for encryption increases as well. As for the proposed scheme the key size as well as the block size is smaller than the other algorithms, that is, 32 bits; the encryption time comes out to be 31 ms, which is significantly lower than other schemes.

5.4. Energy Consumption. The energy consumption is calculated as the average energy consumed during the process by an algorithm. The cost of the proposed scheme is based on energy consumption during encoding, transmitting, and receiving cost of data at sinks. On the basis of [32] we elaborate energy consumption of the proposed scheme and compare it with the existing techniques. Less time taken for encryption means there are few cycles which gives lower energy consumption. As described before, the proposed scheme has less encryption time as compared with other encryption schemes. As shown in Figure 10, the energy consumption of the proposed scheme comes out to be 0.2 μJ which is significantly lower as compared with other schemes. The energy consumption is calculated by considering block size in bytes, and multiplying it with energy consumption for symmetric key comes out to be 0.8 μJ/bit. As described in the scheme, confusion key plays a vital role in the proposed scheme in which secret credentials are used for the session key. A node has to compute 16-bit PRNG, 8-bit time stamp, and 8-bit permuting of selective data to secure perturbing function. On the assumption of [32] the computation cost is 0.25 μJ/bit and our proposed scheme needs to compute 32-bit data for confusion key which means total energy is 32 × 0.25 = 8.0 μJ. In this scheme the confusion key computation cost depends on the random amount of selective data from n-bit data length. We assume that the proposed scheme computes on 32-bit block generation of data.

6. Conclusion

In this paper, we proposed a partial permutation encryption algorithm for network coded MANETs. Instead of permuting the whole packet as in the previous P-Coding scheme, the proposed scheme permutes only GEVs which decreases the computational complexity, making it an efficient encryption scheme in terms of energy, computation, and cost. To guarantee that the proposed scheme is secure against various attacks, we proposed dynamic key generation mechanism for our random key generation. We analyzed the proposed scheme by taking into account different parameters and concluded that our partial permutation scheme is efficient and lightweight. The proposed scheme outperforms other analyzed schemes in terms of efficiency and cost. The proposed scheme has lower encryption time and greater throughput which resulted in 117% improvement from the Blowfish algorithm for MANETs. Blowfish algorithm has 5 times greater throughput than DES. The proposed scheme has 16 times lesser energy consumption than the Blowfish algorithm that makes it an efficient encryption technique for energy constraint devices.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61471034 and by the Ministry of Education of China under Grant no. 6141A02033307.

References

[1] L. Gruenwald, M. Javed, and M. Gu, "Energy—efficient data broadcasting in mobile ad hoc networks," in *Proceedings of the International Database Engineering and Applications Symposium (IDEAS '02)*, July 2002.

[2] L. Junhai, X. Liu, and Y. Danxia, "Research on multicast routing protocols for mobile ad-hoc networks," *Computer Networks*, vol. 52, no. 5, pp. 988–997, 2008.

[3] S. Giordano and W. W. Lu, "Challenges in mobile ad hoc networking," *IEEE Communications Magazine*, vol. 39, no. 6, p. 129, 2001.

[4] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003.

[5] J. Loo, J. Lloret, and J. H. Ortiz, "Mobile Ad Hoc Networks: Current Status and Future Trends," 2011.

[6] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[7] S. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[8] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.

[9] L. Chen, T. Ho, M. Chiang, S. H. Low, and J. C. Doyle, "Congestion control for multicast flows with network coding," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5908–5921, 2012.

[10] D. Annapurna, N. Tejas, K. B. Raja, K. R. Venugopal, and L. M. Patnaik, "An energy efficient multicast algorithm for an Adhoc network using network coding and MAC scheduling," in *Proceedings of the International Conference on Signal Processing and Communication (ICSC '13)*, pp. 62–67, December 2013.

[11] E. Altman, L. Sassatelli, and F. De Pellegrini, "Dynamic control of coding for progressive packet arrivals in DTNs," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 725–735, 2013.

[12] A. M. Sheikh, A. Fiandrotti, and E. Magli, "Distributed scheduling for scalable P2P video streaming with network coding," in *Proceedings of the 32nd IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 11–12, April 2013.

[13] R. R. Rout and S. K. Ghosh, "Enhancement of lifetime using duty cycle and network coding in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 656–667, 2013.

[14] Q. Yan, M. Li, Z. Yang, W. Lou, and H. Zhai, "Throughput analysis of cooperative mobile content distribution in vehicular network using symbol level network coding," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 484–492, 2012.

[15] P. Zhang, C. Lin, Y. Jiang, Y. Fan, and X. Shen, "A lightweight encryption scheme for network-coded mobile Ad Hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2211–2221, 2014.

[16] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906–1918, 2005.

[17] M. Čagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, pp. 172–182, September 2002.

[18] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–11, April 2006.

[19] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network coding-based broadcast in mobile ad hoc networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM '07)*, pp. 1739–1747, May 2007.

[20] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *Proceedings of the Workshop on Network Coding, Theory, and Applications (NetCod '05)*, Riva del Garda, Italy, 2005.

[21] L. Lima, M. Medard, and J. Barros, "Random linear network coding: a free cipher?" in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 546–550, Nice, France, June 2007.

[22] J. Wang, J. Wang, K. Lu, B. Xiao, and N. Gu, "Optimal linear network coding design for secure unicast with multiple streams," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, IEEE, San Diego, Calif, USA, March 2010.

[23] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 2083–2091, April 2008.

[24] J. P. Vilela, L. Lima, and J. Barros, "Lightweight security for network coding," in *Proceedings of the IEEE International*

Conference on Communications (ICC '08), pp. 1750–1754, May 2008.

[25] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 2213–2221, Rio de Janeiro, Brazil, April 2009.

[26] J. Benaloh, "Dense probabilistic encryption," in *Proceedings of the Workshop on Selected Areas in Cryptography*, pp. 120–128, August 1994.

[27] Y. Wei, Z. Yu, and Y. Guan, "Efficient weakly-secure network coding schemes against wiretapping attacks," in *Proceedings of the IEEE International Symposium on Network Coding (NetCod '10)*, pp. 1–6, IEEE, Ontario, Canada, June 2010.

[28] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)*, pp. 71–83, November 2008.

[29] S. Tan, X. Li, and Q. Dong, "Trust based routing mechanism for securing OSLR-based MANET," *Ad Hoc Networks*, vol. 30, pp. 84–98, 2015.

[30] U. Venkanna, J. K. Agarwal, and R. L. Velusamy, "A cooperative routing for MANET based on distributed trust and energy management," *Wireless Personal Communications*, vol. 81, no. 3, pp. 961–979, 2015.

[31] M. Takeuchi, E. Kohno, T. Ohta, and Y. Kakuda, "Improving assurance of a sustainable route-split MANET routing by adapting node battery exhaustion," *Telecommunication Systems*, vol. 54, no. 1, pp. 35–45, 2013.

[32] G. De Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob '08)*, pp. 580–585, October 2008.

[33] D. S. Abdul Elminaam, H. M. Abdul Kader, and M. M. Hadhoud, "Performance evaluation of symmetric encryption algorithms," *Communications of the IBIMA*, vol. 8, pp. 54–64, 2009.

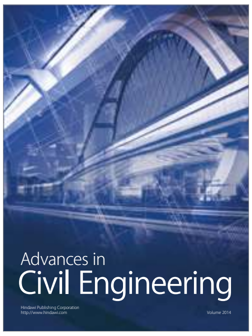
Journal of
Engineering

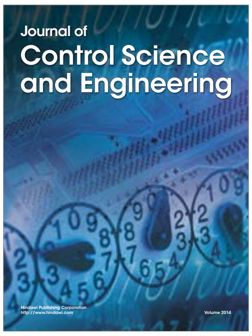
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

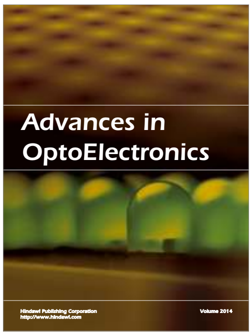
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

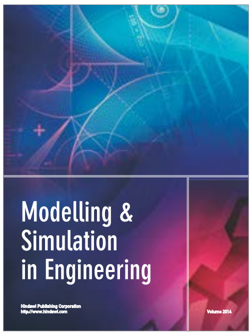
Journal of
Robotics

Journal of
Electrical and Computer Engineering


Advances in
OptoElectronics

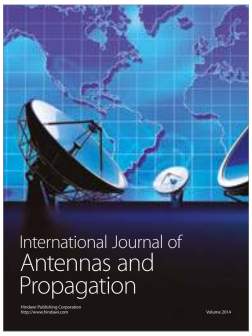
VLSI Design

International Journal of
Navigation and Observation

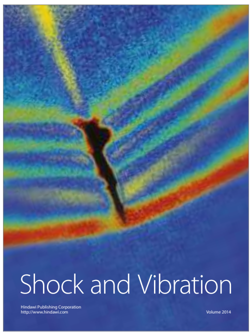

Modelling & Simulation in Engineering

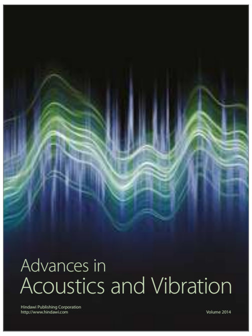
International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration