A Way of Safeguard using Concept of Recurrence Relation and Fuzzy logic against Security Breach in Wireless Communication

*Anirban Bhowmik
State Aided College Teacher, Dept. of Computer Science,M.U.C. Women's College,
Purba Bardhaman, WB, India

Dr.Sunil Karforma
Department of Computer Science, The University of Burdwan, Purba Bardhaman, WB, India

Joydeep Dey
State Aided College Teacher & Former HoD, Dept. of Computer Science,
M.U.C. Women's College, Purba Bardhaman, WB, India

Dr.Arindam Sarkar
Department of Computer Science & Electronics, R.K.M. Vidyamandira, Belur Math, Belur, WB, India
*Corresponding author's e-mail:animca2008@gmail.com

ABSTRACT --Wireless networking plays an important role in public and military applications. Security of information transfer through wireless networks remains a challenging issue. Wireless security is the process of designing, implementing, and ensuring security on a wireless computer network. Jamming and eavesdropping are two primary attacks at the physical layer of a wireless network. This article offers a study on the security vulnerabilities and threats on wireless communication and an efficient comprehensive mechanism for improving the wireless network security. In this paper, a stream cipher based symmetric key encryption with recurrence relation and fuzzy based session key has been proposed for wireless communication by satisfying the key issues like security; increase the strength of symmetric key. In this article we generate 'n' number of sub keys from symmetric key using XOR operation between a random character matrix and symmetric key. In this article the random numbers are generated by using recurrence relation which is a new approach in random number generation in discrete mathematics. Among these sub keys we generate a session key using fuzzy function. Now the encryption is done by using this session key and symmetric key. Here we transmit the session key to the recipient end by amalgamating with the symmetric key. This amalgamated form is send to recipient end for decryption. Here a new authentication scheme is used. Different types of randomness test have been done to test the randomness of our session key. The Brute-force attack analysis for this scheme and comparative study with existing standard methods has been done with satisfactory results.

KEYWORDS: Wireless security, Symmetric key, Recurrence Relation, Random Character Matrix, Session key, Fuzzy logic, Encryption, Decryption.

1 Introduction

Wireless network have become a necessary part in our daily life. Now-a-days security is a vital issue in wireless application because wireless networks are heavily used for transmission of important or private information such as net banking service, e-shopping, bill payment etc. therefore, it is very important to share secret information reliably in the presence of eavesdroppers. Most commonly used security methods based on cryptographic technique are implemented at the upper layer of a wireless network. The existing physical layer security techniques can be classified into five major categories which are (i) theoretical secure capacity (ii) power (iii) code (iv)channel (v) signal detection approaches. In symmetric encryption or single-key encryption technique a common private key is shared by two users through a secure channel. Here physical layer method is employed to distribute secret keys to supply location privacy and to supplement upper layer security algorithms. It is very difficult for attacker to decipher transmitted information when physical security is applied.

Table 1. Classification of security attacks in wireless communication

Passive Attacks	Active Attacks
	Denial of service attack
	Resource consumption
Traffic Analysis	Masquerade attack
	Reply attack
	Information disclosure
Eavesdropping	Massage modification

In the following table summarizes the different types of attacks and different technologies against them with respect to basic security attributes.

Table 2. Attack methods with security technologies

Computer Security Attributes	Attack Methods	Technology for Internet Security
Confidentiality	Eavesdropping, Hacking, IP spoofing, phishing, DoS.	IDS, firewall, cryptographic system, IPSec, SSL
Integrity	Viruses, Worms, Trojans, Eavesdropping, Hacking, IP spoofing, phishing, DoS.	IDS, firewall, anti-malware s/w, IPSec, SSL
Privacy	E-mail bombing, Spamming, Cookies, IP spoofing.	IDS, firewall, anti-malware s/w, IPSec, SSL
Availability	E-mail bombing, Spamming, Cookies, IP spoofing, System boot record interface.	IDS, firewall, anti-malware s/w

In this article, we have given stream cipher based symmetric key encryption with session key to try to ensure security at physical layer in wireless network.

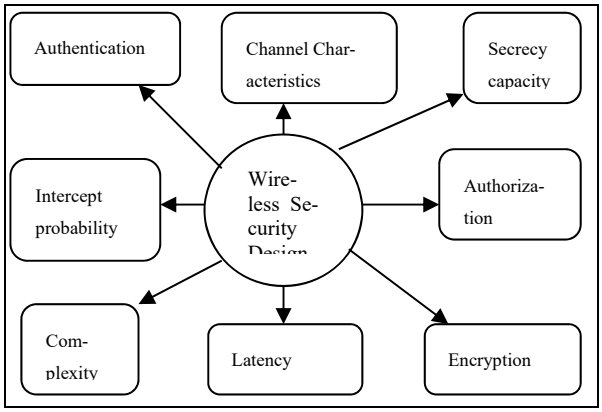

Fig1. Wireless security methodology and design factor.

A symmetric encryption scheme has five parts- i) plaintext ii) encryption algorithm iii) secret key iv) cipher text iv) decryption algorithm [1], [3].

For secure use of symmetric encryption, we should focus on two requirements-

i) Strong encryption algorithm ii) secret key, sender and receiver must have the copies of this key in a secure fashion. The two basic building block of all encryption algorithms are substitution and transposition. There are two types of algorithms stream cipher and block cipher and four types of algorithm modes Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher feedback (CFB), Output Feedback (OFB).Many Symmetric encryption algorithms are: Data Encryption Standard (DES), Triple DES (3DES), International Data Encryption algorithm (IDEA), Blowfish, Advanced Encryption Standard (AES). Symmetric key algorithm is also known as private key algorithm.

1.1 *RECURRENCE RELATION IN RANDOM NUMBER GENERATION*: - Each term of a sequence is a linear function of earlier terms in the sequence is linear recurrence. Recurrence relation is of two types- 1) linear recurrence relation 2) linear non homogeneous recurrence relation [21].

1) Linear recurrence relation:- A linear homogenous recurrence relation of degree k with Constant coefficients is a recurrence relation of the form

$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$, where $c_1, c_2, \ldots, c_k$ are real numbers, and $c_k \neq 0$. a_n is

expressed in terms of the previous k terms of the sequence.

Proposition: Let $a_n = c_1 a_{n-1} + c_2 a_{n-1} + \ldots + c_k a_{n-k}$ be a linear homogeneous recurrence.

i. Assume the sequence a_n satisfies the recurrence.

ii. Assume the sequence a'_n also satisfies the recurrence.

iii. So, $b_n = a_n + a'_n$ and $d_n = \propto a_n$ are also sequences that satisfy the recurrence. ($\propto$ is any constant).

2) Linear non-homogeneous recurrence: - A linear non-homogenous recurrence relation with constant coefficients is a recurrence relation of the form

$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k} + f(n)$, where $c_1, c_2, \ldots, c_k$ are real numbers, and f(n) is a function depending only on n. The recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$, is called the associated homogeneous recurrence relation.

Proposition:

i. Let $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k} + f(n)$ be a linear non homogeneous recurrence.

ii. Assume the sequence b_n satisfies the recurrence.

iii. Another sequence a_n satisfies the non homogeneous recurrence if and only if $h_n = a_n - b_n$ a sequence that satisfies the associated homogeneous recurrence is also.

In our article we use linear non homogeneous recurrence relation for generating random numbers. This is another concept to generate random numbers without using rand (). Any recurrence relation can be used in random number generation. For example here we use the recurrence $a[i] = c[0] * a[i-1] + c[1] * a[i-2] + c[2] * a[i-3] + \ldots + c[m] * a[i-m+1] + pow(2, i)$. The initial condition and coefficient values are given at first, according to these conditions and values the random numbers are generated.

1.2 PSEUDORANDOM FUNCTION (PRF): - A PRF is used to produce a pseudorandom string of bits of some fixed length. The PFR takes as input a seed and some context specific values. In our technique strings of bits are not fixed and dependent on user that is the total number of random numbers is chosen by user as well as context specific values also. Here recurrence relation formula is used in PRF [3].

1.3 FUZZY LOGIC: - Fuzzy logic [4], [5], [6], [7] deals with fuzzy predicates and fuzzy implications made up of fuzzy predicates. It also deals with how to make inferences using fuzzy predicates and fuzzy implications. A fuzzy predicate is described in terms of fuzzy sets and a fuzzy implication is described in terms of fuzzy relations. Fuzzy relations are special kind of fuzzy sets whose domains are Cartesian products of domain. It is also needed in the compositional form of reasoning. Fuzzy sets were introduced by Prof. Lotfi A. Zadeh of University of California at Berkeley [7]. A fuzzy set on a universal domain U is defined by its membership function from U to [0, 1]. Thus by a fuzzy set on U is meant a function $\boldsymbol{A}: \boldsymbol{U} \rightarrow [\mathbf{0}, \mathbf{1}]$.'A' is called the membership function, A(x) is called the membership grade of x. we can write $\boldsymbol{A} = \{\boldsymbol{x}, \boldsymbol{A}(\boldsymbol{x})): \boldsymbol{x} \boldsymbol{\epsilon} \boldsymbol{U}\}$. It deals with reasoning with inexact or fuzzy concept. The fuzzy logic encompasses the fuzzy relations and fuzzy sets and [0, 1] is its truth value set. Most of the fuzzy logic is based on the following definitions for the logical connectives ¬, ∨, and ∧.

$$T(p \vee q) = max[T(p), T(q)], T(p \wedge q) = min[T(p), T(q)], T(\sim p) = 1 - T(p).$$

Fuzzy Membership Functions: - All information contained in a fuzzy set is described by its membership function. The features of this function are given below.

The *core* of a membership function for some fuzzy set $A_{\sim}$ is defined as a region of the universe which is characterized by complete and full membership in the $A_{\sim}$. So, the core comprises those elements *x* of the universe such that $\mu_{A\sim}(x) = 1$.

The *support* of a membership function for some fuzzy set $A_{\sim}$ is defined as a region of the universe that is characterized by nonzero membership in the $A_{\sim}$. So, the support comprises those elements *x* of the universe such that $\mu_{A\sim}^{(x)} > 0$.

The *boundaries* of a membership function for some fuzzy set $A_{\sim}$ are defined as a region of the universe containing elements that have a nonzero membership but not complete membership. That is, the boundaries comprise those elements *x* of the universe such that $0 < \mu_{A_{\sim}}^{(x)} < 1$. These elements of the universe are those with some *degree* of fuzziness, or only partial membership in the fuzzy set$A_{\sim}$. In our scheme, fuzzy concept is used to generate session key. The membership function is chosen based on intuitive understanding of the problem definition [4], [5],[6].Here we deduce the following function $f(x) = (half\ the\ size\ of\ the\ symmetric\ key/ total\ ascii\ bit\ diff(x)), 90 < x <= 255$. This proposed function satisfies all the features of membership function. A diagram is given below to represent the characteristics of our fuzzy membership function.

2 Literature Survey

Now-a-days the data Security has become a serious matter with the progress of communication technology. In the symmetric key encryption, DES was adopted as national standard in 1976. Besides DES, two variations of DES have emerged which are double DES and triple DES where two keys and three keys are used to increase the robustness of encryption. IDEA [3], RC4, RC5, BLOWFISH, TWOFISH [3] are different types of symmetric key encryption algorithm. National Institute of Standards and Technology (NIST) announced the Advanced Encryption Standard (AES), in 2001. AES algorithm is a symmetric block cipher with low complexity and high security level. NIST also proposed Secure Hash Algorithm (SHA) for authentication. When new encryption technique is introduced, cryptanalysts starts to develop to attack. Eli Biham and Adi Samir introduced the concept of differential cryptanalysis [8]. This method looks at pairs of cipher text whose plain texts have particular differences. Mitsuru Matsui invented the linear cryptanalysis attack [10] based on linear approximation. Timing attack is also applied on symmetric key encryption. There also exists Sensor Network Encryption Protocol (SNEP) [2] for security of sensor network systems. Chanchala Joshi and Umesh Kumar Singh [17] focuses on information security risk assessment, prioritizes the information assets and identification and monitoring of specific threats. This also suggests a conceptual framework of info-structure of ISRA. Thus many encryption algorithms are widely available and used in information security and also different types of attacks are available to break the security. In Symmetric keys (or private key) encryption or secret key encryption, only one key is used to encrypt and decrypt data. DES uses one 56-bits key. Double DES uses two 56 bits key and Triple DES (3DES) uses three 56- bits keys. While AES uses various (128,192,256) bits keys. At present different types concepts, logic like fuzzy logic, neural network etc. also introduced in cryptography for increasing the robustness encryption. Our paper proposed a technique called FSKRPSKE which provides a fuzzy based session key from symmetric key using a random character key matrix. Here session key is generated using fuzzy logic. Using these two keys we can encrypt a file (.txt, .doc, etc) and by the reverse way we can decrypt the cipher text to get plain text.

3 Present Problem Scenarios in Wireless Communication

In wireless network, the data and information are exchanged among different authorized users, but this process is vulnerable to various bitchy threats. Hence, it is paramount importance to improve wireless communication security to fight against different types of attacks like eavesdropping DoS, cyber-criminal activities. The main wireless security methodologies include the authentication, authorization and encryption. Cryptography is used for encryption process and also it improves the achievable communication confidentiality. But it requires additional computational power and robustness in encryption and decryption process. In symmetric key encryption we can transmit huge amount of data between sender and receiver effectively through wireless communication. But the whole encryption is done using a private key (symmetric key). If this private key is revealed by attackers then overall communication is under threat. Existing symmetric key encryption algorithm does not change their key/keys with respect to time. So the use of a single fixed key or multiple fixed keys is a problem in encryption process. Thus security in wireless communication is very necessary because of its more and more use in Smartphone, online banking, e-shopping etc.

4 Solution Domain in and Objectives

In this paper we are motivated to discuss diverse wireless attacks and the corresponding defense mechanisms. Wireless networks generally adopt the OSI protocol architecture. The security threats and vulnerabilities associated with the OSI protocol are protected separately at each layer to meet the security requirements like authenticity, confidentiality, integrity and availability. Different types of cryptographic techniques are widely used to protect threats and attacks in wireless communication. In this article our objective is to propose a cryptographic technique which provides confidentiality and authenticity. The technique is mainly divides into three phases; key generation phase, encryption phase, authentication phase and decryption phase. In key generation phase, we generate an extra key called session key [4] from the symmetric key by using a random character matrix, fuzzy logic. Since this session key may change time to time so the use of both session key and symmetric key and CLS operation in encryption provides the extra robustness in our technique. Here we deduce a novel technique for authentication proof by using both the symmetric key and session key. Thus, the use of session key with symmetric key and an authentication cum encryption provides the added flavor and beauty in our proposed technique.

5 Methodology

Our proposed technique is composed of four parts which are (i) Session key generation (ii) Encryption with symmetric key and session key (iii) Authentication check and session key transpired. iv) Decryption. For session key generation to crate random numbers we use the formula for non homogeneous recurrence relation. The summary of our scheme is described by a compact algorithm, given below.

ALGORITHM:

Input: - plain text, symmetric key.

Output: - encrypted file with header and tailer.

Method: -

1. `Call RCG ()` // Random character generation using non homogeneous recurrence relation
2. `Call MGA ()` // matrix is generated to create 'n' number of key population from symmetric key.
3. `Call SKG ( )` // generate session key using 'n' number of key populations and fuzzy logic.
4. `Call EP ( )` // Encryption Process i.e., cipher file is generated using two keys.
5. `Call Create_Header_Tailer ( )` // header and tailer structure is created using two keys with XOR operation.
6. `Call Concate(header, encrypted file,tailer)` // total structure is created and it is ready for transmission over network.
7. `Call AuthenticationCheck ( )` // check authentication using two keys and also generate session key using symmetric key.
8. `Call DrypPhase ( )` // plain text is generated.

All the above methods in the algorithm are described below in details.

5.1 SESSION KEY GENERATION PHASE:

The session key generation process is divided into two parts. First is pre defined matrix generation and second is session key generation from symmetric key using fuzzy logic.

In the first part the predefined matrix is a square matrix with random characters, this random characters are generated using non homogeneous recurrence relation. A details algorithm is given below for random character generation. The number of column of matrix is half of the size of the symmetric key. If the key size is 'n' byte then no. of row and column of matrix is n/2.

ALGORITHM-1: *Random Character Generation using Non-Homogeneous Recurrence Relation (RCG)*

Input: seed value, coefficient value and non homogeneous recurrence equation.

Output: random character value.

Method:

```
1. Set i,j,m,n,f,lr as integer and a[m],r[n],c[m]as integer array.
2. n<- total random number
3. m<- totalno.of coefficient in non-homogeneous recurrencerelation.
4. For i=0 to m
5. c[i]<- get_coeff()
6. a[i]<-get_seedVal()
   end for
7. lr<- get_largestPrimeFact(a[2] xor c[3]).
         {/* a[2] and c [3] are chosen by the user.*/}
8. a[0]<- a[0] xor lr.
9.  for i=1 to m
10. a[i]=(a[i]xor a[i-1])xor lr
    end for
11. for i=3 to n
```

```
12. a[i]<-get_val(rec_Funct(i))
13. if(a[i]<0)
14. a[i]<- -a[i]
    end if
15. r[i]<-a[i]
16.  f<-((((a[i]xor c[1])xor c[3])xor c[5])xor...c[m])
17. a[i]<-f
        end for
18. if(n>=3)
       r[n]<-get_shuffle(r[n])
       end if
19: End
```

ALGORITHM-2: *Matrix Generation Algorithm (MGA)*

Input: - randarr[m]: character array.

Output: - a square matrix (kmatrix[m][m]).

Method: -

```
1.Set m, i and j as integer.
2. m= half(symmetric key size).
3. kmatrix[m][m]={0}
4. for i=0 to m
5.  randarr[i]= get_randomchar();
6. end for
7. for i=0 to m
8.   for j=0 to m
9.    kmatrix [i][j]=randarr[j]
10.  end for
11. randarr[i]←rightShft(randarr[i])
12. end for
13.End.
```

ALGORITHM-3: *Session Key Generation (SKG*)

Input: - symmetric key and kmatrix[x][y]

Output: - session key (SK[n]).

Method: -

```
1. Set i, j, row, col,m fval as integer.
2. Set m= length (symmetric key), tmp[m/2][m/2]= {0},
 SYK[m] = symmetric key and keyarr[m/2],SK[n] as character array.
3. row←get_row( kmatrix[m/2][m/2])& col← get_colmn(kmatrix[m/2][m/2])
 {/*row=col=m/2*/}
 {/* step 4 to step 8 describes key population*/}
4. for i=0 to row do
5.   for j=0 to col do
6.   tmp[i][j]←bitwise_XOROP( SYK[j],kmatrix[i][j])
7.   end for
8. end for
 {/* following part of algorithm find the fittest key among m number
  of key population using fuzzy logic.*/}
9. Set fval=0
```

```
10. for i=0 to row do
11.   fval+= bit_Difference (SYK [col], tmp[i][col])
12.     if ((col/fval)<10/m) then
13.       SK[n]←tmp[i][col]
14.        Keyarr[col]=kmatrix[i][col]
15.     end if
16. end for
17. End
```

5.2 ENCRYPTION PHASE:

Now in our proposed technique encryption is done by using symmetric key and session key. The encryption with session key provides extra flavor of robustness. In both cases XOR [9] operation is executed. The encryption algorithm is given below.

ALGORITHM-1**:** *Encryption Process (EP)*

Input: - plain text, symmetric key, session key.

Output: - encrypted file.

Method: -

```
1. Set file_Plain as plain text file and file_Cipher as cipher text file.
2. Set file_Output as temporary file.
3. if ( !eof ) then
4. file_Output= bit_XOROP ( file_Plain , session key)
5. file_Cipher= bit_XOROP ( file_Output , symmetric key)
   end if
6. End
```

After encryption with two keys we create a format with Header, cipher text and Tailer [13] using the function *Concate()*. The result of this function is the compact form of text which is ready for transmission to the receiver end. We use Tailer part to check authentication and Header part for session key generation in recipient end. Now the Header and Tailer structure is created using the following algorithm.

HEADER AND TAILER CREATION: -

ALGORITHM-2**:** *Create_Header_Tailer ( )*

Input: - symmetric key, session key (SK[m]).

Output: - Header and Tailer.

Method: -

```
1. Set F_half, L_half and diagEl as character arrays.
2. Set key_Mat [][] as 2D character array.
3. F_half← first half of symmetric key, and L_half← last half of symmetric
key.
4. set m= ascii valueOf(1st character of symmetric key)
5. Header← ((keyarr[] XOR L_half))<< (m mod length(symmetricKey)).
6. Key_Mat← Call create_matrix (F_half, session key)
7. ColmnEl← get_2ndColmn (key_Mat)
8. Tailer← bit_XOROP (ColmnEl, L_half)  // diagEl is XORed with L_half, bit by bit up
   to the last bit of L_half.
9. End
```

If symmetric key is 16 byte the session key is 8 byte and the total structure is given below which is created by calling the function Concate ( ) which is given in main algorithm.

Header (8byte	Encrypted file	Tailer (8 byte)

5.3 DECRYPTION PHASE:

The decryption phase is occurred in recipient end, first of all, Header section, encrypted file and Tailer section are separated using the symmetric key. Here we call *Create_Header_Tailer ()* function so that we can reveal the session key using the symmetric key from Header section and we can check the authentication from Tailer part using the function *AuthenticationCheck()*. If authentication phase shows green signal then plain text is generated from encrypted file using both session key and symmetric key in reverse process of encryption phase.

5.4 *SIGNIFICANCE OF AUTHENTICATION:* **-** Authentication mechanisms [8] provide the proof of identities. The authentication process ensures that the origin of document is correctly identified i.e, the document is coming from right user. In our scheme we use authentication part for proof of identities. We know that symmetric key encryption provides authentication and confidentiality. But we are qualified this statement using an extra authentication scheme in our proposed technique. Here we use two structures of Header and Tailer. There are complex calculations for Header and Tailer generation. Tailer structure is used for authentication check. In receiver side symmetric key and particular row are used to generate session key. Now using this session key and symmetric key we check identities of user from Tailer part. Thus our technique protects the fabrication.

6 Results and Discussions

Table 3. Specifications of H/w and S/w used in the experiment

Computer	Lenovo G80.
Processor	Intel® Pentium® CPU B950@210GHz
RAM	2GB
Compiler	Turbo C
Disc Drive	SA 9500325AS ATA
Operating System	Windows 7 Ultimate (32 Bits)

In this section, simulation results of the proposed technique are presented. All the programs and calculations are done in a machine with above configuration. In our experiments, several sizes of different types of files are used as plain text.

Our result section is divided into four parts: (i) Attacks Analysis (ii) Randomness test & Entropy test of session key (iii) Analysis of encryption technique and (iv) Comparison among standard algorithms and some other techniques.

DIFFERENT TYPES OF ATTACKS ANALYSIS: -

There are different types of attacks are exists to recover the key in use rather than simply to recover the plain text. There are two general approaches are – (i) Brute-force attack [11][12] (ii) Dictionary Attack [22].

6.1 BRUTE-FORCE ATTACK:-

A good encryption technique satisfies the requirements of resisting brute-force attack. In this attack, attacker tries to translate the cipher text into plain text using every possible key. On average, half of all possible keys must be tried to achieve success. In most networking system, algorithms are known to all so in this case, brute-force attack will impossible if the algorithm uses large number of keys. At present the fastest super computer is Tianhe-2 having speeded 33.86 petaflops i.e., 33.86×10^{15} floating point operations per second. Let us consider each trial requires 2000 FLOPS to complete one check. So number of trials complete per second is: 16.93×10^{12}. The number of second in a year is: 365*24*60*60=3153600 sec. Now from the above key space the formula for break the keys is $2^{3x/2}$/ ($16.93*10^{12}$ *3153600) =Y. So if k increase then Y increases. The following table and graphs shows the average time required for exhaustive key search [3].

Table4. Exhaustive Search analysis

Symmetric key size (x bits)	No. of Trials in standard algorithms(2^x)	Time Required(in years) at $16.93X10^{12}$ Decryption/s in standard algorithms	No. of Trails in our proposed technique ($2^{(3x/2)}$)	Time Required(years) at $16.93X10^{12}$ Decryption/s in proposed technique
56	2^{56}	0.001349	2^{84}	362289
64	2^{64}	0.34550	2^{96}	1483938
128	2^{128}	$6.3734x10^{18}$	2^{192}	$1.1775x10^{38}$
166	2^{166}	$1.7519x10^{30}$	2^{249}	$1.6943x10^{55}$
192	2^{192}	$1.1756x10^{38}$	2^{288}	$9.3148x10^{66}$
256	2^{256}	$2.1687x10^{57}$	2^{384}	$7.3799x10^{95}$
300	2^{300}	$3.8153x10^{70}$	2^{450}	$5.4454x10^{115}$

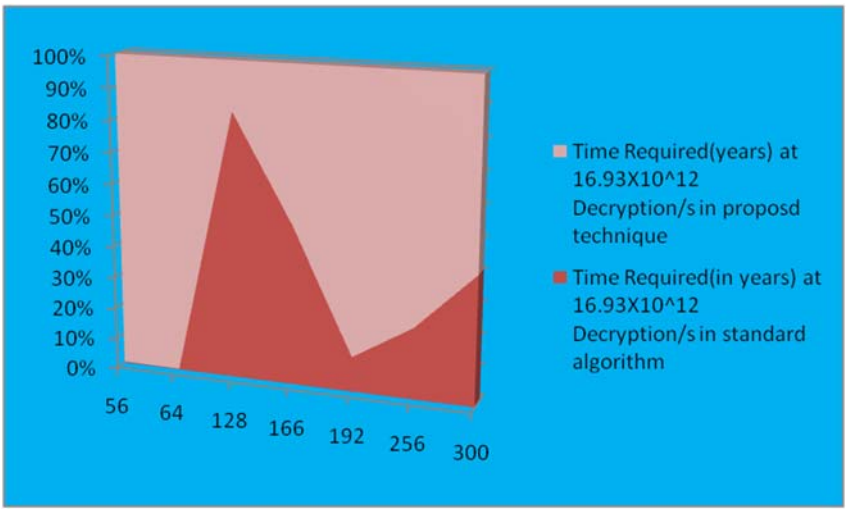

Fig3.3D graph of exhaustive key search

OBSERVATIONS: - From the above table it is seen that with respect to number of trials our proposed technique provides good result than any standard algorithms (like DES, Triple DES, AES etc) with same key size. The x-axis of the graph represents key size in bits. The above table and graph also shows that our proposed technique provides good result for decryption than any standard algorithms (like DES, Triple DES, AES etc) with fixed decryption rate. So, it difficult for attacker to decrypt any cipher text using assumed key. Thus overall result of our technique is good with respect to any standard algorithms in brute force attack.

6.2 *DICTIONARY ATTACKS:* **-**

Passwords found in any on-line or available list of words may be uncovered using dictionary attack by an attacker who tries all words in this list. The traditional dictionaries are not only used to find password but also on-line dictionaries of words from foreign languages, or on specialized topics such as music, film, sports etc. are used. For repeated use of these words in encryption process an adversary may create, an "encrypted" (hashed) list of dictionary or high-probability passwords. This dictionary may be used by attacker in guessing right encryption key for decryption. Dictionary attacks are more efficient than a brute force attack because it cannot try nearly as many combinations and if the key is not contained in the dictionary, it will never successfully find it.

In our proposed methodology, we have used random number generation functions, concept of matrix and fuzzy function and as a result the session key generated in this way not only contains English words or variations or phrases but also contains different ascii characters, numbers, and special characters. This would exhaust attacker's dictionary without a positive match.

6.3 RANDOMNESS TEST & ENTROPY TEST OF SESSION KEY: **-**

In our technique the session key is generated from symmetric key using fuzzy logic. Now to test the randomness of session key we use some standard techniques such as frequency test [14], entropy [15].

FREQUENCY TEST: -The frequency test is the most basic test for randomness checking. The purpose of this test is to determine whether the number of 1's and 0's in a sequence is approximately the same as would be expected for a truly random sequence.

Mathematical Structure of the Test:

Frequency (n), where n is the length of bit string.

$\mathcal{E}$: the sequence of bits which are generated by RNG or PRNG.

S_{obs}: the absolute value of the sum of the X_i (where $X_i = 2\mathcal{E} - 1$) is the sequence divided by the square root of the length of the sequence.

1) Conversion to ± 1: The zeros and ones of the input sequences ($\mathcal{E}$) are converted to values of -1 and +1and are added together to produce$S_n = X_1 + X_2 + \cdots + X_n$, where $X_i = 2\mathcal{E}_i - 1$.

2) Compute the test static $S_{obs} = ABS(S_n)/\sqrt{n}$.

3) Compute P-value= $erfc(S_{obs}/\sqrt{2})$.

4) If P-value≥ 0.01 then the conclusion is that the sequence is random and if P-value< 0.01 then the sequence is not random. The following table shows the details.

Table5.Table for Frequency test result

Symmetric key size (bits)	Frequency test result of our technique.	Frequency test result of PRNG()
56	4598.256	4272.772
64	4625.147	4310.446
128	4662.584	4336.617
166	4579.128	4270.683
192	4519.967	4270.656
256	4633.422	4323.351
300	4722.322	4423.356

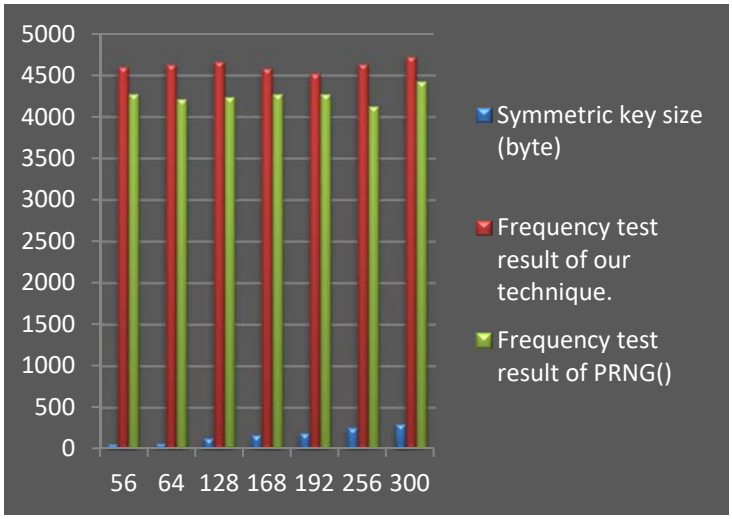

Fig4.Graph of frequency test of above table

OBSERVATIONS: - NIST SP 800-22 specifies that the randomness test must follow the three characteristics such as Uniformity, Scalability and Consistency. In case of uniformity and scalability, the occurrence of a zero or one is equally likely that is the probability of occurrence of zero or one is half. The above table of frequency test result shows uniformity and scalability of our technique. In case of consistency, we can say that the seed value from which we can generate the session key is symmetric key. For cryptographic applications, the symmetric key must be secure. The session key is generated by using a random key matrix and a symmetric key. Now if the key matrix is unknown or may change time to time and if the symmetric key is secured then the next output bit in the sequence should be unpredictable in spite of any knowledge of previous bits in the sequence. It should not be feasible to determine the symmetric key from the knowledge of any generated values. There is no correlation between symmetric key and generated values. Thus our technique proves the forward and backward unpredictability. Furthermore, from the above table and graph it is seen that our proposed technique provides more randomness than PRNG () which is standard technique.

6.4 EXPERIMENT ON ENTROPY VALUE -

Here we describe a comparative study between our technique and standard technique, PRNG () with session key and symmetric key.

Table6.Table for Entropy value

Symmetric key size (bits)	Entropy value of our technique.	Entropy value of PRNG()
56	6.88	7.03
64	6.88	7.02
128	6.88	7.02
166	6.88	7.03
192	6.89	7.01
256	6.89	7.03
300	6.90	7.04

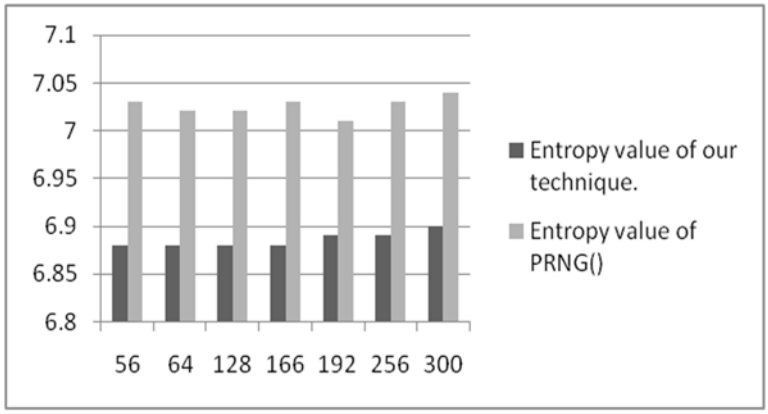

Fig5.2D Graph of Entropy value of above table

OBSERVATIONS: - In cryptography, a cryptosystem is said to be semantically secure if it is computationally impossible for an attacker to extract any information about the plain text from cipher text and its length. Entropy can be defined as randomness or unpredictability of information contained in a message. This randomness breaks the structure of plain text. Entropic security in encryption is similar to semantic security when data have highly entropic distribution. Plain text entropy value is zero. Now from the comparative study of entropy value between our technique and PRNG (), it is seen that the entropy value of our technique is near to the result of PRNG (). The x-axis shows the key length. Thus from the definition of entropic security we say that it is very hard to predict plain text from cipher text if we use our technique to generate session key and the use of this session key and symmetric key in encryption provides robustness.

6.5 COMPARATIVE STUDY ON AVALENCHE EFFECT:-

Here a comparative study between our technique and DES on avalanche effect [3], [16] with fixed key is described below with a table and graph.

Table7. Table for Avalanche effect with fixed key

Text Size (byte)	Total bit flipped in DES	Total bit flipped in our Technique
17	17	17
24	25	24
32	33	32
64	63	64
92	90	90
128	126	123
144	137	141

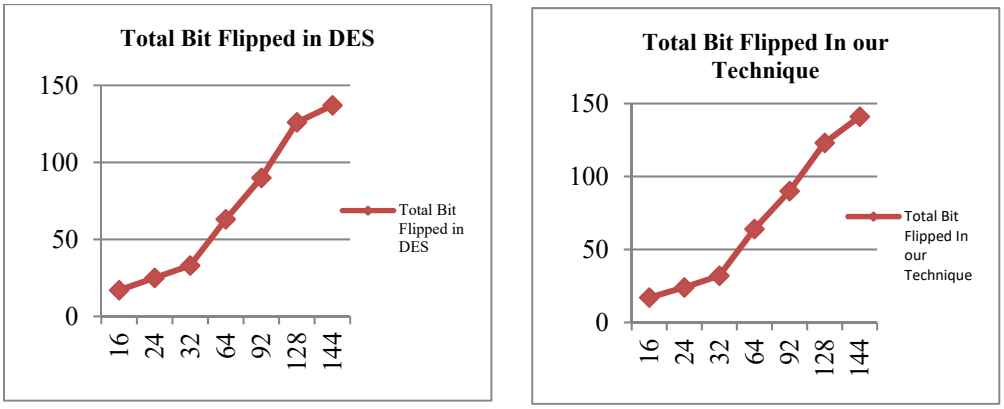

Fig6. Graph of Avalanche effect with fixed key for above table

OBSERVATIONS: - A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the cipher text. In particular, one bit change in the plaintext or one bit in the key should produce a change in many bits of the cipher text. Thus avalanche [3] quantifies the effect on the cipher text when one bit change in plaintext. An encryption algorithm that doesn't provide the avalanche effect can lead to an easy statistical analysis that is if the change of one bit from the input leads to the change of only one bit of the output, then it's easy to guess. Above table and graph provides comparative study between DES and our technique. In the graph x-axis represent text size. This study tells that total number of bit flipped in our encryption technique is more than DES. Here we use fixed size key. Thus our technique (using fixed key) provides good result than any standard algorithm (like DES). So, our proposed scheme satisfies the desirable property for encryption algorithm.

6.6 *ANALYSIS OF OUR ENCRYPTION TECHNIQUE: -*

Cryptanalysis is the study of cipher text, ciphers and cryptosystems. The aim of cryptanalysis is to understand how they work and finding and improving the techniques for defeating or weakening them. There are different types of cryptanalysis attacks such as Cipher text Only Attack, Chosen plain text Attack, Known Plaintext Attack, Chosen cipher text Attack, Breaking an encryption algorithm is basically the finding of the key to the access the encrypted data in plain text. For symmetric key encryption, breaking the algorithm usually means trying to determine the key used to encrypt the text. For a public key encryption, breaking the algorithm usually means acquiring the shared secret information between two recipients. The robustness of an encryption technique is depends on non linearity in cipher text. In our paper we use circular left shift operation and a non linear function to provide non linearity in cipher text. As a result our technique is able to protect any types of cryptanalysis attack. The following graph shows the robustness of our protection mechanism.

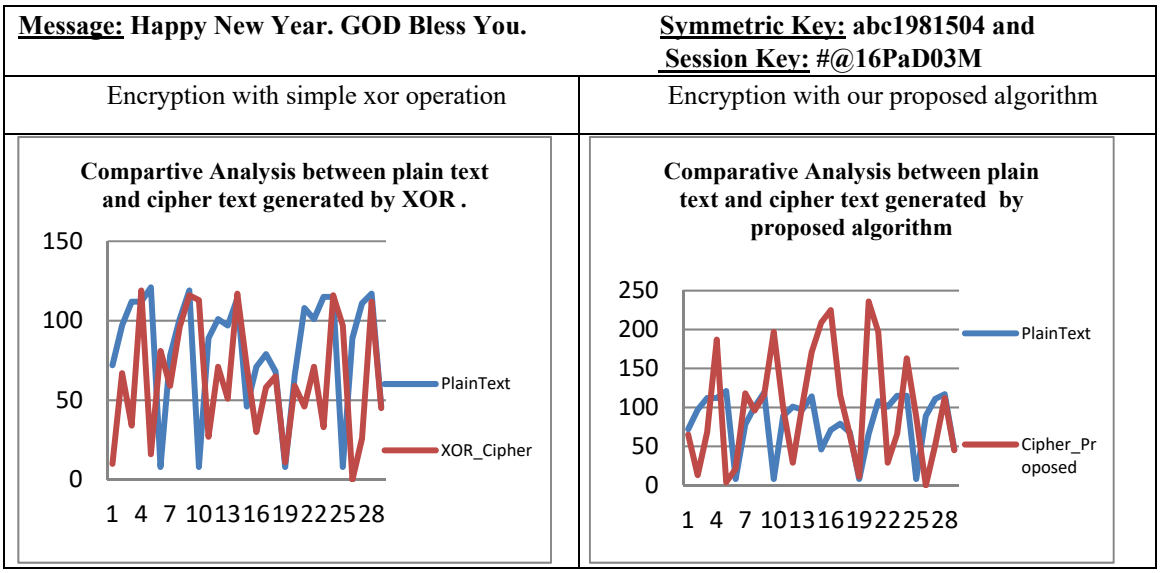

Fig7. Analysis of Encryption Technique

OBSERVATIONS**:** - Non linearity is a main theme in any encryption technique. From the above fig.6 it is seen that our technique provides more non linearity in cipher text than simple XOR operation. If we consider any point (x, y) and (a, b) in the plainText and Cipher_ Proposed respectively of above graph then any periodic gap is not exists between points in the graph i.e., there is no relationship between two graphs. So it is hard to guess plain text or encryption key from cipher text. Thus our encryption scheme is robust as well as it may protect any types of cryptanalysis like known plain text attack, chosen cipher text attack etc. From above Theorem1, it is also seen that this technique satisfies the condition of perfect security because cipher text and plain text are independent.

6.7 COMPARATIVE DISCUSSION: -

In this section the functionality of our scheme is done by comparing our proposed technology with different standard cryptographic algorithms [15, 17, 18, 19, 20, 21].

The following tables show comparison among different standard algorithms and our proposed algorithm.

Table8. Comparisons among proposed technique & standard algorithms

Algorithms	**Important Features**	**Important features of our proposed algorithm**
IDEA	i) IDEA encrypts 64-bit plaintext to 64-bit cipher text blocks, using a 128-bit input key. ii) It uses both confusion and diffusion technique. iii) A dominant design concept in IDEA is mixing operations from three different algebraic groups of 2^n elements. iv) The security of IDEA currently seams that it is bounded only by the weaknesses arising from the relatively small (compared to its key length) block length of 64 bits.	i) Our technique encrypts n-bit plaintext to n-bit cipher text, using m-bit input key. ii) The main design concept of our technique (a) symmetric key generation using the concept of linear congruence. (b) Session key generation using approximation algorithm. (c)Circular left shift is used to produce non linearity in encryption process.
RC5	i) The RC5 block cipher has a word-oriented architecture for variable word sizes w = 16, 32, or 64 bits. ii) The number of rounds r and the key byte-length b are variable. iii) For encryption, there are two steps in each round, (a) bit-wise XOR operation, (b) circular left shift. (c) Addition with the next sub key.	i) Our proposed scheme is stream cipher based. Here two keys are used for encryption/decryption. ii) Key length is variable. iii) For encryption, there are two steps-(a) bit-wise XOR operation (b) Circular left shift with a linear function. It provides number of times CLS occurs.
BLOWFISH	i) This technique is based on stream cipher. It uses addition, XOR operation for encryption. ii)It has a variable key length up to a maximum of 448 bits long which ensures security. iii) Blowfish suits applications where the key remains constant for a long time and it is not suitable for packet switching.	i) Our scheme is also based on steam cipher. It uses XOR, CLS operations to impose more non linearity in cipher text. ii) The use of double keys and one of this key is changeable by nature which provides robustness in our technique. iii) Suitable for packet switching.
DES	i) Linear cryptanalysis provides the most powerful attack on DES to date where enormous number of known plain text pairs is feasible. ii) Differential cryptanalysis is one of the most general cryptanalytic tools to date against modern iterated block ciphers, including DES. It is primarily a chosen-plaintext attack. iii) Storage complexity, both linear and differential cryptanalysis requires only negligible storage. iv) Due to its short key size, the DES algorithm is now considered insecure and should not be used. However, a strengthened version of DES called Triple-DES is used.	i) Our algorithm is based on stream cipher with two keys one is session key which is changeable in nature. So it protects linear cryptanalysis as well as differential cryptanalysis. ii) Our algorithm takes negligible storage for linear and differential cryptanalysis. iii) Our algorithm is secure with respect to key size, because we have used two keys with variable length.

Triple-DES with Three keys	i) Triple-DES counters to the meet-in-the-middle attack by using three stages of encryption with three keys. ii) Tuchman proposed a triple encryption method that uses only two keys. The function follows an Encrypt-Decrypt-Encrypt sequence. $C=E(K_1,D(K_2,E(K_1,P)))$ and $P=D(K_1,E(K_2,D(K_1,C)))$ iii) There is no cryptographic significance in 2nd stage decryption. It only advantage is that it allows users to decrypt information encrypted by users of single DES. iv) There is no practical cryptanalytic attack on 3DES. This method is an improvement over the chosen plain text approach but requires more effort. This attack is based on observation that if the value of 1st phase encryption and final cipher text is known then the problem reduces to double DES.	i) This article proposed a method that counters meet-in–the–middle attack by using two different keys with variable times of CLS. ii) This technique has cryptographic importance in wireless network; it uses two keys such as symmetric key and session key which is generated using fuzzy logic. It also provides strong authentication mechanisms. iii) It is very hard to anticipate the two keys if plaintext-cipher text pair is known.

7 Conclusions

We have presented a key generation technique and an encryption technique based on symmetric key and session key for secure data transmission in wireless network. This session key is generated from symmetric key using some tools such as recurrence relation, fuzzy logic, and random matrix. Here receiver decrypts the cipher text using his or her symmetric key and session key. Our technique also provides the proof of identities which enrich the robustness as well as beauty of our proposed scheme. Thus our technique provides two contributions on symmetric key encryption. Firstly, a new approach for extracting session key from secured symmetric key and a random matrix. This random matrix provides randomness of our session key. Secondly, the random matrix is generated from random numbers and this random numbers are generated from recurrence relation from discrete mathematics. The use of recurrence relation is a new approach in random number generation. It provides variable number of random numbers depending on initial value and context values and total numbers of random numbers depend on user. Thirdly, the encryption technique with two keys i.e., two times encryption with session key and symmetric key and circular left shift operation on partial encrypted form provides robustness of our technique. At last we have included an authentication mechanism in our technique. Comparative statistical tests like entropy value and frequency test etc between proposed technique and standard technique proves the sturdier of our key. Lastly, exhaustive key search analysis shows the acceptability of our technique. To the best of our knowledge our proposed technique is the simplest encryption technique with symmetric key, session key with authentication mechanisms. It is practically having minimal computational overhead during encryption and decryption.

References

[1] A. Das and C. E. Veni Madhavan, Public-key Cryptography: Theory and Practice, Pearson Education,in press.
[2] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in Kilian, J. (ed.) CRYPTO2001. LNCS, vol. 2139, (Springer, Heidelberg, 2001), pp. 213–229.
[3] W. Stallings, Cryptography and Network Security: Principles and Practice, third edition, Prentice Hall, 2003.
[4] Fuzzy Logic: An Introduction [online], http://www.seattlerobotics.org.
[5] Europe Gets into Fuzzy Logic", Electronics EngineeringTimes, 1991.
[6] "Fuzzy Sets and Applications: Selected Papers by L.A.Zadeh", ed. R.R. Yager et al. (John Wiley, New York, 1987).
[7] "U.S. Loses Focus on Fuzzy Logic" (Machine Design, June 21, 1990).
[8] Atul Kahate, "cryptography and network security", Tata McGraw-Hill publishing company, New Delhi, 2008.
[9] E.T. Oladipupo, O.A. Alade, "An Approach to Improve Data Security using Modified XOR Encryption Algorithm", 2014, International Journal of Core Research in Communication Engineering, Volume No. 1, Issue No. 2.
[10] D. Stinson, Cryptography: Theory and Practice, third edition, Chapman & Hall/CRC, 2006.
[11] A. Agrawal, S. Gorbunov, V. Vaikuntanathan, H. Wee, Functional encryption: New perspectives and lower bounds, in R. Canetti, J.A. Garay, (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043. (Springer, Heidelberg, 2013), pp. 500–518.
[12] J.Buchmann, Introduction to Cryptography, second edition, Springer, 2004.
[13] S. A. Chaudhry et al. An improved and provably secure privacy preserving authentication protocol for SIP. Peer-to-Peer Networking and Applications. 2017; 10(1): 1-15.
[14] A. Kak, "Lecture Notes on Computer and Network Security", 2015, Purdue University [Online] Available: https://engineering.purdue.edu/kak/compsec/Lectures.html.
[15] Zaidan B, Zaidan A, Al-Frajat A, Jalab H. On the differences between hiding information and cryptography techniques: An overview Journal of Applied Sciences. 2010; 10:1650–5.
[16] H. Delfs and H. Knebl, Introduction to Cryptography: Principles and Applications, Springer, 2002.
[17] C. Joshi, and U.K. Singh, "A Review on Taxonomies of Attacks and Vulnerability in Computer and Network System". International Journal of Advanced Research in Computer Science and Software Engineering (IJRCSSE) Volume 5, Issue 1, January 2015, pp 742-747.
[18] Siddharth Ghansela "Network Security: Attacks, Tools and Techniques", ijarcsse Volume 3, Issue 6, June 2013.

[19] Mohan V. Pawer, Anuradha J, "Network Security and Types of Attacks in Network", In proceedings International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015), Procedia Computer Science48(2015)503–506. DOI: 10.1016/j.procs.2015.04.126.
[20] Monali S. Gaigole et al, International Journal of Computer Science and Mobile Computing (ISSN 2320–088X) Vol.4 Issue.5 May-2015 pg. 728-735.
[21] J.G. Chakravorty, P.R. Ghosh, Advanced Higher Algebra, U.N. Dhur and Sons Private Ltd., 2018. ISBN 978-93-80673-67-7.
[22] Hong Yaling. Research on computer network security analysis model [J]. Computer CD Software and Applications, 2013(z):1-152.