# Identity-Based Encryption Algorithm using Hybrid Encryption and MAC Address for Key Generation

**Darpan Anand, Jeetendra Pande, Udit Maheshwari**

***Abstract:*** ***—In today's world, secured communication is what every field wants. But due to the escalation of theft and unauthorized access to the data, the need for new encryption algorithms has also gone high. This paper proposed a hybrid encryption scheme which uses ElGamal key exchange algorithm and Genetic algorithms. In this scheme, the users share seventeen 16-bit keys, random in nature, using the ElGamal algorithm, which is used to calculate a shared key. Along with this shared key and MAC(Medium Access Control) addresses of both sender and receiver(MAC is used as it is a unique identifier attached to a network adapter or system), a final key is generated through a 10-round algorithm. Further, this final key is used for data encryption using Genetic algorithms. A new key is generated for each communication session between sender and receiver. The proposed technique shows that it is resistant to most of today's known and common attacks.***

***Keywords: Identity-Based Encryption, Genetic Algorithm, Mutation, Crossover.***

## I. INTRODUCTION

The information security has become a crucial, important and imperative issue in applications of information exchange. It is becoming more important due to the increase in laptops and other computing devices, growth and advancement in technologies, various network technologies and huge quantity of data in the form of digital information. Therefore, to shield the valuable information from unauthorized users' various varieties of cryptanalytic schemes are developed and applied in existing systems. Cryptography,'kryptos' in Greek, means 'Secret Writing'. It is defined as the art (sometimes as science) of transforming messages into unreadable form to make them secure and immune to various attacks. The original message, which is willing to send by sender from its end is plain text while the coded or encrypted message is cipher text, send to receiver.

**Revised Manuscript Received on October 30, 2019.**
* Correspondence Author

**Darpan Anand***, Associate Professor, Department of Computer Science Engineering Chandigarh University, Chandigarh, India. Email: darpan.anand.agra@gmail.com

**Jeetendra Pande**, Assistant Professor, Department of Computer Science, Uttarakhand Open University, Haldwani-263139, India. Email: jpande@uou.ac.in

**Udit Maheshwari**, Hindustan Institute of Technology and Management, India. Email: maheshwariudit29@gmail.com

The method used to change the plain text to encoded/cipher message is ciphering or encryption while the reverse is called as deciphering or decryption method. The filed of study belons these techniques to ensure the security of the message is called as cryptography. The major types of cryptography techniques are public key cryptography, key escrow cryptography, translucent cryptography, symmetric key cryptography, and ID-Based Cryptography. One of the vulnerabilities of these techniques are due to the key, which is used for encryption and decryption. [1]-[3] Shamir introduced the solution to handle the problem of key by the concept of identity-based(Id-Based) cryptography [4]. In this technique, the person is able to deduce the public from the public information of user which are uniquely identified like email, phone number, SSN(In case of USA, etc.), AADHAR(in case of India). The major benefit of using Id-based Encryption(IBE) is that it does not need any certificates to bind the users' names with their public keys. The first practical and provably secure ID-based encryption scheme was proposed by Boneh and Franklin in 2001 [5]. Since then, IBE has undergone quite rapid development and a lot of schemes have been proposed [6]-[9]. The most difficult or most vulnerable area in information security is identity of the entity. Generally, string based pass phrases are being used to ensure the authentication and securing the identity of the user. These is problem with these pass phrases is the key-logging because the pass-phrase can be sniffed through a key logger software. Therefore, in place of pass phrases, the hardware dependent unique information can be used to ensure the security of the identity of the entity. As we know, every network device is equipped with a unique address called as MAC address which is globally unique. This address can be considered as the deriving entity to deduce the public key in ID-Based cryptography as it is unmodifiable and globally unique hardwired value. This is also called as physical network address and it is of 12 hexadecimal numbers, 48 bits in length and the format of this number is as: **MM:MM:MM:SS:SS:SS** or **MM-MM-MM-SS-SS-SS.** The MAC address consists two identities as half part is addressing the identity of the manufacturer of the device and rest half part is addressing the serial number or identity of the hardware. The best thing to pick the MAC address as the seed for ID-Based cryptography is that, no active device can change its MAC address, along with the permission given by the wireless gateway.

Because the authorized and registered MAC address requests can only be allowed to communicate [10]. Spoofing a MAC address is basically identity theft and denotes the altering of the MAC on a NIC. Therefore, many approaches have been introduced to generate NIC fingerprints [11].3

These approaches include Radio Frequency Fingerprinting [12], Passive Data Link Layer Fingerprinting [13], and Acknowledge-Frame Delay Fingerprinting [14].

## II. METHODOLOGY

In this algorithm, ElGamal key exchange algorithm is used for the sharing of key between sender and the receiver, and Genetic Algorithms for the encryption and decryption of data. A shared key is generated first between both the users using ElGamal key exchange algorithm. In this paper, the ElGamal algorithm is modified according to the requirement such that instead of sharing one key, it shares seventeen random keys and these seventeen keys calculates the shared key. Using this key, a final key is calculated by the proposed algorithm. The final key is generated using the shared key as well as the MAC addresses of both sender and receiver. This final key is then used for the process of encryption and decryption. The process of encryption is done by crossover, mutation and re-sequencing techniques. The decryption of message is done by reversing the techniques used in encryption. The ElGamal key exchange algorithm (unmodified) is defined in the following section.

### A. ElGamal Key Exchange Algorithm

In 1984, T. Elgamal announced a public key scheme, ElGamal cryptosystem, based on difficulty of discrete logarithm problem for finite fields [15]. The Elgamal key exchange algorithm has three parts:

*1) Generating Key:*

- Choose a large prime number p, and q, which is a primitive root of p.
- Generate a random integer X, such that $1 < X < p - 1$
- Compute $Y = q^X \bmod p$
- Sender's private key is X and public key is {p,q,Y}

*2) Encryption of a Message:*

- Represent the message as an integer M in the range $0 \leq M \leq p - 1$
- Choose a random integer k such that $1 \leq k \leq p - 1$
- Compute a one-time key $K = (Y)^k \bmod q$
- Encrypt M as the pair of integers $(C_1,C_2)$
- $C_1 = q^k \bmod p$; $C_2 = (KM) \bmod p$

*3) Decryption of a Message:*

• Recover the key by computing $K = (C_1)^X \bmod p$
• Compute $M = (C_2K^{-1}) \bmod p$

### B. Genetic Algorithms

The genetic algorithm is a search algorithm based on the mechanics of natural selection and natural genetics. An initial population is randomly generated which is composed of several chromosomes which are either binary or hex decimal depending on the type of population. The individuals are selected based on their probabilities, genetic operations and fitness values. The genetic algorithm uses two reproduction operators: crossover and mutation. Reproduction gives the searching power to the genetic algorithms. In crossover, parents are paired together. There are several types of crossover operators, and the types available depend on what representation is used for individuals. The one-point crossover means that the parent individuals exchange a random prefix when creating the child individuals. The purpose of the mutation operator is to simulate the effect of transcription errors that can happen with a very low probability when a chromosome is mutated.

## III. PREVIOUS WORK

There are few encryption techniques based on the genetic algorithms. One of the algorithm is described by the Kumar and Rajpal in which encryption is done using the concept of genetic algorithm which uses the genetic processes as crossover and pseudo random sequence generator by NLFFSR(Nonlinear Feed Forward Shift Register). The pseudo random number is used to determine the point of the crossover and therefore, fully encrypted data may be achieved [16]. This work is further extended by Kumar and Rajpal in which, they gave the concept of the mutation implemented after the process of encryption. Finally, encrypted data are further hidden inside the stegoimage [17]. Husainy proposed Image Encryption using genetic algorithm-based Image Encryption using mutation and crossover concept [18]. A. Tragha et al., described a new symmetrical block cipher technique named ICIGA(Improved Cryptography Inspired by Genetic Algorithms) which generates a session key in a random process. The block sizes and the key lengths are variable. ICIGA is an enhancement of the system GIC(Genetic Algorithms Inspired Cryptography) [19]. In 2013, M.A. Al-Husainy proposed a scheme using MAC address and genetic algorithm for encryption purpose. The scheme used receiver's MAC address as a key for encrypting data. The data was encrypted by Genetic algorithm(Mutation and Crossover) and re-sequencing using this key. To ensure enough distortion in the original message and the encrypted data, the measurement of Signal-to-Noise Ration was used. The SNR was calculated by using the following formula, where S and E represent the source and the encrypted image respectively:

$$SNR_{db} = \frac{\sum_{i=1}^{width}\sum_{j=1}^{height}(E_{ij})^2}{\sum_{i=1}^{width}\sum_{j=1}^{height}\left(E_{ij} - S_{ij}\right)^2}$$

To evaluate the key sensitivity feature of the scheme, a one bit change was made. The decryption with wrong key resulted in a completely different image. Therefore, this scheme was highly sensitive to the key. S. Jawaid et. al proposed a scheme for the selection of fittest key using genetic algorithm and autocorrelation. In this scheme, the keys generated were tested for randomness by using autocorrelation test (tests whether random number generator is producing independent random numbers in sequence) and the final key was selected on the basis of autocorrelation value [20].

## IV. PROPOSE D AL GORITHM

To implement this approach we need to use Hybrid Encryption , i.e.,ElGamal Key Exchange Algorithm for key exchange between the sender and the receiver , and Genetic Algorithms- Cross Over and Mutation of Chromosomes(data bits) along with Re-sequencing for the encryption of data. The algorithm is divided into four phases as :-

1) *Sharing Key*
2) *Generating Final Key*
3) *Encryption of Data*
4) *Decryption of Cipher Text*

## A. Sharing Key

In ElGamal algorithm, instead of sharing one key, seventeen keys are shared which are further used to calculate a single shared key for both sender and receiver.

1) Sender S shares seventeen 16-Bit numbers M[i], such that $1 \leq i \leq 17$ (which are generated by using Random Number Generator) using ElGamal key exchange algorithm.
2) For ElGamal algorithm , Receiver R needs prime numbers **q** and a generator **g** (g is a primitive root of q). To generate the values of q , we use Random Number Generator.
3) Now, R chooses a secret number **X** to act as his private key which is a random number such that $1 < X < q$, and he computes the quantity:

$$B = g^{X} (\text{mod } q)$$

4) R publishes his public key $\{q, g, B\}$ and keeps his private key X as secret. Now the sender will encrypt data using this public key and receiver will decrypt it using his private key.
5) Now, Sender S wants to encrypt all the keys of M[i], such that $1 < i \leq 17$, and send one key at a time, where each key of M[i] should satisfy $0 \leq M[i] \leq q - 1$. So, sender S chooses a random secret key k which encrypts one key for a session. He takes his random key k such that $1 \leq k \leq q - 1$, and R's public key $\{q, g, B\}$ to compute one time key :

$$Y = B^{k} \ (\text{mod } q)$$

6) Sender S calculates cipher text as encryption of M[i], as the pair of numbers ($C_1$ , $C_2$ [i]),suchthat$1 \leq i \leq 17$. S calculates this cipher text as following and sends it to R :

$$C_1 = g^{k} \ (\text{mod } q)$$

$$C_2[i] = M[i].Y \ (\text{mod } q)$$

such that $1 < i \leq 17$ and $C_2[1] = M[1]$. Here $C_2$ is calculated for 16-times with a new M each time and sent as an array of size 17 with an operational key M[1] at $C_2$[1].

7) Receiver R recovers the one time key $S_{key}$ and the operational key by computing :

$$Y = C_1{}^{X} \ (\text{mod } q) \ ,$$

$$M[1] = (C_2[1].Y^{-1})(\text{mod } q)$$

The operational key M[1] is converted into a binary string array, ch[16], of 0's and 1's , and each encrypted key is recovered as:-

$$M[i] = (C_2[i].Y^{-1})(\text{mod q})$$

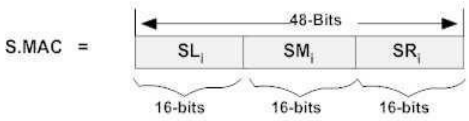
and the shared key S operational key as:

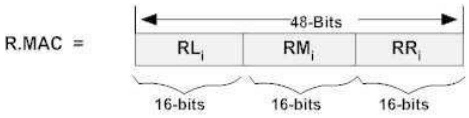
$$\textbf{if}(\text{ch}\lfloor \text{i-1} \rfloor \text{==}\text{'1'})$$
$$\textbf{then} \ ,S_{key} = ShiftRight(S_{key}, 8 - bits)$$
$$S_{key} = S_{key} \bigoplus M[i]$$

Here, the calculation of $S_{key}$ is predefined which states that if a '1' is encountered then the key is shifted right by 8-bits and XORed with previous key. Otherwise, it is simply XORed with the previous key without shifting.

## B. Generating Final Key

- After sharing a key, $S_{key}$, both sender and receiver computes a 6-Byte key for encryption and decryption of message respectively using the same algorithm at their respective ends.
- The sender's MAC address SMAC, having 48-bits, is divided into three parts of 16-Bits each as SL, SM and SR (as Left, Middle and Right part of Sender's MAC). Similarly, receiver's MAC address RMAC is divided into three parts of 16-Bits each as RL, RM and RR (as Left, Middle and Right part of Receiver's MAC).





- The shared key $S_{key}$ undergoes processing through 10 rounds of the following algorithm to compute the final key $F_{key}$ . This calculated final key $F_{key}$ is then used for encryption and decryption at respective ends of sender and receiver. The final key is a 6-Byte key. The 10 rounds algorithm is as follows:

1) ROUND 1: This is the first round of key generation algorithm. In Round 1, the second block of SMAC, i.e., SM1 , is XORed with first block of RMAC ,i.e. RL1, and the first block of SMAC, i.e.,SL1, is XORed with second block of RMAC, i.e., RM1 as shown below:

$$SL_2 \leftarrow SM_1 \bigoplus RL_1 \ ,SM_2 \leftarrow SM_1 \ ,SR_2 \leftarrow SR_1,$$
$$RL_2 \leftarrow SL_1 \bigoplus RM_1 \ ,RM_2 \leftarrow RM_1 \ ,RR_2 \leftarrow RR_1$$

This round gives three new blocks of each sender's and receiver's MACs. These new values will be used as input for Round 2. The new blocks are calculated as shown in the following figure 1 below.

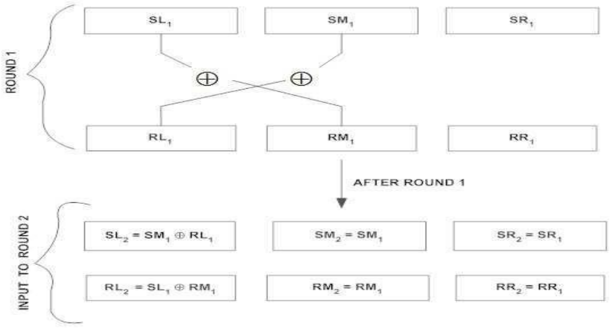


**Figure 1: Round 1**

2) ROUND 2: The output of Round 1 works as the input to Round 2. In this round, the $SR_2$ is XORed with $RM_2$ to give $SM_3$ and $SM_2$ is XORed with $RR_2$ to give $RM_3$. The values for next round are calculated as:

$$SL_3 \leftarrow SL_2, SM_3 \leftarrow SR_2 \bigoplus RM_2, SR_3 \leftarrow SR_2,$$
$$RL_3 \leftarrow RL_2\ , RM_3 \leftarrow SM_2 \bigoplus RR_2, RR_3 \leftarrow RR_2$$

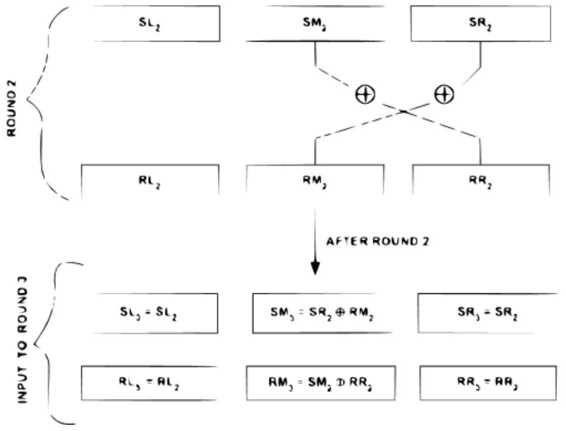


**Figure 2: Round 2**

3) *ROUND 3:* This round makes use of the shared key $S_{key}$. The $S_{key}$ is XORed with both $SR_3$ and $RR_3$. After XORing the key with the blocks, the results are swapped in order to generate $SR_4$ and $RR_4$ respectively. This is represented as:

$$SL_4 \leftarrow SL_3, SM_4 \leftarrow SM_3, SR_4 \leftarrow RR_3 \bigoplus S_{Key},$$
$$RL_4 \leftarrow RL_3, RM_4 \leftarrow RM_3, RR_4 \leftarrow SR_3 \bigoplus S_{Key}$$

4) *ROUND 4:* In this round, all the blocks are Shifted Left by 8-Bits. The binary equivalent of each block of 16-bits is shifted towards left by 8-bits. This gives new blocks as:-

$$SL_5 \leftarrow ShiftLeft(SL_4),\ SM_5 \leftarrow ShiftLeft(SM_4),$$
$$SR_5 \leftarrow ShiftLeft(SR_4),$$
$$RL_5 \leftarrow ShiftLeft(RL_4),\ RM_5 \leftarrow ShiftLeft(RM_4),$$
$$RR_5 \leftarrow ShiftLeft(RR_4)$$

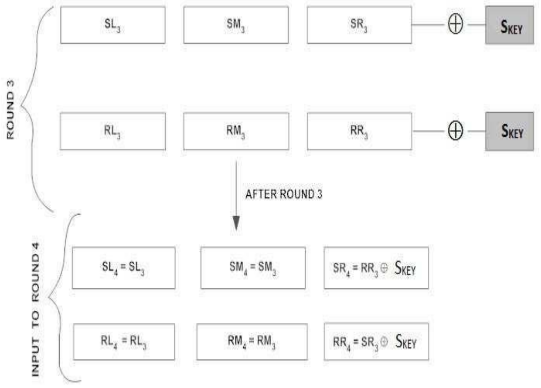


**Figure 3: Round 3**

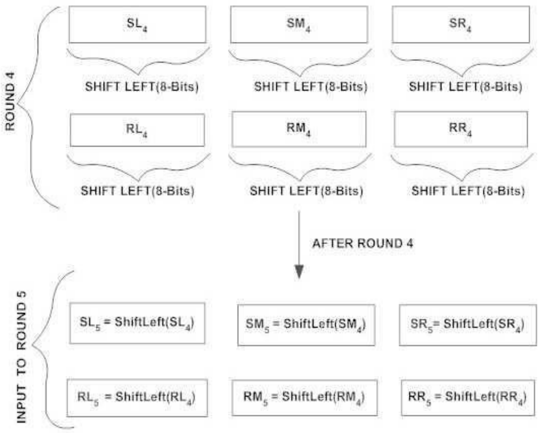


**Figure 4: Round 4**

5) *ROUND 5:* After shifting 8-Bits of all the blocks, the $SM_5$ is XORed with $RR_5$ and $SR_5$ is XORed with $RM_5$ to produce $SR_6$ and $RR_6$ respectively. This is done as :

$$SL_6 \leftarrow SL_5\ ,\ SM_6 \leftarrow SM_5\ ,\ SR_6 \leftarrow SM_5 \bigoplus RR_5,$$
$$RL_6 \leftarrow RL_5\ ,\ RM_6 \leftarrow RM_5\ ,\ RR_6 \leftarrow SR_5 \bigoplus RM_5$$

The following is the figure showing the process undertaken in round 5.

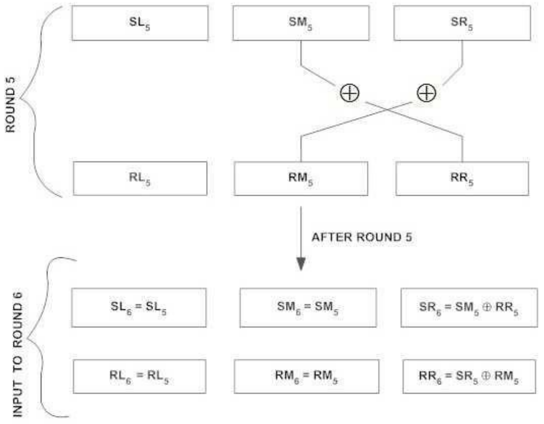


**Figure 5: Round 5**

6) *ROUND 6:* The output from round 5 is processed through round 6. In this round, $SL_6$ is XORed with $RM_6$ and $SM_6$ is XORed with $RL_6$ to give$SM_7$ and $RM_7$ as:

$$SL_7 \leftarrow SL_6, SM_7 \leftarrow SL_6 \bigoplus RM_6, SR_7 \leftarrow SR_6,$$
$$RL_7 \leftarrow RL_6, RM_7 \leftarrow SM_6 \bigoplus RL_6, RR_7 \leftarrow RR_6$$

These notations are represented in figure 6.

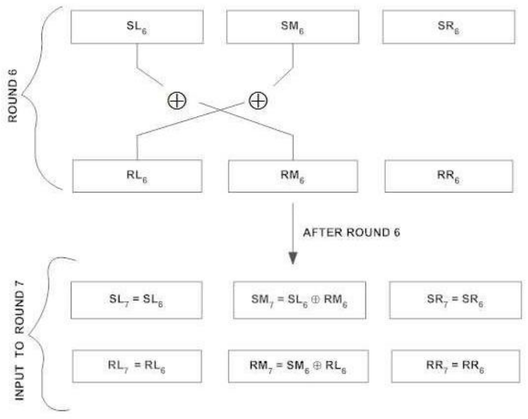


**Figure 6: Round 6**

7) ROUND 7: In this round, the shared key is used again. Here, shared key Skey is XORed with SL$_7$ and RL$_7$. The calculated values are swapped in order to generate SL$_8$ and RL$_8$ respectively, keeping other values same as previous round. The results are represented in the form of following notations:

$$SL_8 \leftarrow SL_7 \bigoplus S_{Key},\ SM_8 \leftarrow SM_7,\ SR_8 \leftarrow RR_7,$$
$$RL_8 \leftarrow RL_7 \bigoplus S_{Key}\ ,\ RM_8 \leftarrow RM_7\ ,\ RR_8 \leftarrow SR_7$$

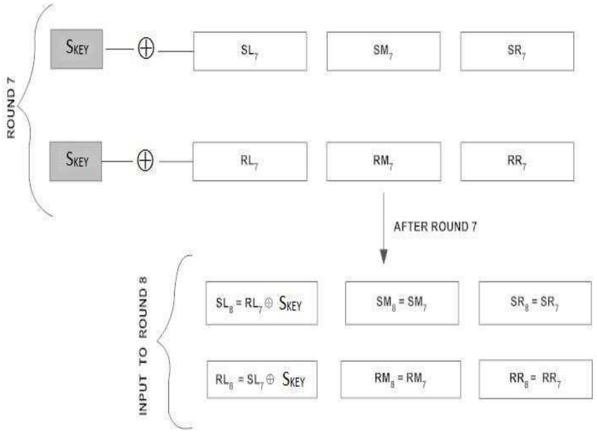


**Figure 7: Round 7**

8) ROUND 8: In this round, all blocks are Shifted Right by 8-bits each. The binary equivalent of each 16-bits block is shifted towards right by 8-bits. This procedure is represented by the following notations:

$$SL_9 \leftarrow ShiftRight(SL_8)\ ,\ SM_9 \leftarrow ShiftRight(SM_8)\ ,$$
$$SR_9 \leftarrow ShiftRight(SR_8),$$
$$RL_9 \leftarrow ShiftRight(RL_8)\ ,\ RM_9 \leftarrow ShiftRight(RM_8)\ ,$$
$$RR_9 \leftarrow ShiftRight(RR_8)$$

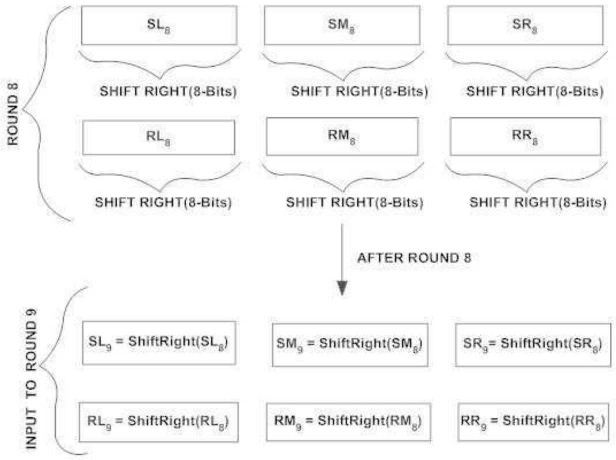


**Figure 8: Round 8**

9) ROUND 9: In this round, the values of blocks are interchanged with their adjacent blocks in the given manner. The figure shows the interchange of values in the given direction which results in:

$$SL_{10} \leftarrow SM_9,\ SM_{10} \leftarrow SR_9\ ,\ SR_{10} \leftarrow SL_9,$$
$$RL_{10} \leftarrow RR_9,\ RM_{10} \leftarrow RL_9\ ,\ RR_{10} \leftarrow RM_9$$

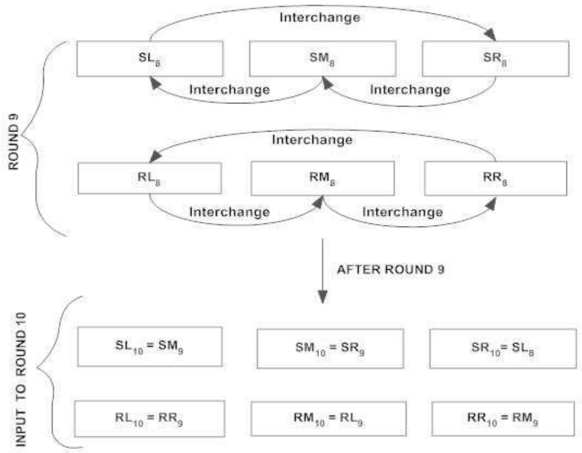


**Figure 9: Round 9**

10) ROUND 10: Round 10 includes the XORing of all the S.MAC blocks with their respective R.MAC blocks to generate the Final 6-Byte key. The final key F$_{Key}$ is also calculated in three parts with each part having 16-bits. F$_{Key}$ is calculated using the technique presented in figure 10. After these 10 Rounds of algorithm, we will get the final key F$_{ey}$ which will be used for both, encryption and decryption at sender's and receiver's ends respectively. The flow of the algorithm is shown in figure 11.

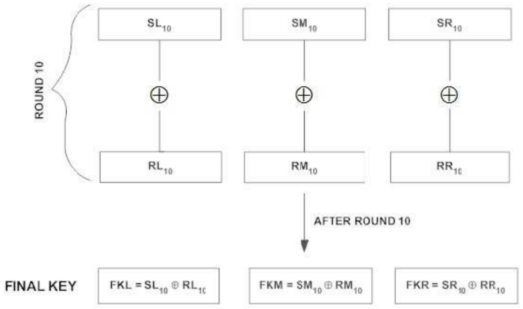


**Figure 10: Round 10**

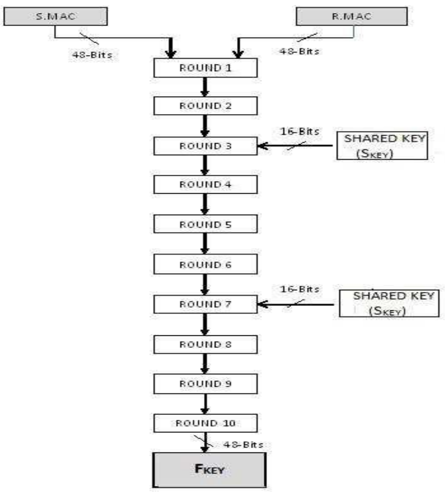
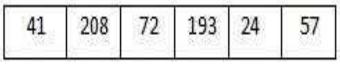


**Figure 11: Flow process of algorithm**

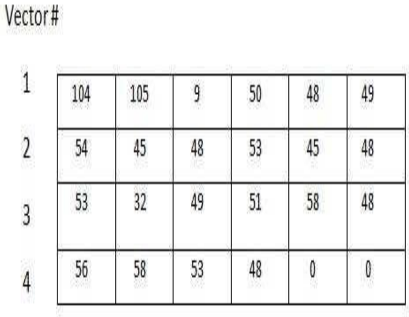
### C. Encryption of Data

After establishing the communication session, the sender will calculate the 6-Byte $F_{ey}$ to use it as the key to encrypt data. For example, the six parts of the key will represent a vector(chromosome) of 6-Bytes in decimal as:



| | | | | | |
|---|---|---|---|---|---|
| 41 | 208 | 72 | 193 | 24 | 57 |

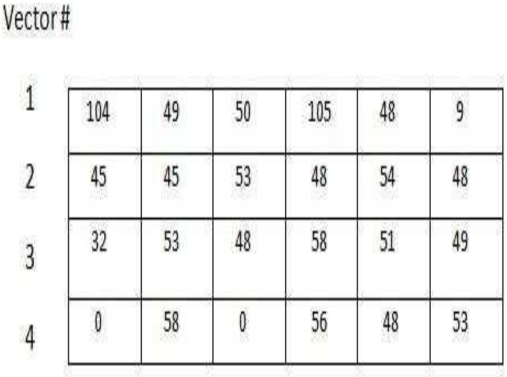
The digital data file will be treated as a set of N-Bytes. The data(d) is read and is concatenated with a time stamp $T_S$. This data along with time stamp is split into a matrix of four (N/6)vectors (chromosomes) of 6-bytes each(i.e. each matrix has 24-Bytes). For example, the source data file(with d-bytes) along with $T_S$ has 24-Bytes represented in vectors as:



| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 104 | 105 | 9 | 50 | 48 | 49 |
| 2 | 54 | 45 | 48 | 53 | 45 | 48 |
| 3 | 53 | 32 | 49 | 51 | 58 | 48 |
| 4 | 56 | 58 | 53 | 48 | 0 | 0 |

Now, the technique performs three main operations in four rounds on the data vectors. First, it performs crossover over the data bytes, then it performs mutation followed by another round of crossover, and finally, re-sequencing to give the cipher text. These rounds are performed as:

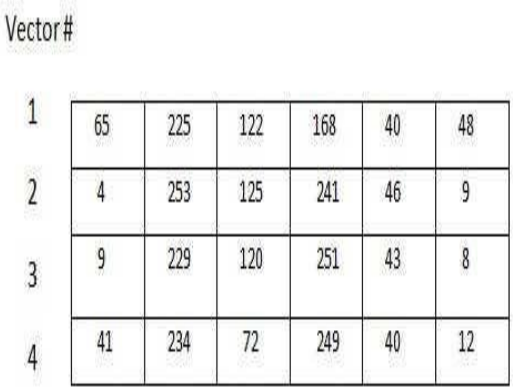
*1) Round 1 - Cross Over: CrossOver the gene order in each vector by using Pseudo Random Number Generation Algorithm with different seed values for each vector (vector number and Skey are used as two seeds in this work). After performing this operation on the original matrix, the new vectors are as follow:*



| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 104 | 49 | 50 | 105 | 48 | 9 |
| 2 | 45 | 45 | 53 | 48 | 54 | 48 |
| 3 | 32 | 53 | 48 | 58 | 51 | 49 |
| 4 | 0 | 58 | 0 | 56 | 48 | 53 |

**Figure 12: Data Vectors after CrossOver**

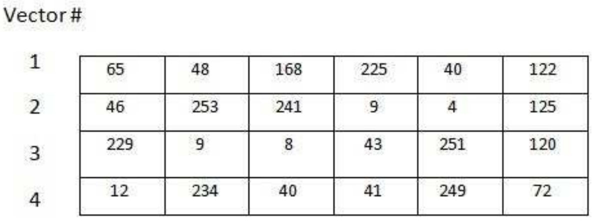
*2) Round 2 - Mutation: Mutation of each gene in each vector is done by applying eXclusive-OR(XOR) over each vector and Fkey. After applying mutation, the data vectors become as:*



| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 65 | 225 | 122 | 168 | 40 | 48 |
| 2 | 4 | 253 | 125 | 241 | 46 | 9 |
| 3 | 9 | 229 | 120 | 251 | 43 | 8 |
| 4 | 41 | 234 | 72 | 249 | 40 | 12 |

**Figure 13: Data Vectors after Mutation**

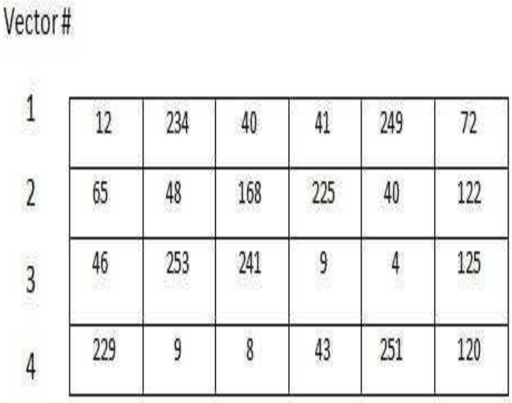
*3) Round 3 - Cross Over: After mutation of data vectors, another round of crossover is applied by using same pseudo random number generator algorithm(with same seeds, i.e., vector number and Skey )as applied in round 1. The new data vectors becomes:*



| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 65 | 48 | 168 | 225 | 40 | 122 |
| 2 | 46 | 253 | 241 | 9 | 4 | 125 |
| 3 | 229 | 9 | 8 | 43 | 251 | 120 |
| 4 | 12 | 234 | 40 | 41 | 249 | 72 |

**Figure 14: Data Vectors after CrossOver**

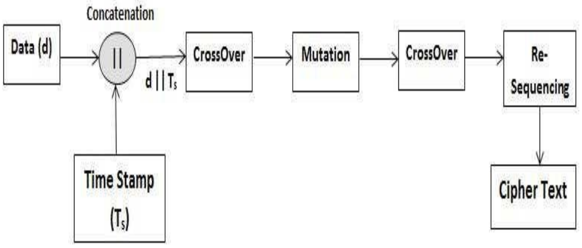
*4) Round 4 - Re-sequencing: This round reorders the sequence of the vectors to make more distortion in the encrypted data vectors. It re-sequence the vectors randomly using pseudo random number generation algorithm. The encrypted data vectors are as follows:*

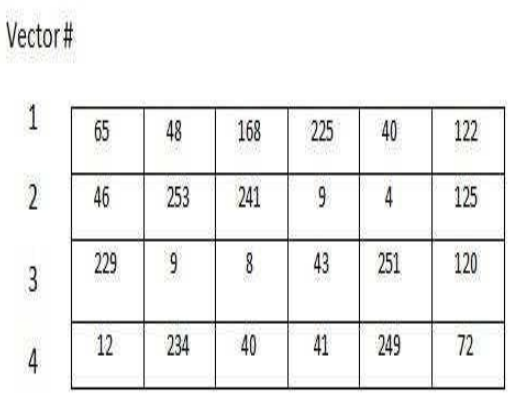


| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 12 | 234 | 40 | 41 | 249 | 72 |
| 2 | 65 | 48 | 168 | 225 | 40 | 122 |
| 3 | 46 | 253 | 241 | 9 | 4 | 125 |
| 4 | 229 | 9 | 8 | 43 | 251 | 120 |

**Figure 15: Data Vectors after Re-Sequencing**

The above vectors are encrypted data vectors which are send to the receiver. The transformation of data into cipher text is shown as below:





| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 65 | 48 | 168 | 225 | 40 | 122 |
| 2 | 46 | 253 | 241 | 9 | 4 | 125 |
| 3 | 229 | 9 | 8 | 43 | 251 | 120 |
| 4 | 12 | 234 | 40 | 41 | 249 | 72 |

**Figure 16: Data Vectors after Reverse Re-Sequencing**

## V. SECURITY ANALYSIS OF PROPOSED ALGORITHM

### A. Brute Force Attack

In sharing key phase, the sender and receiver shares seventeen 16-bit keys M[i] such that $1 \leq i \leq 17$, amongst which sixteen keys( M[2] to M[17]) are used to calculate shared key, based on one extra operational key M[1].Each of these keys is a 16-bit random number , and therefore, each key has $2^{16}$ possibilities. As there are seventeen such keys, hence the total number of possibilities of all these sixteen keys are {$2^{16}$ x $2^{16}$ x $2^{16}$ . . . . upto - 17 times} = $2^{272}$ which is very high to crack in limited time.

### B. Timing Attack/Replay Attack

The timing attack includes the delaying of messages or manipulation of time at which the message was sent. The replay attack is the retransmission of the message that were intercepted. This algorithm sends a time stamp $T_S$ along with the data. Hence, the receiver can easily detect any delays or change in the time stamp due to retransmission of messages or replay attack.

### C. Eavesdropping Attack

This attack refers to the interception of messages exchanged between communicating parties from communication channels. Even if attacker captures the data, he would not be able to decrypt it as only sender and receiver knows the final key. This final key depends on the seventeen shared keys as well as the MAC addresses of sender and receiver. These shared keys can only be decrypted by the receiver as only receiver has the private key to decrypt it.

### D. Cipher Text Only Attack

In this algorithm, the encryption algorithm and the cipher-text are known to the cryptanalyst. The cryptanalyst does not know the secret key used to encrypt or decrypt the shared key. Hence, the cryptanalyst cannot generate the final key which will be used for encryption and decryption. The cryptanalyst cann ot find any relation between the key and the ciphertext as different keys are used to encrypt the shared key which is further used to generate final key.

### E. Chosen Plain Text Attack

In chosen plain text attack, encryption algorithm is known along with the plain text message chosen by cryptanalyst, together with its corresponding cipher text generated with the secret key. In the proposed algorithm, the intruder or eavesdropper does not know the secret key of the sender, and therefore cannot generate the cipher text. Only the sender can encrypt the data as the shared key(shared using ElGamal algorithm) used for encryption can only be decrypted by either the sender or receiver.

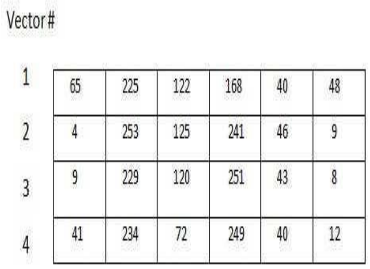
### F. Chosen Cipher Text Attack

In chosen cipher text, the encryption algorithm is known and cipher text is chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key. This secret key(shared key $S_{key}$) used for decryption is only known to the receiver. This shared key can only be decrypted either by the receiver or the sender.

The keys used here are generated by random number generator algorithm and to find these random numbers is itself a difficult task for the attacker. Moreover, we have used Pseudo random numbers in the encryption and decryption phases using the pseudo random number algorithm which uses the shared key and vector numbers as seeds. The number of possibilities for bruteforcing the shared key is $2^{272}$ . Hence, detecting the pseudo random numbers is also very difficult.

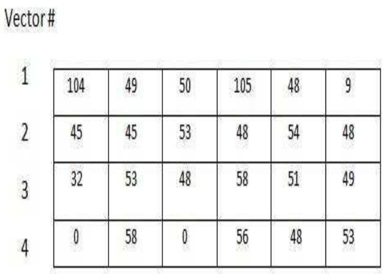
## VI. CONCLUSION

In this paper, we present an identity based encryption algorithm. This algorithm uses hybrid encryption - Elgamal key exchange algorithm for exchanging seventeen 16-bit keys which, along with MACs of both the users, will generate a shared key. This shared key will undergo a 10-round process for generating a final key which will be used for both encryption and decryption process. This algorithm eliminates the use of KDC (Key Distribution Center) or any third party for sharing key and therefore, can be used in an organization's LAN OR WAN. As it is using pseudo random numbers and MAC address of both users(unique identity), therefore, it is highly secured. This algorithm can work on any device that has a MAC address like computer system, mobile phones, routers, etc. Hence, this algorithm can be implemented up to network layer.



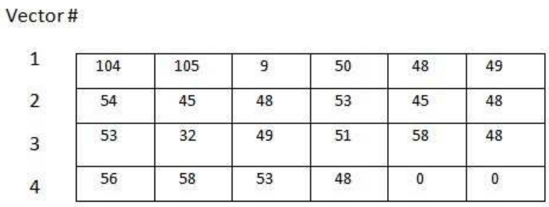
| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 65 | 225 | 122 | 168 | 40 | 48 |
| 2 | 4 | 253 | 125 | 241 | 46 | 9 |
| 3 | 9 | 229 | 120 | 251 | 43 | 8 |
| 4 | 41 | 234 | 72 | 249 | 40 | 12 |

**Figure 17: Data Vectors after Reverse CrossOver**



| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 104 | 49 | 50 | 105 | 48 | 9 |
| 2 | 45 | 45 | 53 | 48 | 54 | 48 |
| 3 | 32 | 53 | 48 | 58 | 51 | 49 |
| 4 | 0 | 58 | 0 | 56 | 48 | 53 |

**Figure 18: Data Vectors after Reverse Mutation**



| Vector# | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 104 | 105 | 9 | 50 | 48 | 49 |
| 2 | 54 | 45 | 48 | 53 | 45 | 48 |
| 3 | 53 | 32 | 49 | 51 | 58 | 48 |
| 4 | 56 | 58 | 53 | 48 | 0 | 0 |

**Figure 19: Original Data Vectors after Reverse CrossOver**

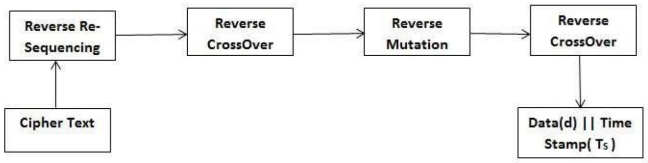


**Figure 20: Flow of Decryption Process**

## REFERENCES

1. W. P. Petkovic, M. Jonker, “Special issue on secure data management,” Journal of Computer Security, 17(1), pp. 1–3, 2009.
2. A. Kahate, CRYPTOGRAPHY AND NETWORK SECURITY. Tata- McGraw-Hill, 2nd edition, 2008.
3. A. S. Tanenbaum, Computer Networks. Prentice Hall PTR,Pearson Education LTD.l,4th edition.
4. A. Shamir, “Identity-based cryptosystems and signature schemes,” Proc. of Advances in Cryptology - Crypto’84, LNCS 196, Springer-Verlag,1984, pp. 47–53.
5. D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” Advances in Cryptology-CRYPTO’01,LNCS 2139,Springer-Verlag,2001, pp. 213–239.
6. C. Cocks, “An identity based encryption scheme based on quadratic residues,” Proc. of Advances in Cryptography and Coding 2001, LNCS 2260, Springer-Verlag, 2001, pp. 360–363.
7. D. Boneh and X. Boyen, “Efficient selective-id identity based encryption without random oracles .” Proc. of Advances in Cryptology Eurocrypt 2004, LNCS 3027, Springer-Verlag, pp. 223–238, 2004.
8. X. B. D. Boneh and E. Goh, “Hierarchical identity based encryption with constant size ciphertext,” Proc. of Advances in Cryptology – Eurocrypt 2005, LNCS 3494, Springer-Verlag, 2005, pp. 440–456.
9. B. Waters, “Efficient identity-based encryption without random ora- cles,” Proc. of Advances in Cryptology - Eurocrypt 2005, LNCS 3494, Springer-Verlag, 2005, pp. 114–127.
10. X. H. Lei M., Z. Qi and S. V. Vrbsky, “Protecting location privacy with dynamic mac address exchanging in wireless networks,” Proc. of the 2007 Intelligence and Security Informatics (ISI’07), New Brunswick, New Jersey,USA IEEE, May.
11. U. P. Gauenther Lackner and P. Teufl, “Combating wireless lan mac-layer addressspoofing with fingerprinting methods,” International Jour- nal of Network Security, vol. 9,No.2, pp. 164–172, Sept. 2009.
12. M. B. J. Hall and E. Kranakis, “Radio frequency fingerprinting for intrusion detection in wireless networks,” IEEE Transactions on Dependable and Secure Computing, 2006.
13. J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker, “Passive data link layer 802.11 wireless device driver fingerprinting.” in USENIX Security Symposium, vol. 3, 2006, pp. 16–89.
14. U. P. G. Lackner, M. Lamberger and P. Teufl, “Wifi fingerprinting,” DACH Mobility 2006, Sep. 2006.
15. W. Stallings, Cryptography And Network Security Principles And Prac- tice. Pearson,Fifth Edition.
16. K. A. and Rajpal, “Application of genetic algorithm in the field of steganography,” Journal of Information Technology, 2(1), pp. 12–15, 2004.
17. R. N. Kumar A. and Tayal, “New signal security system for multimedia data transmission using genetic algorithms,” NCC05, January 20-28, IIT Kharagpur, pp. 579–583, 2005.
18. Husainy, “Image encryption using genetic algorithm,” Information Technology Journal, 5(3), pp. 516–519, 2006.
19. O. F. O. Tragha A. and F. Mouloudi, “Iciga: Improved cryptography inspired by genetic algorithms,” International Conference on Hybrid Information Technology (ICHIT 06), pp. 335–341, 2006.
20. N. S. Sania Jawaid, Anam Saiyeda, “Selection of fittest key using genetic algorithm and autocorrelation in cryptography,” Journal of Computer Sciences and Applications, vol. 3,No.2, pp.

## AUTHORS PROFILE

**Mr. Darpan Anand**, an alumnus of Dayalbagh Educational Institute, Agra (UP) holds M. Tech from Dayalbagh Educational Institute, Agra and is submitting Ph.D. in Information Security from Dr. A.P.J. Abdul Kalam Technical University, Lucknow. He worked as a lecturer in the Institute of Engineering and Technology, Dr. Bhim Rao Ambedkar University, Agra for 4 years. Post which he entered into the field of Software development in July 2008 and joined Infotech Enterprises Ltd, Noida as software Engineer. In July 2010, he worked for Airtel and IBM on a GIS-based Java platform. He is into Academics on July 2010. He was also worked as an Assistant Professor in the Department of Computer Science and Engineering at Sharada Gourp of Institutions. As researcher, he published various papers in Springer, IEEE, ACM, Elsevier, etc. Current, he is working as Associate Professor in the Department of Computer Science and Engineering in Chandigarh University, Punjab and his research field is Information Security, Computer Network, Software Defined Network, Machine Learning, etc.

**Dr. Jeetendra Pande** is working as an Assistant Professor of Computer Science Department at Uttarakhand Open University, Haldwani. His area of interest includes Component Based Software Development, Cyber Security, Open Education Practices and Software Engineering.

**Udit Maheshwari** received his degree of B.Tech in Computer Science & Engineering (2016) from Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh (India) where he worked on his research thesis on Genetic Cryptographic Algorithms. He joined Tata Consultancy Services as System Engineer in 2017 and has been working in the field of Information Security since then. He has been involved in various information security related projects including Vulnerability Assessment and Penetration Testing, Consultation for Risk Advisory and Security Analysis. His main area of expertise is Information Security and Cryptographic Algorithms. His current field of interest includes Quantum Cryptographic Algorithms.