A DEFENSIVE APPROACH AGAINST MATHEMATICAL CRYPTANALYSIS USING SYMMETRIC KEY AND FUZZY BASED SESSION KEY

[1] Anirban Bhowmik, [2]Dr.Sunil Karforma, [3]Joydeep Dey, [4]Dr.Arindam Sarkar

[1]*Assistant Professor*
Department of Computer Application, C.R.T.I.
Tinkonia, Goods shed Road, Burdwan, India
Email: animca2008@gmail.com
[2]*Head of the Department*
Department of Computer Science, The University of Burdwan,
Burdwan, India
Email: dr.sunilkarforma@gmail.com
[3]*Guest Lecturer*
Department of Computer Science, M.U.C. Women's College,
B.C. Road, Burdwan, India
Email: joydeepmcabu@gmail.com
[4]*Assistant Professor*
Department of Computer Science & Electronics, R.K.M. Vidyamandira,
Belur Math,Belur,India
Email: arindam.vb@gmail.com

Abstract

At present market, with the help of science, the information technology has developed rapidly as well as the information security as an important part of information technology is also strongly affecting people's work and life. When the development of science and technology brings convenience to characters, it also exposes many security issues. In recent years, information security incidents such as information leakage, SQL injection vulnerabilities, network penetration, and hacking attacks have triggered irreparable losses to enterprises, society, and individuals. In our paper, a symmetric key encryption with fuzzy based session key has been proposed for satisfying the key issues like security; increase the strength of symmetric key. Here the session key is generated using fuzzy logic. Now the encryption is done by using this session key and symmetric key. Here a new authentication scheme is developed to generate cipher text. Different types of experiments to test the robustness of keys and encryption technique and comparative study with existing standard techniques have been done with satisfactory results. This paper has taken steps to develop an attack capability for use within a cyber challenge environment. The principles discussed within this paper aim to be applicable to all challenges against mathematical cryptanalysis and cryptographic protocol in general and also it is used to cover threat modeling, construction of cyber security test beds and take footsteps against offensive cyber operations.

Keywords: *Symmetric key, Session key, Fuzzy logic, Encryption, Decryption, Cryptanalysis.*

INTRODUCTION

Offensive Cryptography is defined as means and actions taken to defeat a cryptographic defense, and otherwise harm the data interests of our adversary based on knowledge discrimination. In most cases offensive cryptography (also known as 'cryptanalysis') amounts to the challenge of how to reveal a cryptographically hidden secret. Offensive security, the practice of exploitation has greatly enhanced our understanding of what it means for computers to be trustworthy. Offensive computing is a hacker curriculum. Exploitation is a programming. It is the kind of programming that every programmer should, at least understand in terms of its capabilities and limits, because it will be practiced on his code. Our security is only as good as our understanding of this kind of programming, because it's the essential nature of general-purpose systems to allow huge number of other execution paths than merely the intended ones. The meaning of Security and trustworthiness of code is that attackers' inability to program it [17]. In our paper we focus on programming on symmetric key encryption technique with two keys. Symmetric encryption or single-key encryption, was the only type of encryption in use prior to the development of public key encryption in the 1970s.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.

ENCRYPTION & DECRYPTION: - In technical terms, the process of encoding plain text message into cipher text message is called as encryption. The decryption is exactly the opposite of encryption i.e. decryption transforms a cipher text message back into plain text [8].

ENCRYPTION DOMAINS & CODOMAINS [22]

•A denotes a finite set called the *alphabet of definition*. For example, A = {0, 1}, the

binary alphabet, is a frequently used alphabet of definition.

• M denotes a set called the *message space*. M consists of strings of symbols from

an alphabet of definition.

• C denotes a set called the *cipher text space*. C consists of strings of symbols from an alphabet of definition, which may differ from the alphabet of definition for M.

ENCRYPTION & DECRYPTION TRANSFORMATION [22]

• K denotes a set called the *key space*. An element of K is called a *key*.

• Each element $e \in K$ uniquely determines a bijection from M to C, denoted by Ee.

Ee is called an *encryption function* or an *encryption transformation*. Note that Ee must be a bijection if the process is to be reversed and a unique plaintext message recovered for each distinct cipher text.

• For each $d \in K, D_d : (d, C) \rightarrow M$. D_d is called a *decryption function* or *decryption transformation*.

• The process of applying the transformation Ee to a message m ∈ M is usually referred to as *encrypting* m or the *encryption* of m.

• The process of applying the transformation Dd to a cipher text c is usually referred to as *decrypting* c or the *decryption* of c.

The following theorem describes on perfect security in cipher text.

Theorem1. Let $\mathcal{E}$ = (E, D) be a Shannon cipher defined over (K, M, C). Consider a random experiment in which k and m are random variables, such that

• k is uniformly distributed over K,

• m is distributed over M, and

• k and m are independent.

Define the random variable c: = E (k, m). Then we have:

• If $\mathcal{E}$ is perfectly secure, then c and m are independent;

• Conversely, if c and m are independent, and each message in M occurs with nonzero probability, then E is perfectly secure [22].

Proof. We define M* to be the set of messages that occur with nonzero probability. We begin with a simple observation. Consider any fixed m € M* and c€ C. Then we have

$Pr[c = c \mid m = m] = Pr[E(k,m) = c \mid m = m]$, and since k and m are independent, so are E (k, m) and m, and hence $Pr[E(k,m) = c \mid m = m] = Pr[E(k,m) = c]$.

Putting this all together, we have: $Pr[c = c \mid m = m] = Pr[E(k,m) = c]$. (1.1)

We now prove the first implication. So assume that $\mathcal{E}$ is perfectly secure. We want to show that c and m are independent. To do this, let m € M* and c € C be given. It will suffice to show that $Pr[c = c \mid m = m] = Pr[c = c]$.

We have

$Pr[c = c] = \sum_{m' \in M^*} \Pr[c = c \mid m = m_0] \Pr[m = m_0]$ (by total probability)

$= \sum_{m' \in M^*} \Pr[E(k, m_0) = c] \Pr[m = m_0]$ (by (1.1))

$= \sum_{m' \in M^*} \Pr[E(k, m) = c] \Pr[m = m_0]$ by definition of perfect security

$= Pr[E(k,m) = c] \sum_{m' \in M^*} Pr[m = m_0]$

$= Pr[E(k,m) = c]$ (probabilities sum to 1)

$= Pr[c = c \mid m = m]$ (again by (1.1))

This shows that c and m are independent. That proves the first implication.

For the second, we assume that c and m are independent, and moreover, that every message occurs with nonzero probability (so M* = M). We want to show that $\mathcal{E}$ is perfectly secure, which means that for each m_0, m_1 € M, and each c € C,

We have

$$Pr[E(k,m_0) = c] = Pr[E(k,m_1) = c]. \quad (1.2)$$

But we have

$Pr[E(k,m_0) = c] = Pr[c = c \mid m = m_0]$ (by (1.1))

$= Pr[c = c]$ (by independence of c and m)

$= Pr[c = c \mid m = m_1]$ (again by independence of c and m)

$= Pr[E(k,m_1) = c]$ (again by (1.1)).

That shows that $\mathcal{E}$ is perfectly secure.

SESSION KEY: - The session key is one type of key which is changed time to time. In our scheme, this session key is used in stream cipher method based encryption. Stream cipher is semantically secure but there are some limitations on it [8].

A stream cipher is well equipped to encrypt a single message from A to B. But A may wish to send several messages to B. For simplicity suppose A wishes to encrypt two messages m_1 and m_2. The simple solution is to encrypt both messages using the same stream cipher

Key s: $c_1 \leftarrow m_1 \, XOR \, G(s) \; and \; c_2 \leftarrow m_2 \, XOR \, G(s)$

The above construction shows that it is insecure in a very strong sense. An adversary who intercepts c_1 and c_2 can compute

$$\Delta := c_1 \, XOR \, c_2 = (m_1 \, XOR \, G(s)) \, XOR (m_2 \, XOR \, G(s)) = (m_1 \, XOR \, m_2)$$

and obtain the xor of m_1 and m_2. English text contains enough redundancy that given Δ = m1 *xor* m2 the adversary can recover both m_1 and m_2 in the clear. Hence, the construction of c_1 and c_2 leaks the plaintexts after seeing only two sufficiently long cipher texts. Thus in particular, a stream cipher key should never be used to encrypt more than one message [22].

FUZZY LOGIC: - Fuzzy logic [4], [5], [6], [7] deals with fuzzy predicates and fuzzy implications made up of fuzzy predicates. It deals with how to make inferences using fuzzy predicates and fuzzy implications. A fuzzy predicate is described in terms of fuzzy sets and fuzzy implications 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 A: U→ [0, 1].'A' is called the membership function, A(x) is called the membership grade of x. we can write A= {x,A(x)):x€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∨q)= max[T(p), T(q)], T(p∧q)= min [T(p), T(q)], T (¬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 that region of

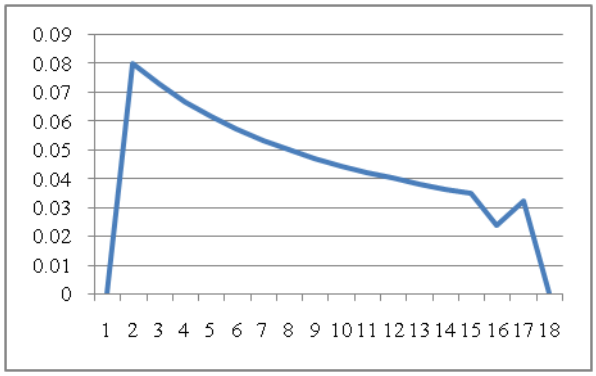
the universe that is characterized by complete and full membership in the set$A_{\sim}$. That is, the core comprises those elements *x* of the universe such that $\mu_{A\sim}(x) = 1$.

The *support* of a membership function for some fuzzy set $A_{\sim}$ is defined as that region

of the universe that is characterized by nonzero membership in the set $A_{\sim}$. That is, 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 that 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 the function is 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.



This figure represents our fuzzy membership function

RELATED WORK

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. Mingxuan Li, Zhushi Yang,Ling He, YangXin Teng has publisheded a paper on Research and Application of Information Security Offense and Defense Exercise in Electric Power Industry which indicates that why it is necessary to improve the information security awareness of personnel who are engaged in information security work in power grid companies and to respond to emergency information security incidents, and indirectly ensure safe, stable, and reliable operation of the power grid [19].

Mitsuru Matsui invented the linear cryptanalysis attack [10] based on linear approximation. Ahmed Hweishel A. Alfarjat, Hanumanthappa J. has introduced a concept over the Security issues using Elliptic Curve Cryptography. Their research work also explores the Bucket Brigade Attack on Bluetooth security using Elliptic Curve Cryptography (ECC). Also it is implied that Bucket Brigade Attack (BBA) is one of the amazing solution to the problem of key agreement or key swapping [18]. Ivan Burke and R.P. van Heerden has introduced a concept on how to develop an automated attack capability for use within a cyber challenge environment. This paper is a practical application of an ontological model for cyber attack scenarios [17]. Michael Kranch has introduced an approach to demonstrate why offensive (hacking) techniques are the best method for teaching cyber security's core competencies [18]. Timing attack is also applied on symmetric key encryption. There also exists Sensor Network Encryption Protocol (SNEP) [2] for security of sensor network systems. 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 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, pdf etc) and by the reverse way we can decrypt the cipher text to get plain text in optimum way.

PROBLEM DOMAIN

Offensive cryptography is a vital issue in cyber space and also in network security. In defensive cryptography there exist different types of cryptographic algorithms. In symmetric key encryption technique we can transmit huge amount of data between sender and receiver effectively. 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. The intruders may occur offensive case by revealing cryptographically secret things using the secret key. Existing standard 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.

SOLUTION DOMAIN

Here we have introduced a defensive cryptographic technique for secure communication in general. Our proposed technique is based on stream cipher method with two keys [8].

From the above context (session key of introduction part), it is seen that stream cipher based encryption is week for a single key. But for providing robustness in encryption technique, we have introduced another key called a session key which can be changed time to time. The randomness property is injected in session key. So we have used two keys one is symmetric key and another is session key in encryption.

This session key is generated by using a random character matrix, fuzzy logic and the symmetric key. Since this session key may change time to time so the use of both session key and symmetric key in encryption provides the extra robustness against cryptanalysis.

Here we deduce a novel technique by using both the symmetric key and session key for authentication purpose. Thus, the use of session key with symmetric key and an authentication cum encryption provides the added flavor in our proposed technique. In this paper, the following section 5 describes *methodology*, section 6 describes *result section* and section 7 describes *conclusion.*

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. 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 Matrix_GA () // matrix createS ‘n’ number of key population from symmetric key.

2. Call Session_KG ( ) // generate session key using ‘n’ number of key populations and fuzzy logic.

3. Call EncProc ( ) // Encryption Process i.e., cipher file is generated using two keys.

4. Call Create_Header_Tail ( ) // header and tailer structure is created using two keys with XOR operation.

5. Call Concat(header, encrypted file,tailer) // total structure is created and it is ready for transmission over network.

6. Call Authen_Check ( )// check authentication using two keys and generate session key using symmetric key.

7. Call DecrypPhase ( ) // plain text is generated.

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

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. The predefined matrix is a square matrix and the number of column 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: Matrix Generation Algorithm (Matrix_GA)

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-2: Session Key Generation (Session_KG)

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 charac
ter 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

ENCRYPTION PHASE WITH SYMMETRIC KEY AND SESSION KEY:

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 with (session key /3) times circular left shift operations. The encryption algorithm is given below.

ALGORITHM-3**:** Encryption Process (EncProc)

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,file_tmp as temporary file.

3. if (!eof) then

4. file_Output= bit_XOROP (file_Plain , session key)

5. file_tmp= file_output <<(session key length)/3.

5. file_Cipher= bit_XOROP (file_Output , symmetric key)
6. end if
7. end for
8. End

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

HEADER AND TAILER CREATION: -

ALGORITHM-4**:** Create_Header_Tail ( )

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(1^{st} 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)

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_Tail ()* function so that we can reveal the session key using the symmetric key from Header section and we can check the authentication from Tail part using the function *Authen_Check()*. 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. The whole process is done under the function *DecrypPhase( )*.

SIGNIFICANCE OF AUTHENTICATION: - Authentication mechanisms [8] provide the proof of identities. The authentication process ensures that the origin or source of document is correctly identified i.e. the document is coming from right person. In our scheme we have used authentication 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 Header and Tailer. There are complex calculations for Header and Tailer generation. Tailer structure is used for authentication purppose. In receiver side symmetric key and particular row are used to generate session key. Now using this session key and symmetric key we check authentication from Tailer part. Thus our technique protects the fabrication. This new authentication technique also boosts up our proposed method against cryptanalysis [8][22].

RESULT & DISCUSSION

The encryption-decryption process is implemented in TURBO C interface in a PC of Intel Core i3 processor and 2GB RAM. In this section, simulation results of the proposed scheme are presented. In our experiments, several sizes of files are used as plain text.

DIFFERENT TYPES OF ATTACKS: -

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) Cryptanalysis (ii) Brute-force attack [11], [12].

KEY SPACE ANALYSIS

The size of key space is the total number of different keys in encryption process. The brute-force attack is impractical in such crypto systems where key space is large. Now we consider a general case where secret key is k bits. There are two keys are in our proposed scheme, first is symmetric key with size k bits and second is session key whose size is k/2 bits. Now for the symmetric key, the key space is 2^k and for session key, the key space is $2^{k/2}$ and total key space is $2^{3k/2}$. Using this large key space we discuss following things.

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., $33.86x10^{15}$ 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: $16.93x10^{12}$. 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 $2^{3k/2}$/ ($16.93*10^{12}$ *3153600) =Y. So if k increase then Y increases. The following table and graphs shows the average time required for exhaustive key search [3].

Table1. Table for exhaustive key search

Symmetric key size (k bits)	No. of Trials in standard algorithms(2^k)	Time Required(in years) at $16.93X10^{12}$ Decryption/s in standard algorithms	No. of Trails in our proposed technique ($2^{(3k/2)}$)	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}	$1.1775x10^{38}$

168	2^{168}	$7.0077x10^{30}$	2^{252}	$1.3555x10^{56}$
192	2^{192}	$1.1756x10^{38}$	2^{288}	$9.3148x10^{66}$
256	2^{256}	$2.1687x10^{57}$	2^{384}	$7.3799x10^{95}$

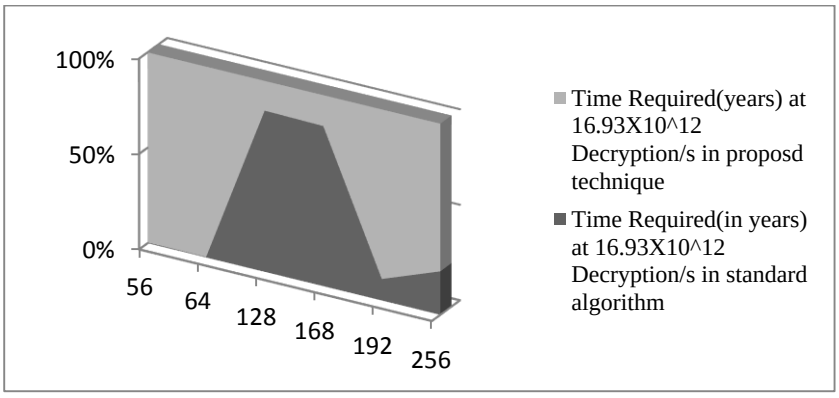


Fig1.3D graph of exhaustive key search

OBSERVATIONS: - From the above table1 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 table1 and fig.1 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.

RANDOMNESS TEST OF SESSION KEY: - Randomness means all elements of the sequence are generated independently of each other, and the value of the next element in the sequence cannot be predicted previously, regardless of how many elements have already been produced. Random and pseudorandom numbers generated for cryptographic applications should be unpredictable (forward and backward).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 ON SESSION KEY: -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.

BlockFrequency (*M, n*), where: *M* The length of each block. *n* The length of the bit string.

χ^2(Obs): A measure of how well the observed proportion of one's within a given M bit block match the expected proportion (1/2) [22].

The following table shows the details.

Table2. Table for Frequency test result

Symmetric key size (bits)	Frequency test result of our technique.	Frequency test result of PRNG()
56	4597.255	4273.771
64	4615.148	4220.445
128	4662.584	4235.618
168	4579.128	4270.675
192	4529.966	4271.756
256	4634.421	4133.350

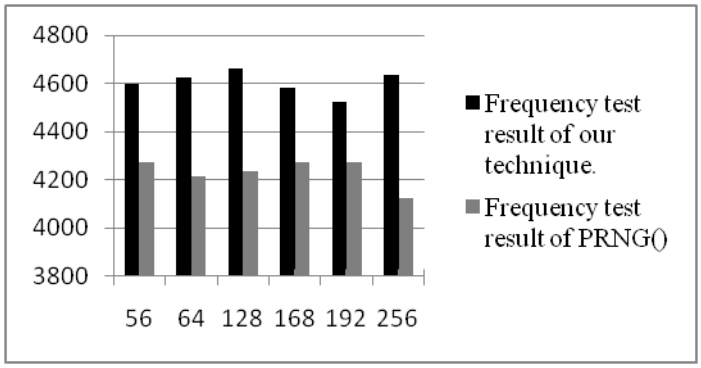


Fig2.Graph of frequency test of above table

OBSERVATIONS: - Thus the sequence generated by the above algorithm passes serial test. 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 table2 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.

CORRELATION ANALYSIS: - The following table3 shows the correlation analysis on results of frequency test. The formula for correlation coefficient is given below

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{[n\sum x^2 - (\sum x)^2]\,[n\sum y^2 - (\sum y)^2]}}$$

Table3. Analysis of correlation coefficient of frequency test result

Mean of X values	Mean of Y values	SD of X values	SD of Y values	Coefficient of correlation	Coefficient of determination	Significance test value	standard error slope value
4603.084	4234.269	46.09	54.16	-0.568	32.284%	-1.381	0.484

(X: Result of our technique and Y: Result of PRNG ())

From the table3 it is seen that there is no correlation between values of proposed technique and PRNG (). 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 a standard technique and as a result it is secure against different statistical attacks and differential attacks.

EXPERIMENT ON ENTROPY VALUE -:

The entropy is a measure of the disorder or randomness in a closed system. The entropy of uncertainty of a random variable X with probabilities $p_i, \ldots, p_n$ is defined as

$H(X) = -\sum_{i=1}^{n} p_i \log p_i$.

Entropy source is a physical source of information whose output either appears to be random in it or by applying some filtering/distillation process. This output is used as input to either a RNG or PRNG [22].

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

Table4. Table for Entropy value test

Symmetric key size (bits)	Entropy value of our technique.	Entropy value of PRNG()
56	6.88	7.03
64	6.87	7.02
128	6.88	7.03
168	6.88	7.03
192	6.89	7.02
256	6.89	7.03

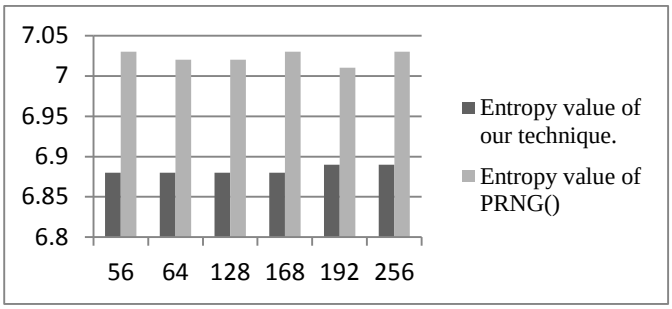


Fig3.2D Graph of Entropy value of above table

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.

Thus from above two tests it is clear that our session key is strong with respect to randomness and provides perfect security as it is uniformly distributed in key space.

KEY SENSITIVE ANALYSIS: - An ideal encryption technique should be sensitive with respect to the secret key i.e. the change of a single bit in the secret key should produce a completely different cipher text. For testing the key sensitivity of the proposed encryption procedure, we have performed the encryption process in the files (.txt) with slight changes in the secret key. The avalanche effect is shown only for changed session key and with fixed symmetric key. The following table5 and fig.4 shows the total scenario.

Table5.Comparative Analysis on Avalanche Effect

Key	Ascii Difference	Total no. of added characters	Total no. of deleted characters	Total no. of changed characters
Crypto@13011	0	3527	3546	2541
crypto@13011	8	3715	3711	2361
Crypt**q**@13011	9	3861	3837	2272
Crypto@1301**6**	108	3929	3875	2218
Crypto**#**13011	28	3616	3609	2459
Cry**s**to@13011	81	4263	4244	1893
Crypto@1301**z**	8	3796	3779	2315
Crypto@13**4**11	119	3869	3846	2244
Crypto@**0**3011	5	4104	4083	2048
Arypto@13011	66	4092	4090	2032

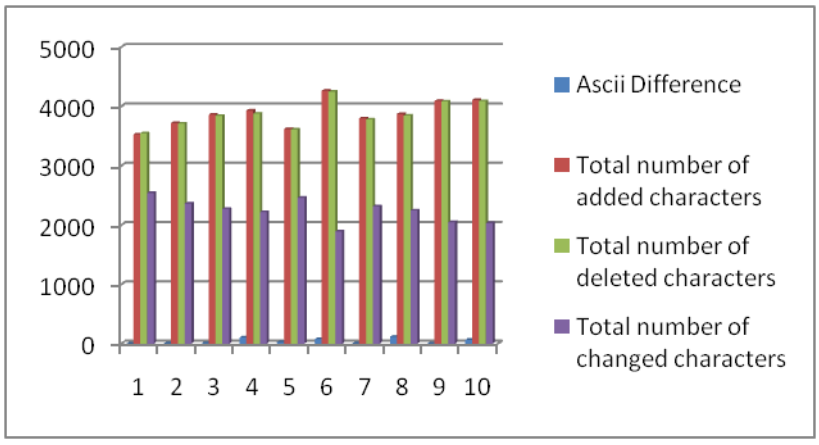


Fig.4: Graph of avalanche effect

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 table5 and fig.4 shows comparative analysis 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) as well as satisfies the desirable property for encryption algorithm.

CORRELATION ANALYSIS: - Here we calculate the correlation between ascii difference and total number of changed characters. A secure encryption scheme should transform a text file (.docx, .txt) into a random like encrypted file with low correlation. The formula for Pearson correlation coefficient is given below.

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{[n\sum x^2 - (\sum x)^2]\,[n\sum y^2 - (\sum y)^2]}}$$

The correlation values are given below through table 6.

Table 6: Data Analysis using Pearson correlation coefficient.

SD of X Values	SD of Y values	Correlation Coefficient (r)	Coefficient of Determination	Significance of Test Value	Standard error slope values
45.89	200.05	-0.426	18.15%	-1.33	1.394

(X values: No. of changed characters, Y values: ascii difference)

The Pearson correlation coefficient provides the strength and direction of the linear relationship between two variables. From the above table6 it is seen that the value of correlation coefficient between X and Y is -0.426<0. This indicates that there is a strong negative relationship between the variables or the variables may have a nonlinear relationship. The relationship is negative because, as one variable increases, the other decreases. But from the scatter plot it is seen that there is nonlinear relationship exists between two variables (X, Y).

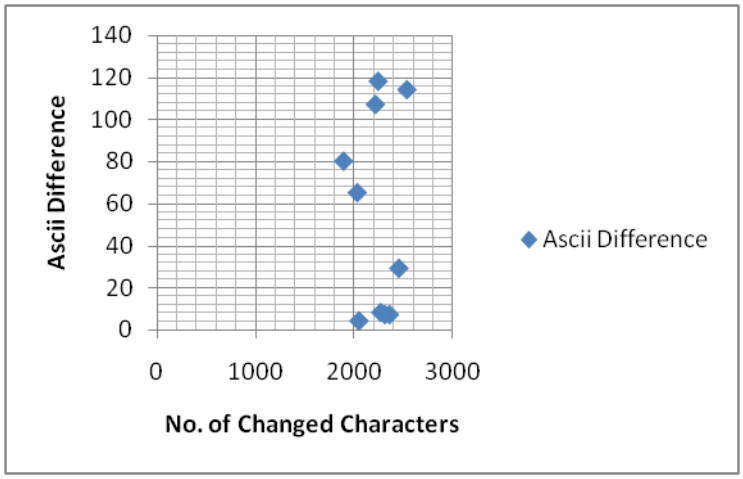


Fig5. Scatter diagram on correlation analysis

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 symmetric key or session key generated in this way not only contains English words or variations or phrases but also contains different ascii characters, numbers, special characters. This would exhaust attacker's dictionary without a positive match.

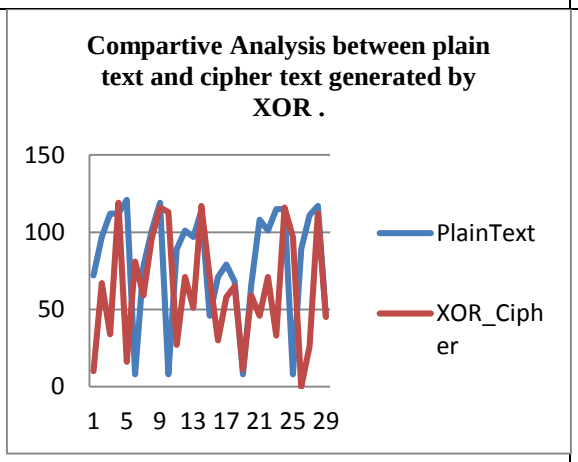
ANALYSIS OF OUR ENCRYPTION TECHNIQUE: - Cryptanalysis is the study of cipher text, ciphers and cryptosystems. The aim of cryptanalysis is to understand how they work and finding and improving the techniques for defeating or weakening them. There are different types of cryptanalysis attacks such as Cipher 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.

Message: **Happy New Year. GOD Bless You. and** **Symmetric Key:** **abc1981504**

Session Key: **#@16PaD03M**

Encryption with simple xor operation



Encryption with our proposed algorithm

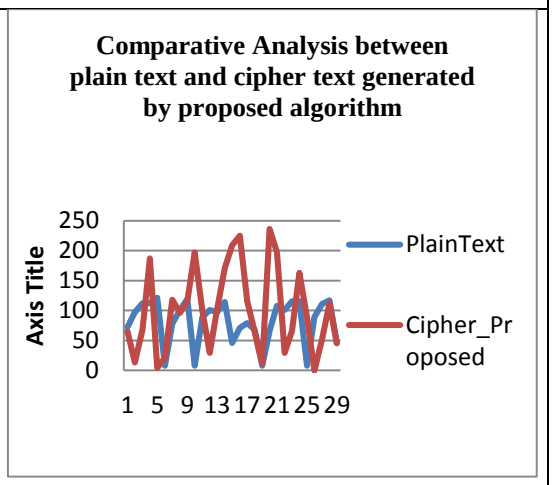


Fig6. Analysis of encryption technique

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.

FUNCTIONALITY ANALYSIS OF OUR PROPOSED TECHNIQUE: - In this section the functionality of our scheme is done by comparing our proposed technology with different

standard cryptographic algorithms, different existing schemes [17, 18, 19].The following table7 shows comparison among different standard algorithms.

Table7: Comparative Analysis among standard algorithms vs proposed algorithm

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) generation of key population using the concept of random matrix. (b) Session key generation using fuzzy logic. (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.	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

	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.	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 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.	i)Our encryption 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) The algorithm takes negligible storage for linear and differential cryptanalysis. iii) Our proposed algorithm is secure with

	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.	respect to key size, because we have used two keys with variable length.

The above comparative study shows strength, and acceptance of our scheme. The proposed protocol provides more functionality such as strong user authentication, mutual authentication between the two ends; it establishes a secure session key for the user. It is worth notice that our proposed protocol provides indispensable security features.

Our proposed algorithm works on application layer. The different types of complex and strong mathematical concepts like concept of random matrix, fuzzy logic, circular left shift are used and as a result when the algorithms run, it provides minimum side channel information so that attacker cannot guess the encryption key. Our scheme also provides confidentiality and integrity by checking authentication in both sender and recipient side.

The following table provides the pertinence of our methodology compared to other existing techniques.

Table8. Comparison among different techniques and proposed technique

Schemes→ Security Properties↓	Technique 1 (Ref.17)	Technique2 (Ref.18)	Technique 3 (Ref.19)	Proposed technique
Confidentiality	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes
Authenticity (message authentication and user authentication)	No	No	No	Message authentication using symmetric key & user

				authentication by RSA.
Freshness Protection	No	No	No	Yes
Privacy Protection	Yes	Yes	Yes	Yes
Man-in –middle attack	No	No	No	Yes
Tenability	No	No	Yes	Yes
Vulnerability	Yes	Yes	Yes	Yes
Impersonation attack	No	No	No	Yes
Cryptanalysis(linear and differential)	Yes	Yes	Yes	Yes
Session key establishment	No	No	No	Yes
Secure against Information-leakage attack	Yes	Yes	Yes	Yes

CONCLUSION

At present, it is seen that different types of attacks or offensive cryptography provides a threats on network communication such as chosen cipher text attacks, chosen plain text attacks, brute force attacks, dictionary attacks etc. With the improvement of technology, different attacks appear with improved version. In this paper we have presented a defensive cryptographic technique; an encryption technique based on symmetric key and session key to protect cryptanalysis. This session key is generated from symmetric key using some tools such as fuzzy logic, random matrix. Here receiver decrypts the cipher text using his or her symmetric key and session key. This random matrix provides randomness of our session key. The encryption technique with two keys i.e., two times encryption with session key and symmetric key provides robustness of our technique. At last we have included an authentication mechanism in our technique and provide the proof of identities which enrich the robustness as well as beauty of encryption. Comparative statistical tests like entropy value and frequency test etc between

proposed technique and standard techniques 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 than other proposed techniques. It is practically having minimal computational overhead during encryption and decryption.

REFERENCE

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. Ivan Daniel Burke, Renier van Heerden 2016, 'Automating Cyber Offensive Operations for Cyber Challenges', *11th International Conference on Cyber Warfare and Security: ICCWS2016,* At Boston, MA, USA.
18. Grant, T, Burke, ID & Van Heerden, RP 2012, 'Comparing Models of Offensive Cyber Operations', *Proceedings of the 7th International Warfare and Security*, pp. 108-121.
19. Mingxuan Li1, Zhushi Yang2,Ling He1, YangXin Teng, 'Research and Application of Information Security Offense and Defense Exercise in Electric Power Industry', 3rd Joint International Information Technology, Mechanical and Electronic Engineering Conference (JIMEC 2018).
21. Marie-Sarah Lacharit´e1, 'Security of BLS and BGLS signatures in a multi-user Setting': Cryptography and Communication (2018) 10:41–58, DOI 10.1007/s12095-017-0253-6.
22. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, HAND BOOK OF APPLIED CRYPTOGRAPHY, CRC Press, Inc. Boca Raton, FL, USA ©1996 ISBN: 0849385237.