Design of Triple Key Security Algorithm and Identification of Single-key Attack on Multiple Rounds in Mobile Communications

Muhammad Akram[1], Syed Ashraf Ali[2], C. A.Rahim[3]
[1]engineer_1564@yahoo.com [2] syedashrafali@yahoo.com
[3]abdulrahim@pakaims.edu.pk

The Institute of Management Sciences (PAK-AIMS), 54660 Lahore, Pakistan

Summary
In cipher algorithms, encryption and decryption is based on the same key. There are some limitations in cipher algorithms, for example in polyalphabetic substitution cipher the key size must be equal to plaintext otherwise it will be repeated and if the key is known then encryption become useless. This paper aims to improve the said limitations by a proposed algorithm TKSA in which the key is modified on polyalphabetic substitution cipher to maintain the size of key and plaintext. Each plaintext character is substituted by alternative message. The mode of substitution is transformed cyclically which depends on a current position of the modified communication. Three keys are used in encryption and decryption process on 8 or 16 rounds with the XOR of 1st key. This study also identifies a single-key attack on multiple rounds block cipher in mobile communications and applied the proposed technique to prevent the attack. By utilization of the TKSA algorithm, the decryption is illustrated, and security is analyzed in detail with mathematical examples.

Keywords:
Camera ready paper, TEM Journal.

1. INTRODUCTION

An Information plays very vital role for any organization since it is its asset, and hence must be protected from the illegal access. If confidentiality, integrity or availability (CIA model) of any information is lost (compromised), then that information may be used for the purposes harmful to the respective organization. It becomes, therefore, very much necessary for any organization to make its data and information resources out of the reach of the illegal users by applying the cipher cryptography. Mostly the ciphers are based on simple functions such as round or iterated block cipher with repeated round function [1]. Encryption is widely used as an effective method to protect data and information in many real life applications. The cryptographic algorithms are used for encryption using symmetric and asymmetric key, some cipher use the same key for encryption and decryption. The keys may be equal with a simple transformation to go between the two keys [2]. A shared secret for encryption and decryption is used to maintain information, however this is main drawback of symmetric key encryption with respect to public key. The block diagram of both symmetric and asymmetric key algorithm is shown in figure 1 and Figure 2. In asymmetric cryptography, a public/private key pair is generated randomly to allow access to the public key. For huge data encryption symmetric algorithm is much faster than asymmetric, but it has some drawbacks that size of key should be equal to size of plaintext otherwise it is repeated continuously and causing repeated histogram. The classes of symmetric block cipher is shown figure 3, these classes are based on mode of operation and iterations. Symmetric-key-block-cipher are based on two calculations such as encryption and decryption which takes n bits of plain-text by providing the similar number of bits including k bits for mystery-key.

Mode of operation determines the methods of using block cipher to larger plaintexts, this mode is classified into deterministic and probabilistic as shown in figure 3. Such block-cipher methods are planned for mystery and crypto-graphic primitive which is recognized in figuring texts, these methods are called Electronic-Code-Book (E-C-B) [3]. Iterated Product Ciphers is categorized in unbalanced Feistel-cipher, Feistel-cipher and substitution-permutation-networks. In unbalanced Feistel the “left half” and the “right half” are not of equal size therefore theses networks are called generalized unbalanced Feistel which consist of a series of rounds. Whereas the Feistel-cipher is utilized by adding circular capacity on the normal content to gives encrypted content in which the block-cipher calculations use data-encryption standard (DES).

Manuscript received October 5, 2024
Manuscript revised October 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.10.21

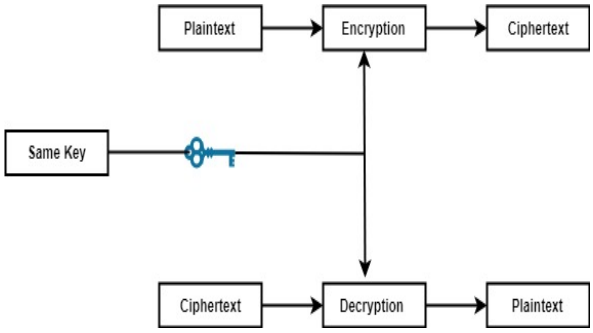

Figure 1. Symmetric key

It comprises of 64-bits as input and 64-bits as encryption along with 56-bits key-length. Another iterative base cipher is Substitution-Permutation-network in which substitution-change is part of block cipher. Substitution-stage is organized by including the round box or S-boxes. The most broadly utilized symmetric-cipher is Advanced-Encryption-Standard (AES) with 128-block block-size and incorporate 3 key-lengths more likely than not upheld 128, 192 and 256 blocks. If the key size is greater than block size with a uniform distribution, there are more than one key and if a block cipher is designed randomly, then the key space should be into same classes. So there is requirement to recover the key of block cipher. Another utilization of block cipher is in mobile communication particularly for 3GPP evolved radio access and GSM networks, the 8 round block ciphers are used [4].This block cipher is called KASUMI and it has 64-bit block and 128-bit for key with nonlinear S-boxes. Two functions, FO and FL are composed for each round performing logical operations with sub keys. However there is a single key attack against reduced-round with respect to complexity. In case of asymmetric key encryption, the public key is available to anyone to encrypt the plaintext on the network and only authentic user can decrypt by using the secret private key [5].

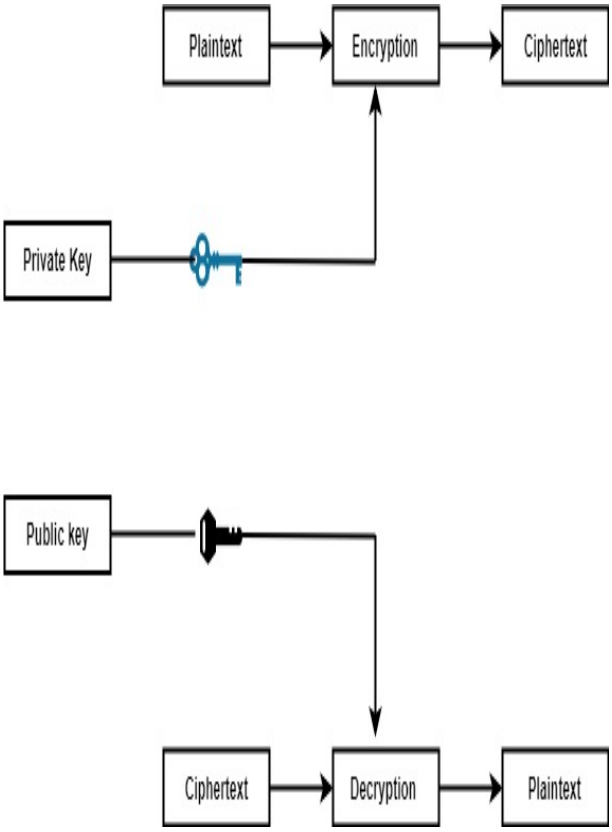

Figure 2: Asymmetric Key

In asymmetric key encryption algorithm, computationally different steps are involved for example in scenario of sender A and receiver B the following steps are used with different terms and conditions

a) Sender A and receiver B must know the public key while the private keys are secret

b) A can encrypt the Plain Text by using B's public key

c) A can transmit Cipher Text to B

d) By using the private key B can receive the cipher text

Finally the Plain Text message is received by B.

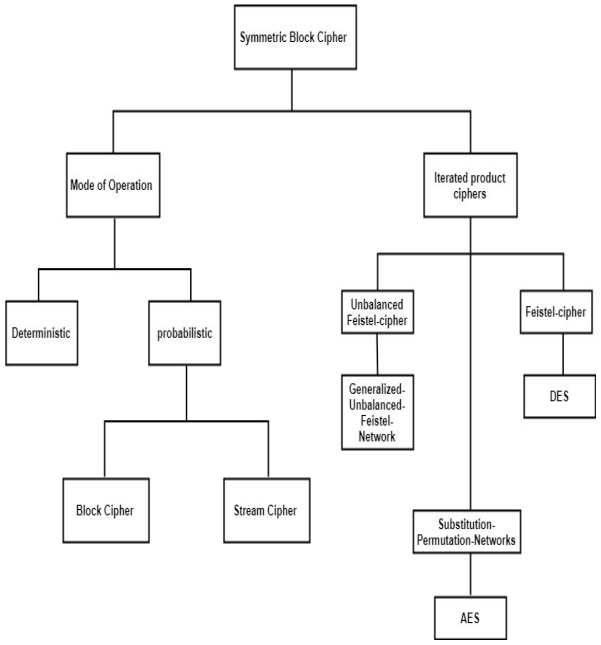

Figure 3. Symmetric Block Cipher classes

The problem of asymmetric encryption works slower as compared to symmetric encryption. Most asymmetric algorithms depend on the properties of hard problems in mathematics. These problems usually work intensive in one direction and nearly impossible in the other direction. For example, factoring the product of two large prime numbers. If one of the prime number is known then factoring becomes easy. But by knowing only the product it is very difficult to factorize and find the prime numbers. So in case of asymmetric algorithm, speed and more computationally costly is major drawback of using public key cryptography. In this study an algorithm Triple Key Security Algorithm (TKSA) is proposed to overcome the said limitations and drawbacks of both symmetric and asymmetric. The organization of this study is based on sections i.e., literature review is discussed in section II, comparative study and problem identification is discussed in section III, the proposed algorithm TKSA is demonstrated in section IV, implementation method of proposed algorithm is defined in section V, implementation of TKSA against attacks on multiple rounds block cipher is mapped in section VI, results and analytics is discussed in section VII and conclusion is presented in section VIII.

2. Literature Review as Related work

Previously various security techniques and encryption algorithms have been carried out for information security. To understand this study the following security algorithms are discussed for a comparison between suggested implementation of triple key security algorithm.

2.1. Symmetric Key Cryptography

In the symmetric key encryption with some enhanced techniques [6] uses key generation by random number in algorithm as the concept of internal key generation for the size of 512 bits at receiver end. In this technique the sender may store the internal key and send via another path.

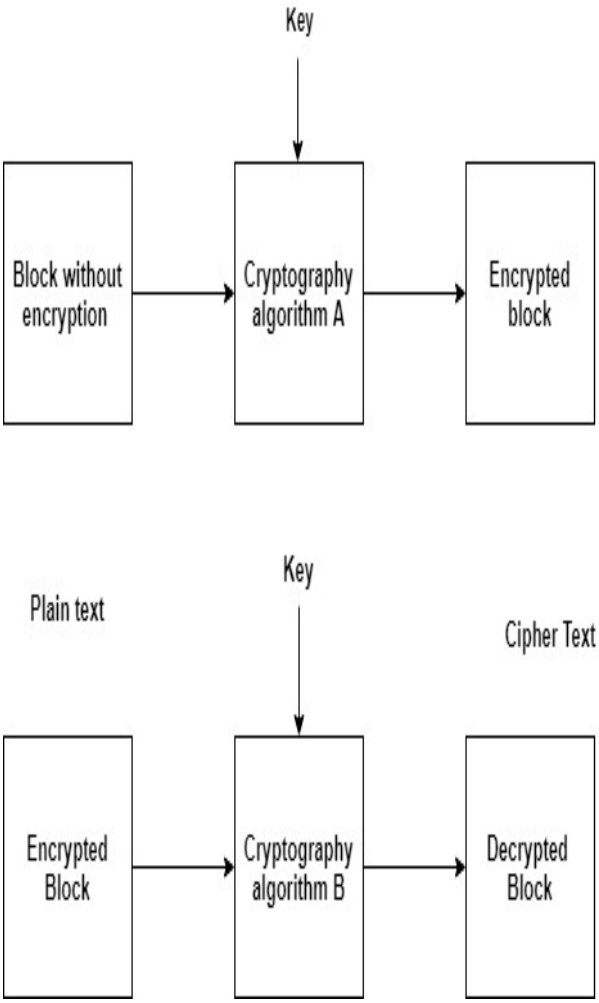

Figure 4: Basic Concept for Symmetric cryptography

Basically it is substitution method which is using block based technique for encryption of multiple times message where the proposed key blocks contain ASCII code from 0 to 255 in a random order for all possible words and characters. By using 256*2=512 bit key size for encryption of a text message at receiving end for decryption of any file, the receiver has to know the key

blocks and to apply 2512 trial run. So this algorithm has following lengthy steps

1: To define variable length
2. To calculate random number
3. Calculate variable Total
4. Conversion into binary String
5. Calculation of random value
6. Selection of another variable value
8. Calculation of encryption number.
9. Calculation of encryption value
10. Selection of another variable to represent as an encryption number.
11. Availability of random number and encryption number
12. To store binary string into table for next iteration.
13. Exit

This algorithm is based on block cipher method which takes less time for 2 Mb file size [6] however Key transportation has a major problem, due to this the communication channels may be taped.

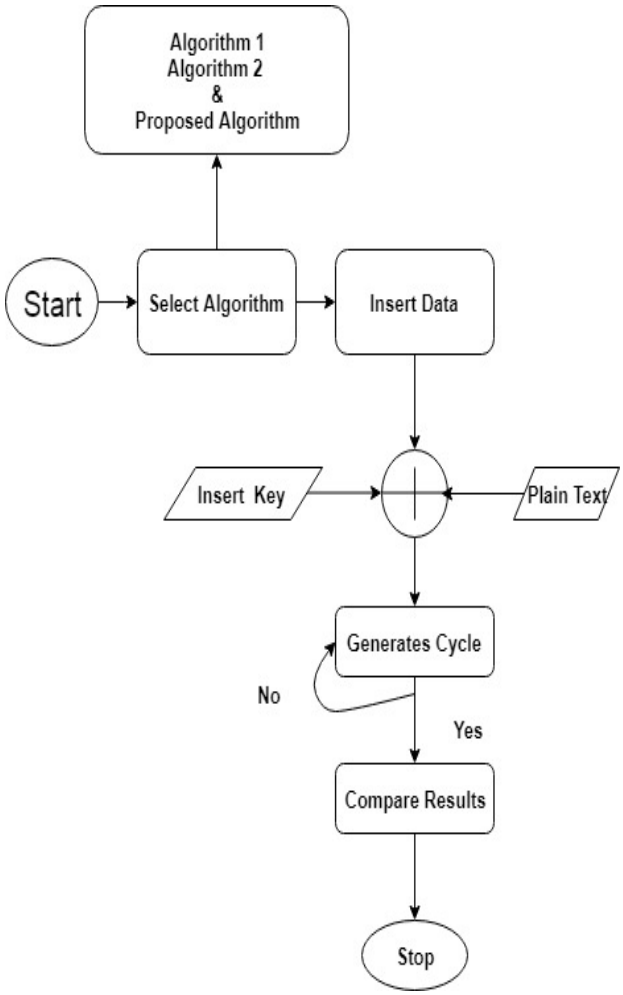

Figure 5: Result Evaluation Model

2.2. Asymmetric Key Cryptography Network based Security model

The asymmetric encryption algorithm includes RSA [7], Diffi-Hellman algorithm [8]. Asymmetric ciphers are often 2-3 orders of magnitude slower than the symmetric ciphers. Public key schemes are neither more secure than private key (security depends on the key size for both), nor do they replace private key schemes (they are too slow to do so), rather these complement private key ciphers. One of the best-known & widely used public-key asymmetric encryption algorithms is RSA, which uses large integers (e.g. 1024 bits) and its security is due to cost of factoring large numbers. Cryptographic techniques are also divided into two broad categories depending upon how the plaintext is encrypted. Stream ciphers process messages a bit or byte at a time when encrypting or decrypting. A stream cipher processes input elements continuously, producing output one element at a time. Network based security model uses deffi-hellman algorithm for sharing secret key with AES-256 for key generation [9]. The process begins with a private key then a public key is generated which is a derivative of the private key. Sender A and receiver B then exchange their public keys and now both has their own private key and the other systems public key. Initially the shared secret key can be used in the AES as the round key which encrypts, transmits and the distant end decrypts. This method is limited by its inverse cipher which takes more codes.

3. Comparative study and problem identification

There are some popular parameters used to compare the encryption decryption algorithms. Key management is important feature of the algorithm because the encryption process is completed using this key data. The size of key and generating process of key act their role in the process of encryption. The symmetric encryption algorithm uses the same key for encryption and decryption but in Asymmetric algorithm different keys are used for encryption and decryption. Throughput is the parameter that indicate the power consumption of the algorithm. If throughput increases, the power consumption decreases. Blowfish of symmetric encryption technique is have very high throughput as compares to others. Tenability is used to define encrypted parts. There is no tenability in symmetric encryption algorithm except of the blowfish but the asymmetric encryption algorithm used the tenability. The encryption ratio is specify the measurement of the amount of data to be encrypted. To reduce the complexity on computation the encryption

ratio should be minimized. It is high in both symmetric encryption algorithm and asymmetric encryption algorithm but can be moderate in 3DES technique of symmetric encryption algorithm. The cryptography techniques used these days are very many and growing day by day as are the cryptanalysis attacks. Many people have developed cryptographic systems to get the data and information protected from the unauthorized persons. Several algorithms offer different levels of strength for various applications. Most of the ciphers are either very complex, or less efficient. The computations, permutations, and other operations require a considerable effort and time. It is therefore the need to combine their strengths to make our real life systems more secure, more reliable, and more efficient. TKSA algorithm is designed and implemented to secure communication on the networks. The proposed algorithm is expected to play its role for the much needed secure systems in our daily life. In this research the algorithm is designed for both network security and data security which is based on the symmetric key cryptography. TKSA algorithm is a block cipher and uses the same key for encryption and decryption. The size of the data block in the designed algorithm is 128 bits. Three keys are used in encryption and three for decryption process. The size of the first key is 128 bits, size of the second key is 64 bits and the size of the third key is also 64 bits.

4. Proposed TKSA Algorithm

Communication have certain importance in our life. The main and important sources of communication are computer networks and mobile. People can share personal information and other important data through internet and mobile networks. It is important for an organization and company to protect its user's information from unauthorized access. A company and an organization can secure their E-communications by using different cryptographic techniques and different security algorithms. To make the communication secure different algorithm are designed. There are two types of the security algorithms 1) symmetric algorithms 2) asymmetric algorithms. In symmetric algorithm the user uses the same key for encryption and decryption, DES and AES are the two important symmetric key cryptographic algorithms. In asymmetric key cryptography user, a uses the public key of user B to encrypt the message and user B uses its private key to decrypt the message. Asymmetric key Algorithms use the public and private key for encryption and decryption. RSA, Al-Gamal are the important asymmetric key algorithms. The designed algorithm is based on the symmetric key cryptography. The designed algorithm is a block cipher and uses the same key for encryption and decryption. The size of the data block in the designed algorithm is 128 bits. Three keys are used in encryption and decryption process. The size of the first key is 128 bits and size of the second and third key (sub-key) is 64 bits. Algorithm is based on the 8 or 16 Iterations or 8 or 16 Rounds. Each round consists of 128 bits of data, three keys and two functions. First Round takes the 128 bits of data and divide the data into two chunks or into two units of 64 bits. These two data units are the input to the function-1and function-2. Function-1 and Function-2 perform different operation on data and key is added to the data. Both the function produced the output and this output is swapped and in the end key-1 added to data which will generate the Cipher text. Similarly, 8 round are executed and in the end algorithm generates the cipher text. There are three keys are used in this algorithm subkey-1 and subkey-2. Key-1 is generated from the shared secret and subkey-1 and subkey-2 is generated from first and second chunk of 64 bits of key-1.subKey-1 and subkey-2 are used for the function one and function two respectively. Round ends with the XOR of the key-1 with data from both functions. Working Process of the Key Generation can be explain with following steps

i. Input:

Input is the 128 bits of secret code. The key generation function will perform the following operation to produce the 8 different round keys.

ii. CON:

CON mean constant. The key generation function will generate the CON by XORing all the bytes of the key. CON will be a byte in size. After finding the value of the CON, each byte of the key XORed with the CON. The whole process is shown in the above diagram. At the end of this function the whole vector or sub arrays of bytes is divided into 4 equal chunks of data. The length of each chunk will 32 bits.

iii. CMP&AND:

After dividing the key into four parts. Take the complement of the first array A1 and select the last byte of the first array A1 and perform AND operation and save the result in new array A21. A11 XORed with all other arrays A12, A13, A14 and generate new arrays A22, A23, A24.

iv. Circular Shift:

After combining the Arrays A21, A22, A23 and A24 we will get a vector and perform circular shifting on the Vector. Again divide the array into four parts and take the complement of the first subarray and perform XOR with all other the array. Combine the sub array into a vectors and perform circular sifting two time.

v. Matrix Operations:

The resultant vector is then converted into matrix of 4*4 and perform circular shift on the matrix. There are two types of shifting one is bottom-up and second is the right left (row wise shift and column wise shift)

vi. Matrix vector:

Convert the matrix into four sub arrays and convert these arrays into a vector of 128 bits. The positions of the subarrays are random. For example, we have sub arrays 1,2,3,4 and arrangement of the array in vector form will be 4,2,3,1.

Subkey-1:

For sub key-1, divide the matrix vector into two parts and first part for the key-1 and second part for the key two. For sub key-1 from part one, determine the constant value CON and perform XOR with all other bytes in the first part of the vector. Further by performing the following operations we will get the sub key-1.

- Reverse the 64 bits' vector
- Perform two time circular shifts

Subkey-2:

For sub key-2, divide the matrix vector into two parts and first part for the key-1 and second part for the key two. For sub key-1 from part two, determine the constant value CON and perform XOR with all other bytes in the second part of the vector. Further by performing the following operations we will get the sub key-2.

- Perform two time circular shifts

Round key:

The round key can be obtained by the following operations.

- Combine the results of the sub key-1 and sub key-2 into a vector as (subkey-1, subkey-2).
- Perform two-time circular shift on the matrix vector
- Perform XOR between sub key vector and matrix vector.
- Perform two-time circular shift and we will get the final key for the round.

5. Implementation Method of Proposed Algorithm

The TKSA can be implemented by following steps

1. Substitution

Replacement of plaintext with cipher text is known as substitution, the units of the plaintext are rearranged in complex order. Substitution-box (s-box) [10] are used to hide the relationship between the key and cipher text. To implement TKSA algorithm the matrix relation S-box for encryption and inverse s-box decryption is used.

2. S-Box Generation

Following method is used to generate the s-box.

$$K = \begin{bmatrix} 1 & 3 \\ 2 & 7 \end{bmatrix} \quad (1)$$

Here k is the key matrix. Determinant of the key matrix should not zero. Following steps are used in the construction of S-BOX

a) Take the input byte 000000000
b) Divide the input into two parts 0000 and 0000
c) Find the inverse of 00 and second part remain same (1111 0000)
d) Convert the above output 1111 into digits which is equal to 15 and 0000 will be 0
e) In matrix form $\begin{bmatrix} 15 \\ 0 \end{bmatrix}$
f) Use the relation MK mod16=$\begin{bmatrix} 15 \\ 13 \end{bmatrix}$
g) In binary form 1111 1101 which is equal to fd

TABLE 1. S-Box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F D	1 4	3 B	5 2	7 9	9 0	B 7	D E	F 5	1 C	3 3	5 A	7 1	9 8	B F	D 6
1	E A	0 1	2 8	4 F	6 6	8 D	A 4	C B	E 2	0 9	2 0	4 7	6 E	8 5	A C	E A
2	D 7	F E	1 5	3 C	5 3	7 A	9 1	B 8	D F	F 6	1 D	3 4	5 B	7 2	9 9	B 0
3	C 4	E B	0 2	2 9	4 0	6 7	8 E	A 5	C C	E 3	0 A	2 1	4 8	6 F	8 6	A D
4	B 1	D 8	F F	1 6	3 D	5 4	7 B	9 2	B 9	D 0	F 7	1 E	3 5	5 C	7 3	9 A
5	A E	C 5	E C	0 3	2 A	4 1	6 8	8 F	A 6	C D	E 4	0 B	2 2	4 9	6 0	8 7
6	9 B	B 2	D 9	F 0	1 7	3 E	5 5	7 C	9 3	B A	D 1	F 8	1 F	3 6	5 D	7 4
7	8 8	A F	C 6	E D	0 4	2 B	4 2	6 9	8 0	A 7	C E	E 5	0 C	2 3	4 A	6 1
8	7 5	9 C	B 3	D A	F 1	1 8	3 F	5 6	7 D	9 4	B B	D 2	F 9	1 0	3 7	5 E
9	6 2	8 9	A 0	C 7	E E	0 5	2 C	4 3	6 A	8 1	A 8	C F	E 6	0 D	2 4	4 B
A	5 F	7 6	9 D	B 4	D B	F 2	1 9	3 0	5 7	7 E	9 5	B C	D 3	F A	1 1	3 8
B	4 C	6 3	8 A	A 1	C 8	E F	0 6	2 D	4 4	6 B	8 2	A 9	C 0	E 7	0 E	2 5
C	3 9	5 0	7 7	9 E	B 5	D C	F 3	1 A	3 1	5 8	7 F	9 6	B D	D 4	F B	1 2
D	2 6	4 D	6 4	8 B	A 2	C 9	E 0	0 7	2 E	4 5	6 C	8 3	A A	C 1	E 8	0 F
E	1 3	3 A	5 1	7 8	9 F	B 6	D D	F 4	1 B	3 2	5 9	7 0	9 7	B E	D 5	F C
F	0 0	2 7	4 E	6 5	8 C	A 3	C A	E 1	0 8	2 F	4 6	6 D	8 4	A B	C 2	E 9

3. Inverse Substitution:

Inverse substitution is the reverse of the substitution done in the encryption process and in decryption process we will use the inverse substitution.

4. Inverse S-Box Generation:

Following method is used to generate the inverse s-box.

$$KI = 1/1\begin{bmatrix} 7 & -2 \\ -3 & 1 \end{bmatrix} \quad (2)$$

Here KI is the inverse key matrix. Following steps are used in the construction of inverse S-BOX

- Take the input from the s-box which is fd and in decimal form {15:13}
- Take the product of key matrix and message matrix.
- The result of the matrix product is {15:0}
- Take the inverse of the output 15 in binary 0000 and 0000 remain same
- The inverse of the fd is 00

TABLE 2. Inverse S-box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F0	11	32	53	74	95	B6	D7	F8	19	3A	5B	7C	9D	BE	DF
1	8D	AE	CF	E0	01	22	43	64	85	A6	C7	E8	09	2A	4B	6C
2	1A	3B	5C	7D	9E	BF	D0	F1	12	33	54	75	96	B7	D8	F9
3	A7	C8	E9	0A	2B	4C	6D	8E	AF	C0	E1	02	23	44	65	86
4	34	55	76	97	B8	D9	FA	1B	3C	5D	7E	9F	B0	D1	F2	13
5	C1	E2	03	24	45	66	87	A8	C9	EA	0B	2C	4D	6E	8F	A0
6	5E	7F	90	B1	D2	F3	14	35	56	77	98	B9	DA	FB	1C	3D
7	EB	0C	2D	4E	6F	80	A1	C2	E3	04	25	46	67	88	A9	CA
8	78	99	BA	DB	FC	1D	3E	5F	70	91	B2	D3	F4	15	36	57
9	05	26	47	68	89	AA	CB	EC	0D	2E	4F	60	81	A2	C3	E4
A	92	B3	D4	F5	16	37	58	79	9A	BB	DC	FD	1E	3F	50	71
B	2F	40	61	82	A3	C4	E5	06	27	48	69	8A	AB	CC	ED	0E
C	BC	DD	FE	1F	30	51	72	93	B4	D5	F6	17	38	59	7A	9B
D	49	6A	8B	AC	CD	EE	0F	20	41	62	83	A4	C5	E6	07	28
E	D6	F7	18	39	5A	7B	9C	BD	DE	FF	1F	31	52	73	94	B5
F	63	84	A5	C6	E7	08	29	4A	6B	8C	AD	CE	EF	00	21	42

5. FUNCTION-1 (ONE):

Function one consist of the following functions:

- A.C XOR A+C
- Reverse
- Circular Shift
- Swapping
- Message Constant
- Key Constant

A.C XOR A+C:

In this case the input data which is A & C denotes the key value. First find the A.C and then A+C. perform the XOR between A.C and A+C. C is the sub key for the function-1. Reverse: Let suppose the output of the A.C XORA+C is 123456 and reverse is 654321. By applying key derivation function a secret and random derivation function key (DF key) can be generated with three inputs and producing three output keys such as chain key, constant key and message key [11] as shown in figure 6. Constant (key): The function-1 will generate key constant by XORing all the bytes of the key. CON will be a byte in size. After finding the value of the CON, each byte of the Message is XORed with the CON. Constant (Message): CON mean constant. The function-1 will generate Message constant by XORing all the bytes of the message. CON will be a byte in size. After finding the value of the CON, each byte of the Message is XORed with the CON. Swapping: Let us have a data 1234567891. Divide the data into two parts 12345 and 67891. After swapping we have the data 6789112345. Substitution: The substitution box to substitute the values in data vector are used. Function-2 consists of the following processes.

- Substitution
- Message constant
- Reverse Message constant
- Key constant

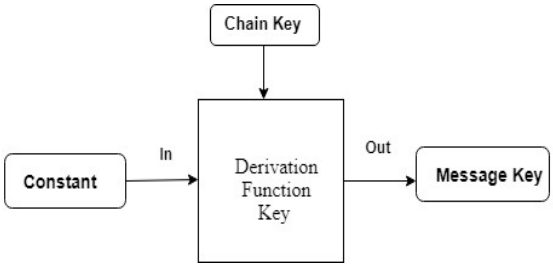

Figure 6: Chain key derivation function for Constant and message

For Substitution, We will replace the values in the input data to the values from s-box.

For Key constant: CON mean constant. The function-2 will generate key constant by XORing all the bytes of the subkey-2. CON will be a byte in size. After finding the value of the CON, each byte of the Message is XORed with the CON. Message constant: CON mean constant. The function-2 will generate Message constant by XORing all the bytes of the message. CON will be a 2 bytes in size. After finding the value of the CON, each 2-bytechunk of the Message is XORed with the CON. Reverse Message constant:The function-2 will generate Message constant by XORing all the bytes of the message. CON will be a 2 bytes in size. After finding the value of the CON, we will find the reverse of the CON and then each 2-byte chunk of the Message is XORed with the CON. Round Cipher (Final): The round cipher is shown in figure 7, each cipher and a number of blocks to create a function in multiple times with following steps

- Combine the output from the function 1&2.
- Perform XOR operation with the round key.
- 3time circular shift to output after XOR operation.
- In end we have round cipher.

Inverse Round Cipher: In inverse round cipher is shown in figure 8, input will be the cipher text from encryption algorithm or from the previous round. Input cipher will be 128 bits and in inverse round cipher algorithm have to perform the following function.

- Take 128 bits of input cipher text
- Perform 3time left circular shift
- Divide the data into 2 parts

Second part of data will be the input to the function-1 and first part will be the input to the function-2The function-1 and function-2 requires inversion, for inverse function-1 by using s-box and circular right shift reverse to generate plain text output. Function-1 replaced the data byte with other byte from the substitution table. But in decryption algorithm, replace the byte of data with the original data

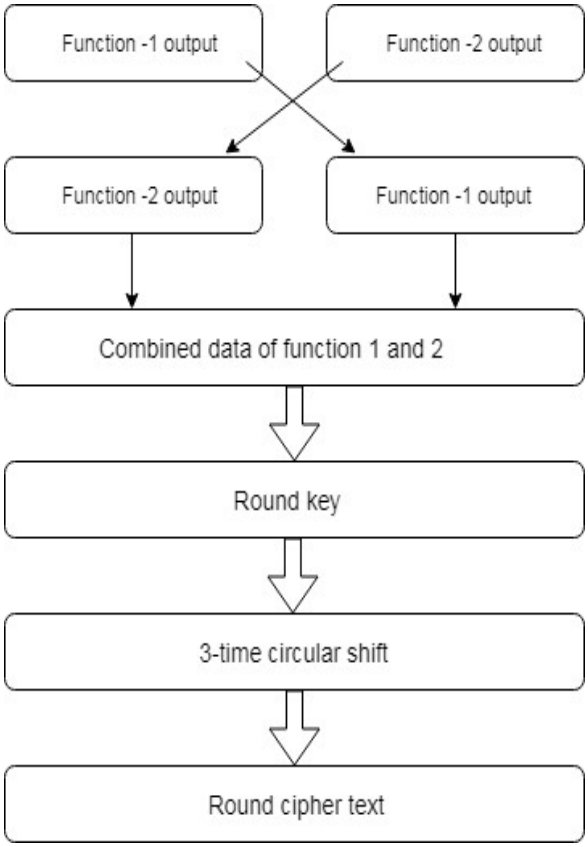

Figure 7: Encryption Final Round

The inverse function-1 will generate key constant by XORing all the bytes of the subkey-1 (according to the round sub key). CON will be a byte in size. After finding the value of the CON, each byte of the Message is XORed with the CON. The whole process is shown in the above diagram for the inverse function-1. Then the function-1 will generate Message constant by XORing all the bytes of the message. CON will be a byte in size. After finding the value of the CON, each byte of the Message is XORed with the CON. First step of the function -1 is the operation-1. In operation we will take the 64 bits of data and 64 bits of key. Encrypted message can be obtained by the following relation.

Y=A.C XOR A+C	(3)

Here A is message, C is the key and Y is the encrypted message. In decryption algorithm we will use the following relation to recover the message A

$A=Y^{-1}.C^{-1}\ Y^{-1} \oplus C^{-1}$	(4)

Here A is the message with 64 bits of length and C is also 64bits. The reverse function can be explain by example, let suppose the output of the A.C XORA+C is 123456 and reverse is 654321 by using both right and left circular shifts. For inverse function-2 key constant, message constant and reverse message constant via inverse substitution is used.

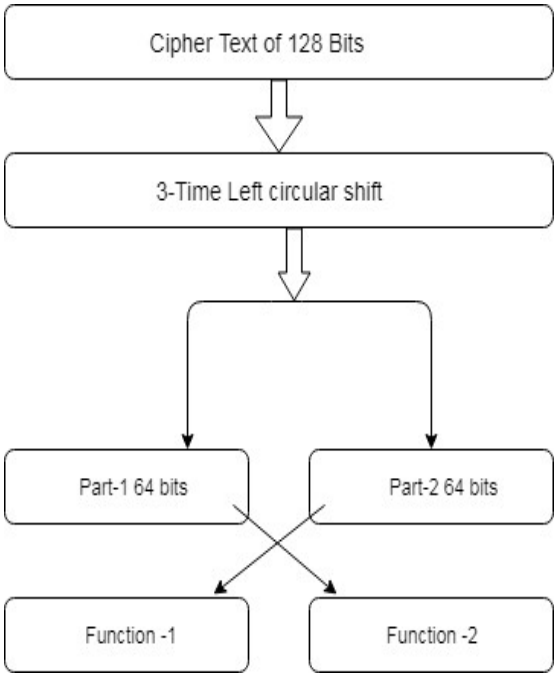

Figure 8: Decryption inverse Round

The algorithm will replace the data values in the input data to the values from inverse s-box in order to recover the original text from the cipher text. With reference to key constant the function-1 will generate key constant by XORing all the bytes of the subkey-2 (according to round subkey). CON will be a byte in size. After finding the value of the CON, each byte of the Message is XORed with the CON. The whole process is shown in the above diagram for the inverse function-2. With reference to message constant the inverse function-2 will generate Message constant by XORing all the bytes of the message. CON will be a 2 bytes in size. After finding the value of the CON, each 2-byte chunk of the Message is XORed with the CON. The whole process is shown in the above diagram for inverse function-2. For reverse message constant the inverse function-2 will generate Message constant by XORing all the bytes of the message. CON will be a 2 bytes in size. After finding the value of the CON, we will find the reverse of the CON and then each 2-byte chunk of the Message is XORed with the CON. The whole process is shown in the above diagram for inverse function-2. For example let suppose the output of the A.C XORA+C is 123456 and reverse is 654321 by implementing circular shifts.

6. Implementation of TKSA against Attacks on Multiple Rounds Block Cipher

There are different security attacks for the block cipher like rectangle-attack, sandwich-attack and single-key attack. The rectangle attacks for related-key setting are powerful also for key-recovery attacks [12]. The sandwich attack is composed of a single S-box layer this attack uses a distinguisher with three layers to have high probability characteristics [13]. A single-key attack on 16-round applies higher order differential attacks which is the most powerful against reduced-round with time complexity.

Proposed work on Single Key Attack

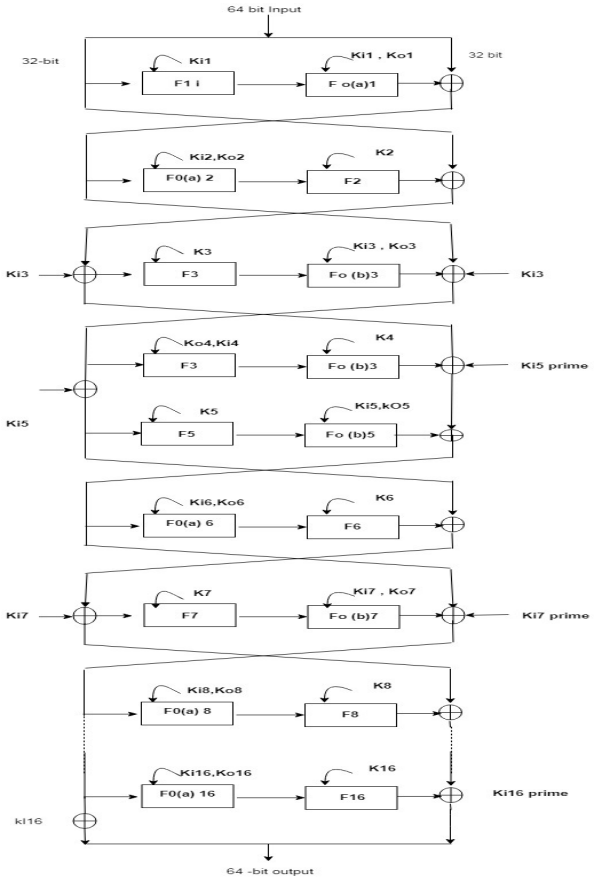

Figure 9: TKSA for 16- rounds

To implement TKSA for 16- rounds, the function is divided in to 4 rounds of operation with 16-bit input 16-bit for output. The function is processed through even and odd numbers of rounds respectively with 9 bit for input on left and 7 bit for right. The keys of F function are utilized for each round with sub key K of 16-bit. For 32 bit key, two keys Ki1 and Ki2 are further driven by 32 bit key K1 and K2. Mathematically defined as

$$K1 = Ki1 \| Ki2 \quad K2 = Ki2 \| Ki3 \tag{5}$$

Likewise with respect to above equations the three keys can be formed as

$$K3 = (Ki1\,[n] \| Ki2\,[n]) \oplus (Ki1\,[n+2] \| Ki2\,[n+2]) \tag{6}$$

In above equation n indicates the total number of rounds, in our case 16 rounds to be treated for this each round has a single key like $K1$ to $K16$. Let start with 0th round if

$$[n + (\text{integer})] > 15$$

then the most significant bit can be covered and for nth round to produce the key with reference to $K1$ to $K3$. As the TKSA is based on the S-box as security module of SNOW-3G algorithm as a reference Rijndael's S1-box for FSM [14]. To state two-way interaction formula or other statement the FO function is used within the s-box for remaining rounds as shown in figure 9. The 3rd round containing 16 bit is based on s-9 and s7 of s-box for the function of FO(a) which can be splinted in to 9-bit and 7-bit data for next rounds to rotate and cyclic the operation for 16 bit input data.

The output results are XOR after rotation and shifting with 16 bit input and output.

7. Results and Analytics by Mathematical Examples

The simulation results can be tested using MATLAB statistical NIST test while the mathematical examples are given to justify the results and analytics.

Plaintext= asdqweqweasdqwea

Key = asdqweqweasdqwea

Below the hex form of the message and key.

Message= 61 73 64 71 77 65 71 77 65 61 73 64 71 77 65 61

Key= 61 73 64 71 77 65 71 77 65 61 73 64 71 77 65 61

Round key and sub -keys are generated from the key generation algorithm.

keyround1 =17 00 8c 8c 8c 8b 14 98 15 89 15 8d 8d 12 9d 9a

subkey11 = ea f8 66 ec fc 63 fe ea

subkey21 =67 65 67 72 ed 75 65 66

keyround2 =01 11 ed ed ec 61 9c 73 9e fa 05 61 60 89 77 ec

subkey12 =62 66 17 77 60 14 76 76

subkey22 =12 89 9a 13 fa 15 9a 8a

keyround3 =88 9f ff ff 66 88 e8 9b ed 72 01 9b 02 76 05 fe

subkey13 =9b 06 8f 88 04 8d 05 04

subkey23 =9b 73 72 01 fe 8c 72 64

keyround4 =fb fc 73 73 9a e8 63 fa 73 16 98 fb 12 03 89 72

subkey14 = 64 13 e9 fd 01 fa 00 99

subkey24 =60 88 88 8d fb fb 88 16

keyround5 =fe 65 03 03 f8 60 05 73 8f 8a ec 76 8d 8f ea 02

subkey15 =99 11 63 71 11 66 10 f9

subkey25 = 17 e9 ed fd f8 67 ed 9f

keyround6 = ee 89 9f 9f 76 01 8a 13 63 eb f9 11 f8 67 e8 9e

subkey16 =67 17 01 99 16 02 17 ec

subkey26 = 03 65 66 61 61 03 66 fa

keyround7 =e9 f9 70 70 03 9b ee 88 10 ff 75 14 67 9e e8 71

subkey17 =8b 03 89 60 06 10 07 ee

subkey27 =15 03 9e 8e 9e 11 9e 67

keyround8 = f9 76 99 99 02 65 f9 77 16 fc 8e 8a 11 72 73 98

subkey18 =63 98 73 8c 9a 8c 9b e8

subkey28 =03 07 fb 67 70 8d fb 07

For the decryption of the cipher text obtained in the final round or in the round 8 we have to perform the following operation. For brevity only the decryption of round 8 will be performed. Divide the received data into two parts. First part is the input to the function2 and second part is the input to the functio1.

plaintext8 = 9e 8a 87 75 ae d2 79 58 2c 47 93 0c 36 da f3 31 is the input to the decryption function and algorithm have to perform 3-time right shift then we will get z95. z95 =75 ae d2 79 58 2c 47 93 0c 36 da f3 31 9e 8a 87 in the next step perform XOR operation between round 8 key and z95. The result will be.

Z95 =8c d8 4b e0 5a 49 be e4 1a ca 54 79 20 ec f9 1f divide the data obtained from Z95 into two equal parts of 64 bits.

forf1= e4 1a ca 54 79 20 ec f9

For 16 rounds each function can be explained at the encryption and decryption end. With respect to key length Larger the length key, higher will be the security. Key length is important to any encryption and

decryption algorithm. In our designed algorithm the length of the key is 128 bits. We have used three keys in the designed algorithm in order to encrypt and decrypt the data. key-1, the length of the key-1 is 128 bits.

Table 3 Comparison Table

Algorithm	**Round**	**Key Length**	**Sub-key length**
AES	10,12,14	128 Bits	No
DES	16	56Bits	No
TKSA	8,16	128Bits	Yes 64Bits

Table 4 Avalanche Effect with message change

Bit Changed	Changed Cipher Text	Original Cipher Text	No of bits changed
0	5a a8 b2 ee 58 c1 14 ba a7 7a 90 7d 04 cc 25 ce	5a a8 b2 ee 58 c1 14 ba a7 7a 90 7d 04 cc 25 ce	no
1	e1 21 80 67 d1 48 9d 33 a7 7a 90 7d 87 4f 25 81	5a a8 b2 ee 58 c1 14 ba a7 7a 90 7d 04 cc 25 ce	64
2	d5 9a 0f 53 e5 7c a9 07 a7 7a 90 7d 04 cc 25 ce	5a a8 b2 ee 58 c1 14 ba a7 7a 90 7d 04 cc 25 ce	44
3	e1 13 b2 55 e3 7a af 01 b7 52 90 3e 04 cc e4 ce	5a a8 b2 ee 58 c1 14 ba a7 7a 90 7d 04 cc 25 ce	49
4	64 96 37 17 af 74 69 c7 a2 18 5e 78 ca 01 ae 01	5a a8 b2 ee 58 c1 14 ba a7 7a 90 7d 04 cc 25 ce	79
5	30 c2 63 49 2d 20 bb 9d fa 16 d4 72 c6 8d a0 81	5a a8 b2 ee 58 c1 14 a a7 7a 90 7d 04 cc 25 ce	64

The other two keys are sub keys i.e. Key-2 and key-3, the length of each sub key is 64 bits. More length of the key, makes the brute force attack less feasible. Therefore, in the designed algorithm, the two sub keys are derived from the key-1 and the length of each sub key is 64 bits. These sub key used for the encryption and decryption in the different functions. A final area of block cipher design, and one that has received less attention than S-box design, is the key schedule algorithm. With any Feistel block cipher, the key is used to generate one sub key for each round. In general, we would like to select sub keys to maximize the difficulty of deducing individual sub keys and the difficulty of working back to the main key. The data block length in the designed algorithm is 128 bits. Higher the data block length higher will the data security.

Table 5 Avalanche Effect with key change

Bit changed	Original cipher text	Changed cipher text	No of bits changed
0	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	0
1	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	5c 4d f9 ee a3 19 6a 94 8c e1 b2 a1 ef 39 0f 43	53
2	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	57 2e 3e 8e 0f 11 46 38 c0 ee 7e 67 a0 90 00 41	62

3	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	99 c8 39 6a 25 1f aa d2 ca 66 b5 e5 a9 b5 0b cd	47
4	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	7a 64 d0 6b 24 10 a5 9a a0 4c 75 c4 e6 92 81 84	46
5	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	df 4e 03 cc 0e 0b 01 24 58 35 ea f9 7c e8 a1 82	68
6	74 c2 71 61 a1 3b a5 90 45 c6 75 a4 4c b6 4a 07	9c 47 57 e1 69 f0 07 13 6a ac 95 c8 e5 d2 a6 62	58

The data block in the designed algorithm will be divided into two parts and each part of 64-bit length. Input to the function one is the first chunk and input to the second function is the chunk 2 and the length of each chunk is 64 bits.

One noteworthy feature of this structure is that it is not a Feistel structure. Recall that in the classic Feistel structure, half of the data block is used to modify the other half of the data block, and then the halves are swapped. Two of the AES finalists, including Rijndael, do not use a Feistel structure but process the entire data block in parallel during each round using substitutions. AES can implement with 10, 12, and 14 round according the requirement of situation and available hardware. DES is designed with 16 round and the proposed algorithm TKSA can implement with 8 and 16 round. Key length of 128 bits used in AES and Key length of 56 bits used in DES, but the proposed algorithm TKSA (Three Keys Security Algorithm) implemented with three keys. The length of the key-1 is 128 bits. The other two keys are sub keys i.e. Key-2 and key-3, the length of each sub key is 64 bits. The comparison is shown in table 3. Avalanche affect for plain text changed is explained as that a change in one bit of the input should produce a change in many bits of the output. Let us have a message to encrypt “abcdefghijklm000”. This message is our original message. The following table show the number of bits changed in the original message and corresponding change in the cipher text. As the result shown in table 4, if only one bit will changed in original cipher text then there will be 64 bit change occurs. Avalanche affect (key changed), let a message to encrypt “abcdefghijklm000”. This message is an original message. The table 5 shows the number of bits changed in the original message and corresponding change in the key. Here key is”abcdefghijklm000”. By using S-box with rotational and cyclic processes to use S9 and S7 boxes, the TKSA algorithm make block cipher strong for discussed security attacks particularly single key attack. Keys with XOR and merge operations for each rounds.

8. Conclusion

The aims of this work is to improve the cipher algorithms limitations by a proposed a new algorithm TKSA in which the key is modified on polyalphabetic substitution cipher to maintain the size of key and plaintext. The designed algorithm has shown good avalanche effect. Algorithm have sufficient amount of nonlinearity. The designed algorithm has 8 or 16 steps for encryption and decryption which is comparatively less then Symmetric Key Cryptography and Asymmetric Key Cryptography Network based Security model in tram of performance. Each step is consists of two functions, function-1 and 2. Function-1 and function-2 consist of different operations which are to be performed on the data, for the encryption purpose. The performance of this algorithm is more efficient and fast as compare with Symmetric Key Cryptography and Asymmetric Key Cryptography Network based Security model. In feature a research work can be conducted on this algorithm for the evaluation of performance on different level of network and data security.

References

[1] Ritu Tripathi1, Sanjay Agrawal2 “Comparative Study of Symmetric and Asymmetric Cryptography Techniques” International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume 1, Issue 6, June 2014. ISSN 2348 – 4853

[2] H. Fathima, K.S.R. Matriculation & K.S.R. Kalvi nagar, “Comparative Study of Symmetric Key Algorithms-Des, AES and Blowfish”, International Research Journal Volume 17 Issue 2 Version 1.0 Year 2017

[3] Sangeeta & Er. Arpneek Kaur,” A Review on Symmetric Key Cryptography Algorithms”, Volume 8, No. 4, May 2017 (Special Issue)

[4] Rabie A. Mahmoud, A. Baith Mohamed, Magdy Saeb, “Enhancing KASUMI Security by Affixing A Metamorphic Function and the Ensuing Hardware Implementation”, International Journal of

Computer Science and Communication Security (IJCSCS), Vol. 6, January 2016
[5] Monika Agrawal, Pradeep Mishra, "Comparative Survey on Symmetric Key Encryption Techniques", International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 05 May 2012
[6] Krishna Kumar Pandey, Vikas Rangari & Sitesh KumarSinha, "An Enhanced Symmetric Key Cryptography Algorithm to Improve Data Security," International Journal of Computer Applications (0975 – 8887) Volume 74– No. 20, July 2013
[7] DiffiHellman,https://www.sciencedirect.com/topics/computer-science/diffie-hellman
[8] Gurpreet Singh, Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security", International Journal of Computer Applications, Volume 67– No.19, April 2013
[9] Elaine Barker, Allen Roginsky, "Transitioning the Use of Cryptographic Algorithms and Key Lengths", NIST Special Publication 800-131A Revision 2, March 2019
[10] D.Prasad ,G.P.Arya, C.Chaudhary, V.Kumar, "ENCIPHER A Text Encryption and Decryption Technique Using Substitution-Transposition and Basic Arithmetic and Logic Operation", International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014
[11] T Perrin, M Marlinspike, "The Double Ratchet Algorithm", revision 2016, available at https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf.
[12] B.Zhao, X.Dong, W.Meier, K.Jia, G.Wang, "Generalized Related-Key Rectangle Attacks on Block Ciphers with Linear Key Schedule", JOURNAL OF LATEX CLASS FILES, VOL. XXX, NO. XXX, JUNE 2019
[13] T Anand, M Shanmugam, B Santhoshini, "Rainbow table attack on 3rd generation GSM Telephony secure algorithm - A5/3", International Journal of Recent Technology and Engineering (IJRTE), Volume-7, Issue-5S4, February 2019
[14] Raja Muthalagu, Subeen Jain, "Modifying LFSR of ZUC to Reduce Time for Key-Stream Generation", Journal of Cyber Security, Vol. 5 4, 257–268, Publication 4 August 2017