

# An Energy Efficient Crypto Suit for Secure Underwater Sensor Communication using Genetic Algorithm

Fozia Hanif<sup>1</sup>, Urooj Waheed<sup>2</sup>, Samia Masood<sup>2</sup>, Rehan Shams<sup>3</sup>, Syed Inayatullah<sup>1</sup>

---

## Abstract:

*With the advancement in technology, there has been a keen interest of researchers and industrial institutions in the use of Underwater Sensors Networks (UWSNs). This study is devoted to the secure communication between the underwater sensors networks which are nowadays most widely used for oceanographic abnormalities, and to track submarines that perform the surveillance and navigation. But UWSNs has its limitations such as multipath, propagation delay, low bandwidth, and limited battery as compared to traditional WSNs that causes a low life in comparison with WSNs. Secure communication in UWSNs is more difficult due to the above-mentioned limitations which need ultralightweight components. There are many miscellaneous attacks due to which sensors can lose both data availability and integrity. In this study we have designed a computation and space efficient algorithm for secure underwater sensor communication. The proposed algorithm will generate two-halves of the key through a genetic algorithm (GA). Genetic algorithm is an evolutionary technique, that produces very good results in many engineering problems. In cryptography, the most important part is the key generation procedure that plays a major role in data transfer. The secure key is the basic requirement of data encryption and by the help of GA, this study provides a complex key generation procedure for one part of the key. Genetic algorithm includes some basic steps such as initial population generation, crossover, and mutation. However, a new fitness function is introduced to enhance the efficiency of GA along with the different procedures of crossover and mutation. After that encryption algorithm is proposed for the secure communication between UWSNs and its performance is evaluated based on throughput, running time, space usage, and avalanche effect.*

**Keywords:** *UWSNs; Security; Cryptography; Genetic Algorithm; Linear congruential procedure; pseudo-random number; avalanche effect.*

---

## 1. Introduction

Wireless sensor networks have wide applications in fields such as home, industry, environmental observation, military monitoring, and disaster relief [1]. Recent advances in wireless communications and electronics have enabled the development of

small low-cost sensor nodes that communicate over short distances. Wireless sensor networks are comprised of several sensor nodes that communicate via wireless technology.

In this paper, we will propose a new way of symmetric cryptography for sending the data between underwater sensor networks with

---

<sup>1</sup> Department of Mathematics, University of Karachi, Karachi, Pakistan.

<sup>2</sup> Department of Computer Science, DHA Suffa University, Karachi, Pakistan.

<sup>3</sup> Department of Telecommunication Engineering, Sir Syed University of Engineering and Technology, Karachi, Pakistan.

Corresponding Author: [ms\\_khans2011@hotmail.com](mailto:ms_khans2011@hotmail.com)

high security. In symmetric cryptography, there is only one key which is responsible for both the encryption and decryption. Therefore, the key generation procedure should be very complex to generate a strong key to stop any intruders from guessing or detecting the key. The key generation procedure behaves as a backbone of any cryptographic algorithm; therefore, this study is going to use GA for generating the key for the encryption procedure. Cryptography has always been a most important requirement in the IoT application but as the mode of communication changes the requirement of security changes as well but limitations in underwater sensors are more as compared to other ways of communication [3-4]. Some features of WSN and UWSNs are the same due to the but harsh environment of UWSNs there are more constrained in UWSNs as compared to WSN such as unreliable communication channels, dynamic networks topology, insecure environment, and vulnerability [5].

The proposed algorithm generates its key which is of 128 bits in two steps: the first half of the key will be calculated by the anchor node through Genetic Algorithm (GA) procedure and the rest of part of the key is calculated by the sensor node by using some other procedure and merger of these two parts will be the final key [6, 7]. To avoid the passive attack here the data frame will be sent through some authentication code to avoid the attack and after receiving the frame sensor will simulate the code by itself to match with the incoming code after this matching of authentication code the data frame can open by receiving sensor otherwise it will discard the data frame.

The proposed research scheme will calculate the two parts of the key separately, authentication codes, and then it performs the process of encryption and decryption. The novelty of this study is that GA has never been used in data communication of UWSNs and our result session will prove that how GA will give more randomness to the key generation procedure and this complete lightweight process will not only enhance the security but also provides low computational complexity

[8-9]. This paper is organizing as follows: After the introduction section 1.1 gives the literature review, section 2 indicates the security issues in underwater sensors communication, section 3 discusses Genetic algorithm along with a brief discussion of its steps, after that this paper gives key generation procedure through GA with details steps implementation then section 4 gives the calculation of authentication code and final key formation, section 5 shows the encryption for underwater communication and decryption procedure also. Section 5 indicates an analysis of the result by using different parameters and shows the security analysis and in the end, we have the conclusion and references.

## 2. Literature Review

Cryptography has been a major requirement for many years for any type of communication system, but the cryptographic algorithm is dependent upon the environment through which its communication occurs. Many researchers have made their efforts to perform underwater communication but due to the limitations of the underwater environment, it is not so easy to perform underwater communication smoothly [10,11].

Several GA-based algorithms have been made for secure cryptography [12,13], also many researchers have proved that the performance of GA produced better results [14] but for underwater sensor communication, GA has never been used before.

Soniya Goyat in [15] says that if the quality of the random numbers produced by the method is good then the key generation is always better. Ultra-Lightweight cryptography has been presented for underwater sensor networks that replace the S-box with 8 round iteration block cipher algorithms [16]. The effort of modification of RC6 has also been made by [17]. Due to the computational complexity symmetric cryptography gives better results in underwater sensors as compared to asymmetric cryptographic algorithm [18, 19].

Another improvisation for secure communication in underwater sensors was presented by [20], which deals with XOR, left-right shift for the lower computational cost and generates the random key by pseudo-random number generator that reduces the space storage. To reduce the computational burden [21] has also presented secure underwater communications based on fully hashed MQV. Gove Nitin Kumar Kaur in [22] uses the concept of brain Mu waves, genetic algorithms, and pseudorandom binary sequence. Faiyaz Ahmad has proposed a model that makes use of GA to generate Pseudo-random numbers [23].

The literature review shows that although a lot of efforts have been made to improve the security of UWSNs through different encryption algorithms, most of them are both space and complexity expensive. The need to design a scheme that utilizes the minimal space and computational capacities of the underwater sensors while providing a completely secure and efficient communication still exists. The proposed model addresses all these issues and provides a secure way of communication using minimal sensor resources.

### **3. Security Issues in Underwater Sensors Communication**

Many applications are associated with underwater environments such as surveillance, ocean monitoring, and disaster mitigation to measure the level of the sea due to the melting process of the ice sheet. All this can be possible due to randomly placed underwater sensors that collect some important hydrologic data for example pressure, temperature, and salinity. The most important task which is performed by underwater sensors is to sense the data and pass it to the relevant base station, but security threats are the major issue while transferring the data [24-26].

There are many constraints during underwater communication. UNSNs can be easily affected by various attacks and malicious threats, these attacks can be either active or passive [27]. A passive attack is an attempt by miscellaneous nodes to obtain the

transmitted data without changing the operation that's why it is very difficult to detect. Whereas attractive attacks are easy to indicate, and it tries to delete, alter, distract the transmitted data in the network. The active attack mostly attempts by external nodes which do not belong to the networks. The main feature of security in UWSNs are key management, intrusion management, trust issues, secure localization, secure synchronization, and routing security [28, 29]. To achieve the security requirements and setup or mechanism should be proposed that protect UWSNs from these above-mentioned threats [30].

The main goals of cryptography are repudiation, integrity, confidentiality, and authentication. The encryption schema should satisfy challenges of underwater such that, it should be adaptable for underwater transmission, lower computation with less overhead, cost, and energy-efficient and ensure high security. The main features of any encryption algorithm of UWSNs are to provide integrity and confidentiality in between nodes using less space and high security with lower computations [31,32].

In this study, we develop an algorithm that not only generates the key using the Genetic Algorithm but also provides an encryption scheme for secure communication that proves to be efficient in terms of space usage, running processing time, and the avalanche effect.

#### **3.1. System Architecture**

The underwater environment consists of various sensor nodes that can be able to communicate with each other. The communication between underwater sensor nodes is done with the help of a sink node and a base station which route the data from one sensor to another. For the proposed architecture it is assumed that all sensors can sense, communicate, and be able to calculate. It is a well-known fact that routing the data between underwater sensor nodes is not that much easier as compared to traditional Wireless Sensor Networks (WSNs), because of the continuous movement of the sensors in the ocean. The main purpose of underwater

sensors is to sense the data from the environment and route them between the nodes. The figure 1 shows the UWSNs environment

where different nodes such as sink node, sensor node and the base station.

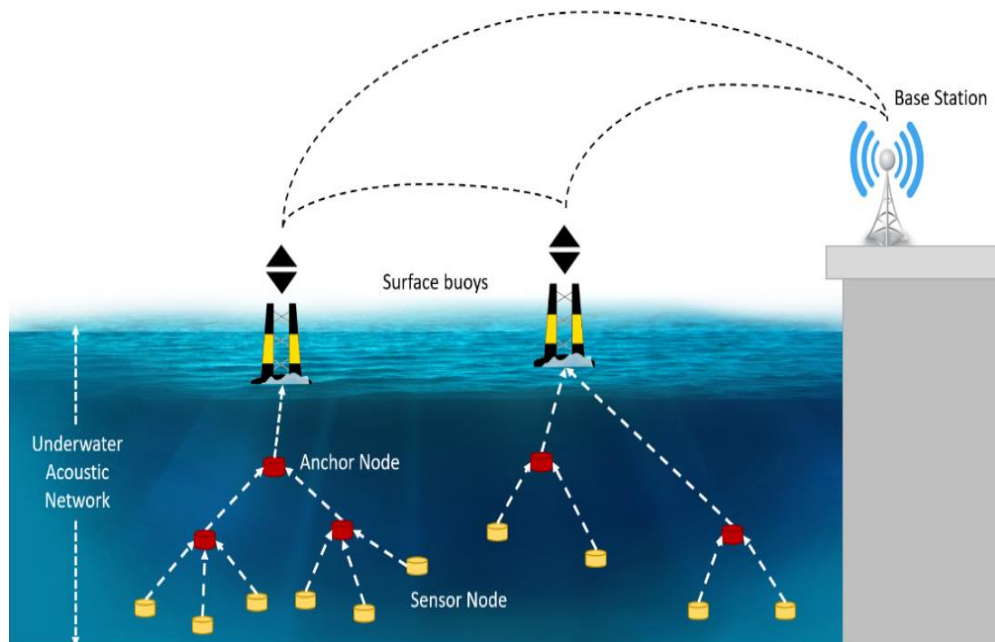


Fig. 1. System architecture for underwater acoustic networks

#### 4. Genetic Algorithm

A genetic algorithm is an evolutionary procedure that is basically used to optimize many problems like shortest path, intrusion in WSN, bandwidth utilization, and many more [33, 34]. The reason behind using the genetic algorithm in generating the key in UWSNs is, that cryptography through GA provides the lightweight complexity which is the measure requirement within the UWSNs. GA approach is completely random which enhances the cryptographic encryption and decryption, also the elitism Genetic algorithm starts with the random results called chromosomes which can be generated through many random procedures, is considered as the results of the given problem [35]. Furtherly these randomly generated results can be made more accurate by using different steps of genetic procedure which are fitness measure, crossover, and mutation. To get more accurate results

through GA it is very important to have a strong fitness function that applies on initial random generation to measure its fitness. The fitness function decides which chromosome can go for the process of crossover. In crossover two chromosomes will produce two more fitted chromosomes that can be tested again, by using a fitness function. After getting better chromosomes from crossover, we apply mutation to achieve global optima from local optima [36].

In the proposed algorithm we have used the above-mentioned steps of genetic algorithm to generate the half part of the key for symmetric cryptography. These traditional steps of GA do have many variations according to a scenario and environment [37]. We have performed these steps in our own way by making the fitness function according to the suitable parameters that are related to the conditions of the cryptographic approach.

#### 4.1. Key Generation Procedure

The cryptographic algorithm starts with the process of key generation; here we are generating the half part of 64 bits key with the help of a genetic algorithm (GA) which is an evolutionary-based procedure. The reason behind choosing GA for the key generation is, that it is completely a random procedure which makes key guessing very difficult, and to make this even more difficult the remaining half part of 64 bits will be generated through some other procedure and combination of both parts will be used for the application of encryption and decryption [38].

#### 4.2. Steps of Proposed Genetic Algorithm

The genetic algorithm is mainly consisting of some traditional steps: initial random population generation, crossover, fitness function, and mutation. Although the procedure to perform these operations are different in each genetic algorithm but the basic steps are the same in all GA. In the next section, we explain the performance of each step of GA in detail.

##### 4.2.1. Initial Random Population

**Generation:** As defined earlier that we are generating the 128 bits key for the proposed cryptographic algorithm and 64 bits key will be generated through GA which means we need to generate 64 bits random numbers as an initial population also called chromosomes in the genetic field. The process for generating the 64 bits binary random numbers is based on linear congregational procedure [39]. The detail of this procedure will be given in section 2.

**4.2.2. Crossover Procedure:** After performing step one, all the newly generated chromosomes will go under the presses of crossover with the help of pseudorandom number generator for 64 bits which will be discussed in 2.1, the resultant number obtained from this procedure will decide about the

crossover point. Crossover is of several types one point, two points, three points, and random point therefore the resultant number will decide the random point for the crossover operation and is given by figure 2 in which the bits after the selected point will be exchanged by both parent chromosomes to get two new resultant species.

Parent 1	0	0	1	1	0	1	1	1	0	0	0	1	0	1	1
Parent 2	0	1	1	1	0	1	0	0	1	1	0	0	1	1	1
Child 1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	1
Child 2	0	1	1	1	0	1	1	1	0	0	0	0	1	0	1

**Fig. 2.** Randomly selected one Point Cross Over and exchanging the bits after 6 random point.

**4.2.3. Fitness Function:** The fitness of all newly generated chromosomes will be checked by using the fitness function. This procedure can reduce the number of populations by the survival of fitness which means, only those species will exist which have the best fitness amongst all. The proposed Fitness function for the proposed algorithm is given by Eq. (1),

$$Fitness\ Function = 1 - \frac{SEC}{Gap\ Value} \quad (1)$$

where,

SEC = Shannon entropy of chromosome

In information theory, entropy is a measure of the uncertainty in a random variable. About this, the term Shannon entropy usually refers, to which quantifies the expected value of the information contained in a message (in classical informatics it is measured in bits).

Shannon entropy allowing to estimate the average minimum number of bits

needed to encode a string of symbols based on the alphabet size and the frequency of the symbols can be calculated by using the following formula,

$$H(X) = -\sum_{i=1}^n p(x_i) \log_b p(x_i) \quad (2)$$

In Eq. (2)  $p(x_i)$  is the lower probability, i.e.  $p(x_i) \rightarrow 0$ , the higher the uncertainty or the surprise [40]. Similarly, the Gap test is performed to calculate the gap between two repeating numbers [41]. The gap test is used to determine the implication of the interval between recurrences of the same digit. If the value of the above fitness function is close to 1 then it will be considered as the fitted value and the threshold for the proposed algorithm is more than 89%. The most fitted value will be recorded as the best.

**4.2.4. Mutation:** The process of mutation is helpful to achieve global optima. All the chromosomes which are greater than the threshold value in the last procedure will go under the process of mutation by again using the random point mutation procedure. The pseudorandom number will generate the random number and the binary bit will be flip of according to the resultant random number to generate a new chromosome as given by figure 3.

After applying mutation on each chromosome, we again calculate the fitness value of each resultant by using (1) and select the best one among them if the calculated fitness value after mutation is less than the previously recorded value then this process will be repeated for 50 rounds till, we have the more fitted value then the recorded one. If still we have the chromosomes whose value is less than the previously recorded value, then the recorded value is considered as final to become the

better half of the key. If the calculated fitness after mutation is greater than the recorded values, then this value is considered as the first half of the key. The whole procedure to generate the first half of the key can be seen in the flow chart of figure 4.

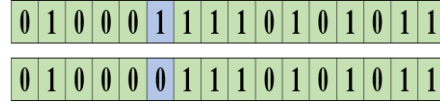


Fig. 3. Random Point Mutation

### 4.3. Linear Congruential Procedure for 64 Bits Binary Number

The A 64-bit linear congruential generator (LCG) is defined by the following recursive formula,

$$X_n \equiv aX_{n-1} - 1 \pmod{m}, n \geq 1 \quad (3)$$

Where  $m$  is the prime modulus, multiplier  $a$  and seed  $X_0$  are between 1 and  $(m-1)$  for a 64-bit computer in Eq. (3). The first bit of a signed integer is the sign bit, so the largest modulus presentable as an ordinary integer is  $2^{63}-1$  for a 64-bit machine. Three prerequisites for an ideal LCG are full period, randomness, and efficiency [42, 43]. The maximal period of an LCG is  $m - 1$ , called a full period LCG. An LCG is relatively easy to implement and reasonably fast. To generate a random number, it is important to have two parameters of an LCG: multiplier and modulus. Here we consider the 64-bit LCGs with prime modulus. Three forms of prime modulus are useful: Mersenne prime modulus, Sophie–Germain prime modulus, and largest prime modulus [44]. The distribution of Mersenne primes is sparse, so we can consider the largest Mersenne number  $2^{61} - 1$ , denoted as MP. There are infinitely many Sophie–Germain primes. The largest Sophie–Germain prime  $2^{63} - 4569$  is chosen and is denoted as SG. The largest prime modulus but not Mersenne prime and Sophie–Germain prime ones smaller than  $2^{63}$  is  $2^{63} - 25$ , denoted as LP. For a 64-bit LCG, we place the AF or DF restriction on the multiplier. Since the number of multipliers is astronomical, an exhaustive search appears to

be impractical. For portability and correctness, two types of restriction on multiplier are distinguished: approximate factoring (AF) multiplier and double-precision floating-point (DF) multiplier [45].

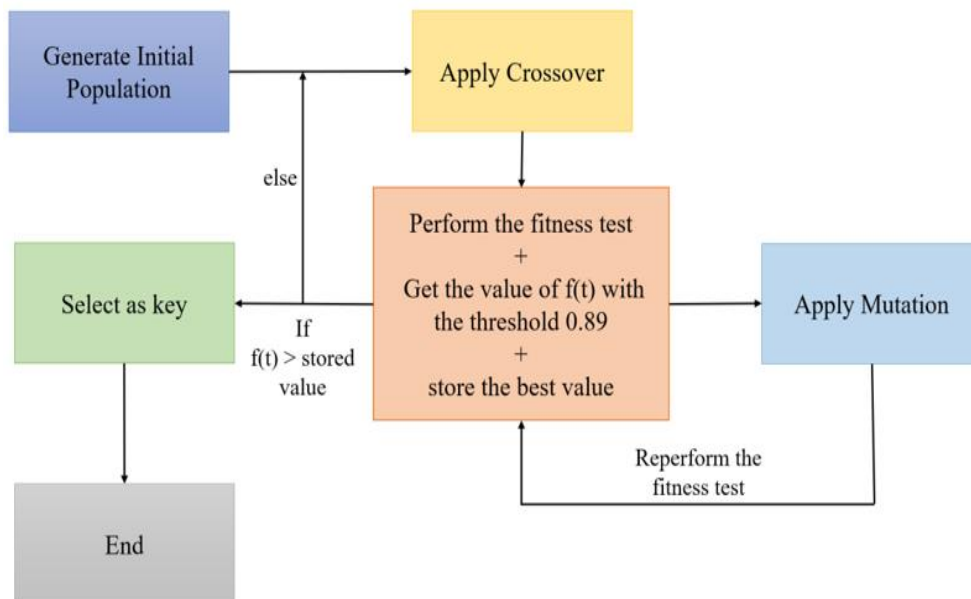


Fig. 4. The complete sequence of the proposed GA

#### 4.4. Pseudo-Random Number

A Pseudo-random number is a deterministically generated numbers that appear to be random. To generate these random numbers of various arithmetic approaches are used, on computers in the past thirty years or so. These approaches are usually recurrence relations and new numbers are generated from the earlier one by applying some simple scrambling operation. The most used method which is fast as well (or generator) is the so -called multiplicative congruential generator (sometimes also called the power residue generator). It consists of computing  $X_{i+1} = X_i \cdot a \pmod{m}$ , where  $X_i$  is a pseudo-random number,  $X_{i+1}$  is the next pseudo-random number,  $a$  is a constant multiplier and, modulo  $m$  means that the number  $X_i \cdot a$  is divided by  $m$  repeatedly till the remainder is less than  $m$  which is 64 in this case. The remainder is then set equal to the next number  $X_{i+1}$ . The process is started with

an initial value  $X_0$  called seed [46]. In the proposed algorithm we perform the one-point crossover that performs according to the random number generated by pseudo-random number and bit of that number mark as a crossover point and performs the one-point crossover as given by the figure 1.

##### 4.4.1. Summary of the 1st part of 64 bits key generation process:

- Initially generate the random number of 64 bits by using the linear congruential procedure.
- First, generate the pseudo-random number and obtained a random number from 0 to 63 as a crossover point, and perform the crossover operation.
- Apply the fitness test on each generated number obtained from step 1 with the

- help of eq (1) and stored the best value, which is greater or equal to 0.89.
- Generate the pseudo-random number and obtained a random number from 0 to 63 as a mutation point and perform a mutation procedure.
  - Again, calculate the fitness value and compare the best value with the stored value. If the stored value is less than the calculated value then stops, otherwise perform the above procedure until 50 rounds. If the calculated value is still less than the stored values, then finally stored values are supposed to be the final 64 bits first part of the key.

## 5. GENETIC ALGORITHM

As discussed earlier that in UWSNs data will be sent through some frames and each frame has different fields for different types of data information. To avoid attacks that may come in the form of a data frame, an authentication code(AC) will be calculated by each sensor to receive any data frame. The sensor will open the incoming data frame after the verification of AC, otherwise, it will be discarded. This authentication code is going to be used in the calculation of the remaining part of the key. The steps of generating the authentication code are as follows,

1. Take the numeric part of the sensor number and make it square take the mid-value

2. and the successive value and convert it into 32 bits binary values.
3. Now consider the alphabetic part of sensor number and value and convert it into 32 bits binary form and take the XOR between the value obtained from step I and II.
4. After applying the XOR the resultant is the authentication code for the incoming data frame.

### 5.1. Final Key Formation

1. Merge the authentication code and the frame number (which is numeric) of the incoming frame and convert it into 32 bits binary form and finally, we obtained the 64 bits value.
2. A combination of the first half of the key which was generated through GA and the other half value obtained from step 1, is considered as the 128 bits key for the proposed cryptographic suit for data communication in UWSNs.

$$\text{Final Key} = 64 \text{ bits from GA} + \text{AC (Frame no.)} \quad (4)$$

### 5.2. Algorithms for The Encryption & Decryption:

The step-by-step detailed encryption and decryption algorithms used in the proposed solution are represented below:

---

## ALGORITHM FOR ENCRYPTION

---

### START

- Step 1: Break the input Text file into 128 bits.  
Step 2: Split it into four equal parts of 32 bits.  
Step 3: P1=1st Part and 3rd Part.  
P2 = 2nd Part and 4th Part.  
Step 4: Split 128 bits key into four equal parts of 32 bits.  
Step 5: K1 = 1st Part and 3rd Part and  
K2 = 2nd Part and 4th Part.

### ENCRYPTION OF P1

- Step 6: Take P<sub>1</sub> Convert all characters of input plaintext into its ASCII.  
Step 7: Store and identify the minimum ASCII value.



Step 8: Perform the modulus operation on each character value by using the minimum ASCII value.

Step 9: Perform XOR with  $K_1 = d_1$

Step 10: Find base 64 value of  $K_1$

Step 11: Pick 8 alphabets =  $B_1$  (16 characters), use it as the first row of matrix write 128 bits of  $d_1$  in the form of the column below each alphabet.

Step 12: Apply the shifting of column (store the arrangement of  $B_1$  after column shifting) =  $R_1$ .

Step 13: Again apply the shifting of rows by writing  $B_1$  as the first column and the set of new values of  $R_1$  to be fix as a row corresponding to each alphabet of key the column (store the arrangement of  $B_1$  after row shifting) =  $E_1$ .

Step 14: Apply [bit XOR [(mod  $K_1+E_1$ ),64]] convert it into For ASCII values.

Step 15: Encrypted text of  $P_1$ .

#### ENCRYPTION OF P2

Step 16:  $P_1$  takes the transpose of  $P_1$  XOR  $K_2$ .

Step 17: Applying left rotation by 5.

Step 18: Add  $K_2$  in the resultant.

Step 19: Apply right rotation by 3

Step 20: add key in the resultant.

Step 21: Apply 2's Compliment.

Step 22: Convert it into ASCII values.

Step 23: Encrypted text of  $P_2$ .

END

---

#### ALGORITHM FOR DECRYPTION

---

START

##### DECRYPTION OF P1

Step 1: Ciphertext  $C_1$

Step 2: Apply XOR with .

Step 3: Calculate the reverse of mod 64 and subtract with  $K_1$ .

Step 4: By using the stored arrangement of  $B_1$  after shifting of rows, perform rearrangement of rows till getting the actual arrangement of  $B_1$  (according to base 64).

Step 5: By using the stored value of  $B_1$  after column shifting, rearrange the columns till getting the actual value of  $K_1$  (according to base 64).

Step 6: Perform XOR with 64 bits key.

Step 7: Perform the reverse mod operation according to the stored ASCII minimum value to get the original text  $P_1$ .

**DECRYPTION OF P2**

Step 8: Take the ASCII values equivalent to ciphertext C<sub>2</sub>.

Step 9: Apply 2's complement.

Step 10: Subtract K<sub>2</sub>.

Step 11: Apply left rotation by 3.

Step 12: Subtract K<sub>2</sub>.

Step 13: Apply right rotation by 5.

Step 14: Take XOR with K<sub>2</sub>.

Step 15: Obtained original text P<sub>2</sub>.

**FINAL TEXT**

Step 16: Split 64 bits of P1 into two parts.

Step 17: P1=1st Part and 3rd Part and

P2 = 2nd Part and 4th Part.

Step 18: Merge all parts according to the sequence.

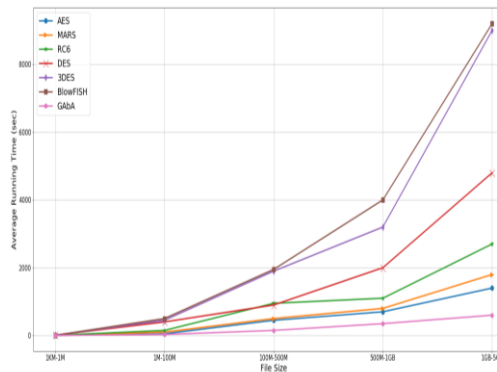
Step 19: The resultant 128 bits is the original plain text.

END

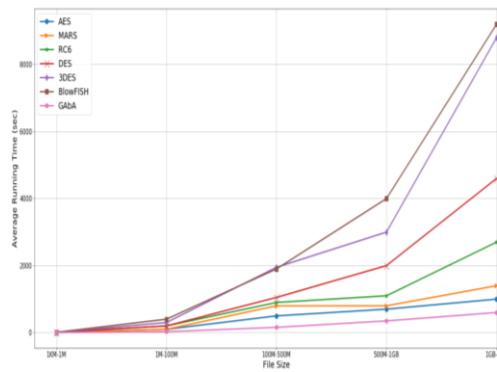
**6. Results and Analysis**

This section provides the results and analysis of the performance of the proposed algorithm. The evaluation can be done based

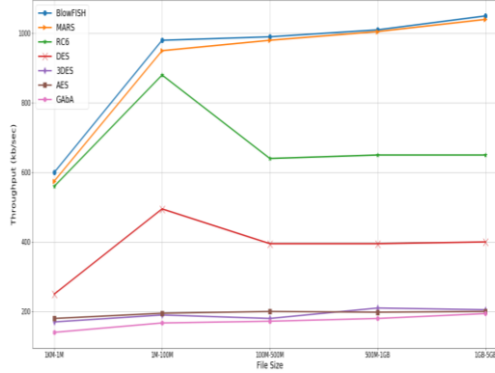
on the main features of any existing cryptographic algorithm. The details of our evaluation will show how the proposed algorithm maintains its security while implemented.



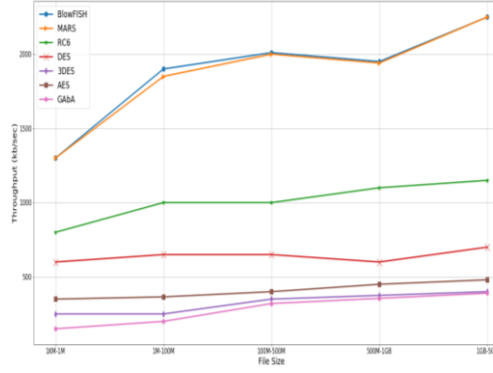
**Fig. 5(a).** Average Running Time for Encryption procedure by different Algorithms



**Fig. 5(b).** Average Running Time for Decryption procedure by different Algorithms



**Fig. 5(c).** Average Throughput for Encryption procedure by different Algorithms



**Fig. 5(d).** Average Throughput for Decryption procedure by different Algorithms

### 6.1. Equations Elitism Criteria in GA

We have generated our key through the genetic algorithm and while generating the random number for a key generation we have used the concept of elitism in our coding which enhances the selection criteria for any generated chromosomes. It is important to maintain adequate selection pressure, as demanded by the application, to avoid genetic drift. Elitism can increase the selection pressure by preventing the loss of low “salience” genes of chromosomes due to deficient selection pressure; it improves the performance of optimality and convergence of GAs in many cases. Elitism provides a means for reducing genetic drift by ensuring that the best chromosome(s) is allowed to pass/copy their traits to the next generation [47].

### 6.2. System Environment

The proposed algorithm was implemented in MATLAB and the comparison has been evaluated against some benchmark symmetric encryption algorithms like 3 DES, MARS, Rivest Cipher (RC6), Data Encryption Algorithm (DES), Blow FISH, and AES in terms of running time, throughput for encryption and decryption along with the avalanche effect and space usage [47].

The data which is used in our experiments are real data that has been used between sensor communication and the proposed algorithm has been tested for different file size which is randomly taken between some intervals of [1K-1M], [1M-100M], [100M-500M], [500M-1 G] and [1G-5G]. The aim is to show that the performance of the proposed algorithm in terms of all the above-mentioned features is better than the existing algorithms which are supposed to be the benchmark techniques. The indicated results are based on average values because each test was conducted several times. The cryptographic algorithm starts with the process of key generation.

### 6.3. Performance Analysis

Here we discuss our implementation and results in terms of performance for security features like throughput, processing time, and the avalanche effect.

Figure 5 (a), 5(b) is showing the average running time of encryption and decryption procedure for the proposed algorithm together with other benchmark procedures for different input file sizes as mentioned above. The running time is mainly the time taken by any

algorithm to encrypt/ decrypt any plain text into ciphertext. According to the figure it can be seen that the proposed algorithm has a better performance in terms of running time for encryption and decryption. After this, we evaluate our GA based method for the throughput of the encryption/decryption procedure. Throughput in (bytes/sec) can be define by using the following formula given in (5),

$$Throughput = \frac{\sum Input File Size}{\sum Execution Time} \quad (5)$$

Figure 5(c), 5(d) showing the average throughput for the encryption and decryption for the proposed algorithm, and as compared to other techniques it can be easily seen that our algorithm has better throughput for random file sizes.

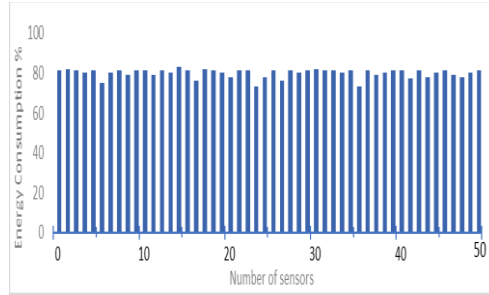
#### 6.4. Energy consumption and network lifetime

Balancing the energy is not an easy task in underwater sensor networks. A balanced network is one in which the remaining amount of energy is the same in the end, which means that each node does not die before others. Sensors balance their energies by sharing the duties which need an extra amount of energy. In underwater sensor networks, the initial energy of each sensor node is the same, but in the proposed technique the energy consumption of each sensor is almost identical after the transmission process. Since in the provided algorithm the sensors are in sleep mode before receiving and after transmitting the data to the base station in each round.

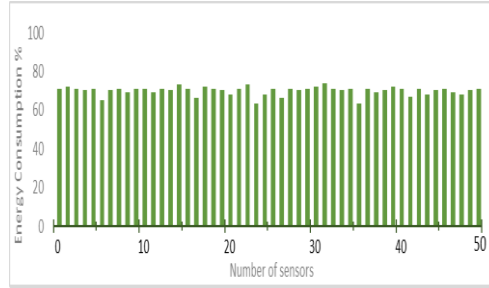
During the execution of the proposed method energy of each sensor after 100 rounds is almost the same. But after 500 rounds of GA, it is evident that the energy of some sensors is almost the same, and some have fluctuated at the end of the network lifetime. From table 1 the mean and standard deviation of the remaining energy of sensor nodes which means that the energy variation of all sensors is almost the same.

**Table 1:** The mean and standard deviation of the remaining energy for three simulation runs

Rounds	100	300	500
Mean	0.38	0.29	0.24
S. D	0.009	0.013	0.025



**Fig. 6(a).** Sensor’s energy level after 100 rounds.



**Fig. 6(b).** Sensor’s energy level after 500 rounds.

#### 6.5. Security Analysis

This section will discuss the evaluation of our proposed algorithm on the basis of the security purpose which is known as avalanche effect. The avalanche effect is measuring the strength of the algorithm for hacking and cracking threats. Real-time threats such as brute force attacks can be measured by avalanche effects, and it requires the number of bits that have been changed during the process of encryption from plain text to ciphertext. Avalanche effect can be calculated by using the formula,

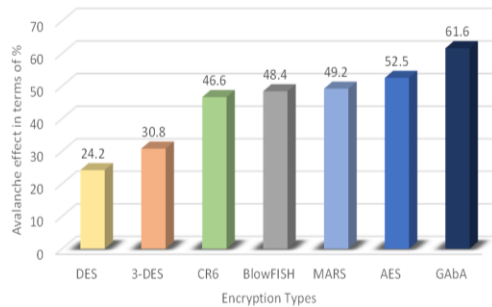
$$\text{Avalanche effect} = \frac{NFBCT}{NBCT} \times 100\% \quad (6)$$

where,

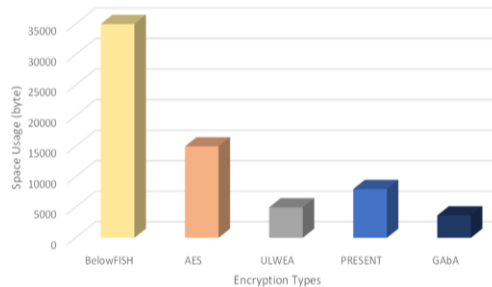
$NFBCT$  = No. of flipping bits in the ciphertext

$NBCT$  = No. of bits in the ciphertext

According to figure 7 AES is the only algorithm that can manage a high avalanche effect as compared to other benchmark algorithms. But proposed the algorithm shows the highest avalanche effect in comparison to AES and of course rest of the other encryption techniques.



**Fig. 7.** Avalanche effect of different algorithms in terms of %



**Fig. 8.** Space usage of different algorithms (in bytes)

## 7. Security Evaluation

As explained earlier, UWSNs possess a limited battery with very low space and lots of security issues. The results from the proposed evaluation can conclude that the presented algorithm produced the lowest running possessing time, highest throughput, and highest avalanche effect, and this made our

proposed algorithm fast, secure, efficient, and reliable.

In this section, we analyze the security of the proposed algorithm against various attacks related to underwater sensor networks. For providing secure cryptographic algorithms, it is important to take care of different kinds of attacks that might occur while transferring the data between the nodes.

**7.1. Plain text attack:** Considered the most basic attack on the cryptographic algorithms, this attack arises when an attacker tries to attack both plain text and ciphertext. This attack works during the data transmission for encryption and snatches the chunks of plain text. Since it is difficult to get the key, therefore, attackers try to generate the method of encryption with the help of some portion of the plain text and ciphertext. Later it is used for the decryption of ciphertext. In the proposed technique the plain text is not transmitted within the sensor nodes only cipher text is sent. So it is impossible that this attack can occur because if the attacker can be able to capture some portion of the ciphertext then it is impossible for the attacker to get the key as the key is updated in every round and the cipher-text also changes in each round.

**7.2. Ciphertext Attack:** A very common attack called ciphertext attack in which an attacker can only be able to access some portion of ciphertexts. This attack could be very dangerous if corresponding plain text is extracted or even more harmful if the key can be deduced. This attack could easily occur if the ciphertext is sent through the network but, in the proposed technique, the ciphertext is divided into four parts, and encryption is implemented piecewise on the plain text, which makes it almost impossible for that this

attack to occur. Also, in the proposed technique the key does not directly send over the network. It will be calculated on the end of the data receiving sensor by using some information given by the network which is not complete. Therefore, without the information of key, it is difficult for the ciphertext to be decrypted by using the proposed technique.

**7.3. Related-key attack:** This attack can occur in any form of cryptanalysis in which the attacker tries to pick the operation of cipher by using different keys that are initially not completely known to accept some mathematical operations related to key are known. For example, if the attacker got some information that the last 80 bits of the keys are always the same as having the information about the actual bits. But in the proposed technique the key is not always the same some each text. As per the key generation procedure which is calculated on the basis of sensor number and randomly generated ID at each round therefore it is not possible that this attack will occur for the proposed technique. As the key is generated by using randomly generated parameters which make this attack impossible to occur by using this technique.

**7.4. Man in the Middle Attack (MIM):** The proposed technique is highly vulnerable to this attack, because of the two reasons, the proposed technique is based on the randomly generated parameters. Secondly, the proposed technique generates the hash function for the authentication code that is why this attack becomes the meaningless for the proposed technique. Three people have involved in this attack: the victim, the attacker, and the person to which the victim tries to communicate. This attack tries to access the secret parameter

values, but due to the involvement of authentication code which will be verified by sender and receiver, that makes this attack weaker for the proposed technique. This attack is possible if the attacker tries to hear the conversation between the sender and receiver, but for the proposed algorithm if this happened then the data will be decrypted by using the key. As defined earlier the key does not send over the network it will be calculated by the receiving sensor that is why man in the middle attack is helpless as it can't get the key.

**7.5. Hello Flood Attack:** This attack sends HELLO packets in order to consume network resources. But in the proposed technique the receiving sensor does not receive any packets without an authentication code, therefore the proposed technique is immune to this attack.

**7.6. DoS Attack:** This attack utilizes the network bandwidth by sending advisory packets which prevent the user from utilizing the services and resources. This attack usually occurs on the cluster head of the network. The proposed methodology overcomes this attack by sending the acknowledgment message by the base station.

**7.7. Compromise Cluster Head Attack:** This attack tries to get all the information from the cluster nodes by making them believe that it is working as a cluster head. The main task of this attack is to extract the basic information from the data. In our proposed technique the cluster head sends the data from the nodes without decrypting it, also in the proposed technique encryption is depending on many parameters such as sensor number, authentication code, randomly generated numbers at each

round. To encrypt the data the attacker, must know the complete information about all the operations along with the secret binary string which is generated by the key generation procedure. Therefore, due to all above-mentioned factors, we can easily say that for the proposed technique this attack is not possible to occur.

## 8. Conclusion

This study has intended to provide the secure crypto base encryption algorithm (GA) for the communication between underwater sensors. The proposed algorithm can be able to generate the first half portion of the key by using a genetic algorithm that is based on 32 bits, after that, it calculates the authentication code for the final key generation of 128 bits. In GA a fitness function has been introduced with the parameters that helped in the identification of the best chromosomes among the random population. By the help of GA steps first part of the key will be generated and the remaining part will be generated by some different procedure. Furthermore, there are two separate procedures of encryption also, with the idea to make encryption procedure trickier without involving complex steps to avoid the computational complexities. Overall procedure has been designed in a way that can be able to avoid different threats which are very obvious in underwater communication. An authentication code has been used to protect the data from passive attacks. The comparison has been implemented with other benchmark symmetric encryption techniques to show the efficiency of the proposed cryptographic algorithm in terms of running processing time, throughput, and the avalanche effect. The reason behind using the GA technique is due to its randomness. GA is a random procedure, and it makes key guessing almost impossible. The novelty of the proposed work is that GA has never been involved in underwater secure communication. This study has proved that

with the help of GA and the present-ed encryption procedure we can efficiently be done the secure communication between underwater sensors and security can be made even better by the implementation of the proposed algorithm.

## 9. Future Works

Future work includes inclusion of lightweight methods for additional protection from new and evolving types of attacks. The availability of improved computational and energy-sufficient sensors will also allow the inclusion of complex algorithms and encryption-decryption techniques to further secure the underwater sensor networks.

## REFERENCES

- [1] Y. Xuanxia, Z. Chen, and Y. Tian. "A lightweight attribute-based encryption scheme for the Internet of Things." *Future Generation Computer Systems* vol. 49, pp.104-112, 2015
- [2] D. Gianluca, and A. L. Duca. "A secure communication suite for underwater acoustic sensor networks." *Sensors*, vol. 12, no. 11, pp 15133-15158, 2012.
- [3] S. Kotari, and MB. M. Krishnan. (2018, August), "Improvisation of underwater wireless sensor network's efficiency for secure communication." *IOP Conference Series: Materials Science and Engineering*. Vol. 402. No. 1.
- [4] C. Angelo et al. (2017, November), "Securing Underwater Communications: Key Agreement based on Fully Hashed MQV." *Proceedings of the International Conference on Underwater Networks & Systems*. Pp. 1-5.
- [5] R. G. Nitinkumar, and B. R. Kaur. "A new approach for data encryption using genetic algorithms and brain mu waves.", *International Journal of Scientific and Engineering Research*, Vol. 2, No. 5, May, 2011.
- [6] A. Faiyaz, S. Khalid, and M. S. Hussain. "Encrypting data using the features of memetic algorithm and cryptography." *International Journal of Pattern Recognition and Artificial Intelligence*, vol, 2, No. 3, pp.109-110, June, 2011
- [7] M. Swati, and S. Bali. "Public key cryptography using genetic algorithm." *International Journal of Recent Technology and Engineering* vol. 2, No.2, pp. 150-154, May, 2013.
- [8] Deng, Lih-Yuan, Henry Horng-Shing Lu, and Tai-Been Chen. "64-Bit and 128-bit DX random number

- generators." *Computing*, vol.89, No1-2 pp.27-43, August, 2010.
- [9] J. Sania, and A. Jamal. "Generating the best fit key in cryptography using genetic algorithm." *International Journal of Computer Applications*, Vol.98, No.20, pp.33-39, July, 2014.
- [10] G. S. Fishman, "Random Tours." Monte Carlo. Springer, New York, NY, 1996.
- [11] K, Donald. "Seminumerical algorithms." *The art of computer programming 2*, 1981.
- [12] P. Srikanth, et al. "Encryption and decryption using genetic algorithm operations and pseudorandom number." *Computer Science and Network*, Vol.6, No.3, pp.455-459, 2017.
- [13] T. Hui-Chin, and H. Chang. "An exhaustive search for good 64-bit linear congruential random number generators with restricted multiplier." *Computer Physics Communications* Vol.182, No.11, pp.2326-2330, November, 2011.
- [14] M. P. More, and P. G. Naik. "Hybrid Security Framework for Activity Based Authentication using RSA & Genetic Algorithm." *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol.3, No.11, pp.6175-6184, November, 2015.
- [15] D. Dumitru, et al. *Evolutionary computation*. CRC press, 2000.
- [16] P. M. Reed, B. S. Minsker, D. E. Goldberg "The practitioner's role in competent search and optimization using genetic algorithms", In *Bridging the Gap: Meeting the World's Water and Environmental Resources Challenges* pp. 1-9, 2001.
- [17] A. Shadi, and M. B. Yassein. "A resource-efficient encryption algorithm for multimedia big data." *Multimedia Tools and Applications*, Vol. 76, No.21, pp.22703-22724, 2017.
- [18] N. M. Irshad, et al. "Implication of genetic algorithm in cryptography to enhance security." *Int. J. Adv. Comput. Sci. Application*, Vol.9, No.6, pp.375-379, June, 2018.
- [19] S. Dutta, et al. "A cryptography algorithm using the operations of genetic algorithm & pseudo random sequence generating functions." *International Journal of Advances in Computer Science and Technology*, Vol.3, No.5, May, 2014.
- [20] Delman, Bethany. "Genetic algorithms in cryptography." [MS.Thesis]. Rochester Institute of Technology, Rochester, New York, 2014.
- [21] S. Goyat. "Genetic key generation for public key cryptography." *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 2, No. 3, pp. 231-233, July, 2012.
- [22] C. Peng, et al. "An ultra-lightweight encryption scheme in underwater acoustic networks" *Journal of Sensors*, Vol.2016, February, 2016.
- [23] M. BAYKARA, et al. "A novel symmetric encryption algorithm and its implementation." *Firat University Turkish Journal of Science and Technology*, Vol. 12, No.1, pp.5-9, 2017.
- [24] G. Han, et al. "Secure communication for underwater acoustic sensor networks." *IEEE communications magazine*, Vol. 53, No.8, pp. 54-60, August, 2015.
- [25] R. Jhingran, V. Thada, and S. Dhaka. "A study on cryptography using genetic algorithm." *International Journal of Computer Applications*, Vol.118, No.20, pp.10-14, January, 2015.
- [26] M. Jouhari et al. "Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and internet of underwater things". *IEEE Access*, Vol. 7, pp.96879-96899, July 2019.
- [27] M. Khalid et al. "A survey of routing issues and associated protocols in underwater wireless sensor networks." *Journal of Sensors*, Vol. 2017, 22, May, 2017.
- [28] J. E. Kim et al. "Security in underwater acoustic sensor network: focus on suitable encryption mechanisms." *Asian Simulation Conference*. Springer, Berlin, Heidelberg, pp. 160-168, 27, October, 2012.
- [29] A. Kumar, and C. Kakali. "An efficient stream cipher using genetic algorithm." *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, pp. 2322-2326, 23, March, 2016.
- [30] A. Kumar and M. K. Ghose. "Overview of information security using genetic algorithm and chaos." *Information Security Journal: A Global Perspective*, Vol.18, No.6, pp.306-315, 9, December, 2009.
- [31] C. Lal et al. "Toward the development of secure underwater acoustic networks." *IEEE Journal of Oceanic Engineering*, Vol.42, No.4, pp.1075-1087, 6, July, 2017.
- [32] Y. Liu, J. Jing, and J. Yang. "Secure underwater acoustic communication based on a robust key generation scheme." *2008 9th International Conference on Signal Processing*. IEEE, pp. 1838-1841, 26, October, 2008.
- [33] G. Ateniese, et al. "SecFUN: Security framework for underwater acoustic sensor networks." *OCEANS 2015-Genova*. IEEE, pp. 1-9, 18, May, 2015.
- [34] S. Moffat, M. Hammoudeh, and R. Hegarty. "A survey on ciphertext-policy attribute-based encryption (CP-ABE) approaches to data security



- on mobile devices and its application to IoT." In Proceedings of the International Conference on Future Networks and Distributed Systems. 19, July, 2017.
- [35] D. Pompili, and I. F. Akyildiz. "Overview of networking protocols for underwater wireless communications." *IEEE Communications Magazine*, Vol.47, No.1, pp.97-102, 10, February, 2009.
- [36] G. R. S. Qaid and S. N. Talbar. "Bit-Level Encryption and Decryption of Images using Genetic Algorithm: A New Approach." *IPASJ International Journal of Information technology (IJIT)* 1.6 (2013).
- [37] M. Stojanovic, "Underwater wireless communications: Current achievements and research challenges." *IEEE Oceanic Engineering Society Newsletter*, Vol. 41, No.2, pp. 1-5, November, 2006.
- [38] S. Y. Tan, K. W. Yeow, and S. O. Hwang "Enhancement of a lightweight attribute-based encryption scheme for the Internet of Things." *IEEE Internet of Things Journal*, Vol. 6, No.4, pp.6384-6395, 25, February, 2019.
- [39] Yang, Guang, et al. "Challenges and security issues in underwater wireless sensor networks." *Procedia Computer Science*, Vol. 147, pp.210-216, 1, January, 2019.
- [40] G. Yang, L. Dai, and Z Wei. "Challenges, threats, security issues and new trends of underwater wireless sensor networks." *Sensors*, Vol.18, No.11, pp.3907, November, 2018.
- [41] K. M. Awan, et al. "Underwater wireless sensor networks: A review of recent issues and challenges." *Wireless Communications and Mobile Computing*, Vol.2019, 1, January,2019.
- [42] Y. Cong, et al. "Security in underwater sensor network." 2010 International Conference on Communications and Mobile Computing. Vol. 1, pp. 162-168, IEEE, 12, April, 2010.
- [43] M. C. Domingo, "Overview of channel models for underwater wireless communication networks." *Physical Communication*, Vol. 1, No.3, pp.163-182, 1, September, 2008.
- [44] R. Ebrahimzadeh and M. Jampour. "Chaotic genetic algorithm based on lorenz chaotic system for optimization problems." *International Journal of Intelligent Systems and Applications*, Vol. 5, No.5, pp.19, 1, April, 2013.
- [45] C. M. G. Gussen, et al. "A survey of underwater wireless communication technologies." *J. Commun. Inf. Sys*, Vol.31, No.1, pp. 242-255, 27, October, 2016.
- [46] I. F. Akyildiz, D. Pompili, and T. Melodia. "Challenges for efficient communication in underwater acoustic sensor networks." *ACM Sigbed Review*, Vol. 1, No.2, pp.3-8, 1, July, 2004.
- [47] P. Sriitha, et al. "A new modified RC6 algorithm for cryptographic applications." *Int. J. Adv. Res. Comput. Commun. Eng*, Vol. 3, No.12, pp.2278-1021, December, 2014.