I.J. Mathematical Sciences and Computing, **2017, 1, 38-48**
Published Online January 2017 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijmsc.2017.01.04

Available online at **http://www.mecs-press.net/ijmsc**

Modified DES using Different Keystreams Based On Primitive Pythagorean Triples

K.Mani [a], A.Devi [b]

[a] *Nehru Memorial College, Puthanampatti, Trichy, TamilNadu, India-621 007*
[b] Cavalier Animation and Media College, Bangalore, Karnataka, India-560 043

Abstract

Symmetric-key encryption is a traditional form of cryptography, in which a single key is used to encrypt and decrypt a message. In symmetric–key algorithm before any encrypted message is being transmitted, the sender and receiver must know the key value in advance. There are several drawbacks in symmetric-key algorithms. In some algorithms, the size of the key should be same as the size of the original plaintext and maintaining and remembering such a key is very difficult. Further, in symmetric-key algorithms, several round has to be performed to produce the ciphertext and perhaps the same key is used in each round which results in subkey generated from the current round is fully depending on the previous round. To avoid these, a novel approach in generating the key from the keystream for any symmetric-key algorithms using the Primitive Pythagorean Triples(PPT) has been proposed in this paper. The main advantage of this method is that the key value generated from the keystream is chosen by both the sender and the receiver. Further, the size of the key sequence is not limited but its size is arbitrary in length. Since, the keystream generated is random, no need to remember such keys by both the sender and the receiver.

Index Terms: Key Generation, Stream Cipher, Primitive Pythagorean Triples, Data Encryption Standard.

1. Introduction

As the demand for effective data security is increasing day by day, any organization has an obligation to protect secret and sensitive data from theft or loss. Such sensitive data can be potentially damaged if it is altered, destroyed, or hacked. This makes it necessary to protect the data. Cryptography attempts to provide such guarantee and it ensures the security of data being transmitted. It is the process of converting messages from a comprehensive form into an incomprehensive one at one end and which reverses the process at the other end so that the message is unreadable by interceptors or eavesdropper without the secret knowledge. It is

* Corresponding author. +91 9945270104
E-mail address: devianbu75@gmail.com

implemented in many day-to-day applications viz., the security of ATM card, computer passwords, e-commerce, military, etc. Cryptographic algorithms are divided into two types viz., symmetric-key and public-key algorithms. In symmetric-key (also called secret key) algorithm, same key is used for both encryption and decryption and it must be transmitted either manually or through a communication channel. A serious concern is that there may be a chance that an enemy can discover the secret key during transmission. Symmetric key algorithms are further divided into stream cipher and block cipher algorithms. A block cipher symmetric-key encryption algorithm transforms a fixed length block of plaintext data into a block of ciphertext data of the same length whereas a stream cipher operates typically on smaller units of plaintext, usually bits or bytes.

Stream ciphers are much faster than block ciphers. Most of the stream ciphers facing the problem of generating one random bit in each round of process as the output stream of cryptosystem [3]. This increases the risk of algebraic correlation against those cryptosystems [3, 4]. Vernam's one-time pad is one of the stream cipher which uses a string of bits which are generated completely at random. Since the entire keystream is random, even an opponent with infinite computational resources can guess the plaintext if he/she sees the plaintext. It is noted that though, Vernam's onetime pad is perfectly secured, remembering and storing such a key is too tedious because the size of the key is always taken as the size of plaintext and hence it is at least practical [5].

In DES symmetric-key algorithm, initially 64-bit key called DES-key is given as input and in each round a 56-bit round key has been generated to produce the ciphertext. Suppose, if the DES keys chosen are weak keys, the Hamming distance between a plaintext and a ciphertext produced by DES may be less which eventually results in the eavesdropper may easily recover the plaintext from the ciphertext. To overcome these, a PPT based key stream is generated in this paper and its length is determined by both the sender and the receiver as in the case of Vernam's one time pad. Similarly, to generate the DES key from the PPT based keystream, the starting position and the seed PPT (from which other PTs are generated) are only known by the parties who are involved in the communication systems which results in keystream from the key sequence is unpredictable. Further, different key is generated for each round from the PPT based keystream which differs from the traditional DES in which only one key is used in all rounds.

The rest of the paper is organized as follows. Section 2 presents the related work in generating the keystream. The definition of PPT, a method for generating the PPTs, properties of PPT are presented in section 3. Assignment of class labels to PPTs and the generation of key stream is discussed in section 4. Modified DES algorithm is discussed in section 5. An illustration of encryption process using DES with PPT's based key is shown in section 6, Finally, section 7 ends with conclusion.

2. Background and Related Work

The Linear Feedback Shift Register (LFSR) [1]is one of the most popular encryption techniques widely used in communication. But the main disadvantage of LFSR based structure is its vulnerability to attack due to inherent linearity in the structure [2]. Majid Bakhtiari and MohdAizainiMaarof [3] designed an efficient stream cipher algorithm to generate 115 random bits in one round of process which increases the resistance of process in front of Berlekamp-Massey, algebraic and correlation attacks. But, many computers are not able to generate random bits efficiently. RC4[6] is an important stream cipher in software application. The first weakness of RC4 is that a large number of bits of initial permutation which is determined by a small subset of key bytes. The second weakness is a key vulnerability, if a part of key is exposed to the attacker. Biham and seberry [8], presented a method called Rolling Arrays which contain variable rotations and permutations. But, the disadvantage is that the total 256 keys must be accumulated for initial permutation and the plain text is not encoded. The keystream is not depending on the plain text to be encrypted. An efficient key-pooled RC4 stream cipher is suggested by Kim et al.[9] for secure transmission of multimedia files in the wireless mobile network. In this method, a IBM-sized key stream pool is used with 32,768 keystream frames for every client device in the registration step is generated .It is more secured than the normal RC4. But, the drawback is the number of keys to be stored is huge. In[10], Sreelaja and Pai recommended an Ant colony optimization(ACO) method for creation of key stream which is used to distribute the characters in the plaintext for encryption. Artificial Ant's which do not discover counterparts with real ants and the encryption time is higher due to the phenomenon deposition is problem dependent. It does not reproduce real ant's performance. In [11], Minaam, Abdul-Kader, Hadhoud, proved that DES has better performance than 3DES. T. Muthumanickam [12] proposed that the Rijndael's SBoxes are the dominant element of the round function in terms of required logic resources. Each Rijndael round requires sixteen copies of the S-Boxes, each of which is an 8-bit×8-bit look-up-table, requiring more hardware resources. In [13], Mijanur Rahaman, Md. Masudul Islam proposed how essentially quantum based computation could change our classical processing and communication in cloud system. Also they are putting up all the major advantages with simple example and major problem with some recent progress in quantum computation. Erdem.S.S, Yanik, T, Ko [14] described a method for performing computations in a finite field GF(2N) by embedding it in a larger ring Rp where the multiplication operation is a convolution product and the squaring operation is a rearrangement of bits. Multiplication in Rp has complexity N+1, which is approximately twice as efficient as optimal normal basis multiplication (ONB).In [15], Longa, P., Miri, A. described an innovative methodology to derive composite operations of the form dP +Q by applying the special addition with identical z -coordinate to the setting of generic scalar multiplications over prime fields. They showed that their methods offer the lowest costs, given by 1I+(9L)M+(3L+5)S and 1I+(9L)M+(2L+6)S, when using only one inversion.

3. Definition, Method and Properties of Generating PPT

3.1. Definition of PPT

A Pythagorean Triple (PT) consists of three positive integers a, b, and c such that $a^2 + b^2 = c^2$. If a, b and c are co-prime then it is called Primitive Pythagorean Triple (PPT). All the solutions of the PT must satisfy the condition [10].

$$\gcd(a, b, c) = 1, a>0, b>0, c>0$$

are given by the formulas

$$a=2st, b=s-t^2, z=s^2+t^2$$

for integers s>0>r>0 such that gcd(s, t) = 1 and

$$s \bmod 2 \neq t.$$

An important fact is that a PPT always consists of all even numbers, or two odd numbers and an even number. A PPT can never be made up of all odd numbers or two even numbers and one odd number because the square of an odd number is always an odd number, the square of an even number is an even number, the sum of two even numbers is an even number and the sum of an odd number and an even number is an odd number. So, when both a and b are even, c is even too. Similarly, when one of a and b is odd and the other is even, c has to be odd.

3.2. Method for Generating PPT

If m and n are any two positive integers with m < n then $a = n^2-m^2$; $b = 2nm$; $c = n^2 + m^2$ then a, b, and c form a PPT. For example, if m=1 and n=2 then a=3, b=4 and c = 5**.** Thus, the first PPT (3, 4, 5) is obtained. Similarly, when m=2 and n=3, the next PPT (5, 12, 13) is obtained. To generate a PPT, from a pair of positive integers, one of them is selected as odd and the other is even and also relatively prime to each other. There are several methods available in generating PPTs. But, to generate PPT$_S$, Barning matrices are considered in this paper.

3.3. Barning Tree

PPT's are indexed and mapped in many ways and it is generated in the form of a tree starting from (3,4,5) called Barning Tree. The other PPTs are generated from the seed PPT(3,4,5) using the three matrix transformation as shown in the Fig.1.

$$T1= \begin{bmatrix} -1 & 2 & 2 \\ -2 & 1 & 2 \\ -2 & 2 & 3 \end{bmatrix} \quad T2= \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{bmatrix} \quad T3= \begin{bmatrix} 1 & -2 & 2 \\ 2 & -1 & 2 \\ 2 & -2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 2 & 2 \\ -2 & 1 & 2 \\ -2 & 2 & 3 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}\begin{bmatrix} 1 & -2 & 2 \\ 2 & -1 & 2 \\ 2 & -2 & 3 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$$

(i)(15,8,17) (ii)(21,20,29) (iii)(5,12,13)

Fig.1. Matrix Transformation

The Barning tree for the first three generation of PPTs are shown in Fig.2.

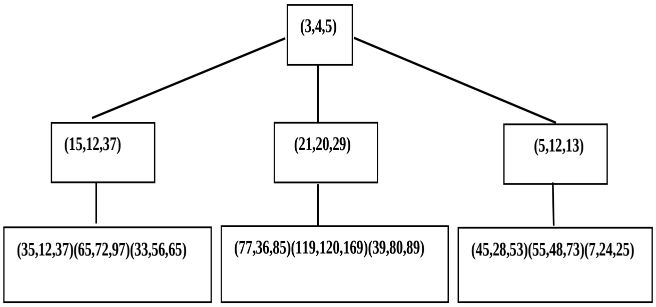

Fig.2. First Three Generations of PPT's

34. Properties of PPT's and assignment of class labels to PPT's.

In [7], Babylon explains the assignment of any one of the class labels viz., A, B, C, D, E, and F to each PPT based on the divisibility of a, b, c by 3, 4, and 5. To assign the class label to each PPT, let x denotes the PPT and w(x) denotes the class label assigned to corresponding PPT. The assignment of class labels to PPT is shown in Table 1. For example, in Table 1, row 1 indicates for a PPT x, if a is divisible by 3 and c is divisible by 5, then the class label w(x) is A. Thus, if x is (3, 4, 5) then w(x) is A. Assignment of class labels viz., B, C, D, E and F are assigned to the PPT's and an example for each type is shown from row to row. PPT's is shown in Table 1. For example, in Table 1, row 1 indicates for a PPT x, if a is divisible by 3 and c is divisible by 5, then the class label w(x) is A. Thus, if x is (3, 4, 5) then w(x) is A. Assignment of class labels viz., B, C, D, E and F are assigned to the PPT's and an example for each type is shown from row to row.

Table 1. Six Different Classes of PPT's and Examples

Class	A	B	C	x	w(x)
A	3	-	5	3,4,5	A
B	5	3	-	5,12,13	B
C	3&5	-	-	7,24,25	D
D	-	3	5	9,40,41	E
E	3	5	-	15,8,17	C
F	-	3&5	-	21,20,29	E

4. Assignment of Class Labels to PPT's and The Generation of Keystream

Before assigning class labels, the PPTs are sorted in ascending order on the basis of magnitude of PPTs. The sequence of class labels obtained from Table2 is ABDECEABEEEBAABDBFBCCBAFDCBDAFDABAFFDDEF. While generating the sequence of class labels, the same label may be repeated for n times continuously or n-1 times or even 1 time. Sometimes two, three etc., same labels may occur in the class. In order to generate a keystream from the sequence of class label, the starting position and the size of the key stream is decided only by the sender and the receiver. Since, they are determined by both parties, it provides an additional level of security. Moreover, the keystream is not necessarily remembered by both parties.

Table 2. Assignment of Class Labels to PPT's

x	w(x)	X	w(x)	x	w(x)	x	w(x)
3,4,5	A	51,140,149	E	105,208,233	C	217,456,505	D
5,12,13	B	55,48,73	B	115,252,277	B	273,136,305	A
7,24,25	D	57,176,185	A	117,44,125	A	275,252,353	B
9,40,41	E	63,16,65	A	119,120,169	F	297,304,425	A
15,8,17	C	65,72,97	B	133,156,205	D	299,180,349	F
21,20,29	E	77,36,85	D	165,52,173	C	319,360,481	F
33,56,65	A	85,132,154	B	175,288,337	B	377,336,505	D
35,12,37	B	91,60,109	F	187,84,205	D	403,396,565	D
39,80,89	E	95,168,193	B	207,224,305	A	459,220,509	E
45,28,53	E	105,88,137	C	209,120,241	F	697,696,985	F

5. Modified DES Algorithm

DES is one of the most popular symmetric-key block cipher algorithm to protect the data during transmission and storage. It consists of five functions viz., initial permutaion(IP), a complex function f_k (Both permutation and substitution operation), simple permutation function (switches(sw) the two halves of data), the function f_k again and finally, the reverse of initial permutation(IP^{-1}). DES can work on bits or bytes. It works on 64-bits blocks using 56-bits key sizes. Despite the keys are 64-bits long, every 8th bit in the key is not used. When the subkeys are created, the 8th bit gets eliminated. Sixteen rounds must be processed in DES which includes both permutation and substitution functions. At the end of sixteenth round, a pre-output will be created after swapping the left and the right block of the output. Then, reversing the order of two blocks into 64-bit b lock, ciphertext will be created after applying the final permutation. The inverse of encryption is decryption. The plaintext and ciphertext spaces of DES are $M=C=\{0, 1\}^{64}$. The DES keys are selected in such a way that if a 64-bit DES key is divided into eight bytes, then the sum of the eight bits of each byte is odd. This means that seven of the eight bits determine the value of the eighth bit. Transmission errors of one bit can be corrected. Thus, the key space is $K=\{(b_1,b_2,\ldots,b_{64}\}\{0,1\}$: $\sum_{i=0}^{7} b8k + i \equiv 1 \bmod 2$, $0 \leq k \leq 7$

It is noted that in conventional DES, only one 64-bits key is given as input and a 56-bits sub key is generated in each round from it. In the modified DES, while generating the first round key, eight characters (64-bits) are taken from PPT sequence of class label called keystream by giving the starting position where the staring position is determined randomly only by both the sender and receiver who are involved in the communication system. From the PPT based key sequence only 56-bits subkey is generated. For the second round, the starting position is again taken randomly by both the sender and receiver and the eight character of PPTs are taken accordingly and the next subkey is generated in similar manner. The process is repeated for all other rounds. Since, in each round a different keystream is taken from the key sequence, subkeys generated from it is also different and the relationship between the subkey generated from the current and previous rounds are highly unpredictable which eventually results in an another way of enhancing security. The proposed modified DES is shown in Fig. 3.

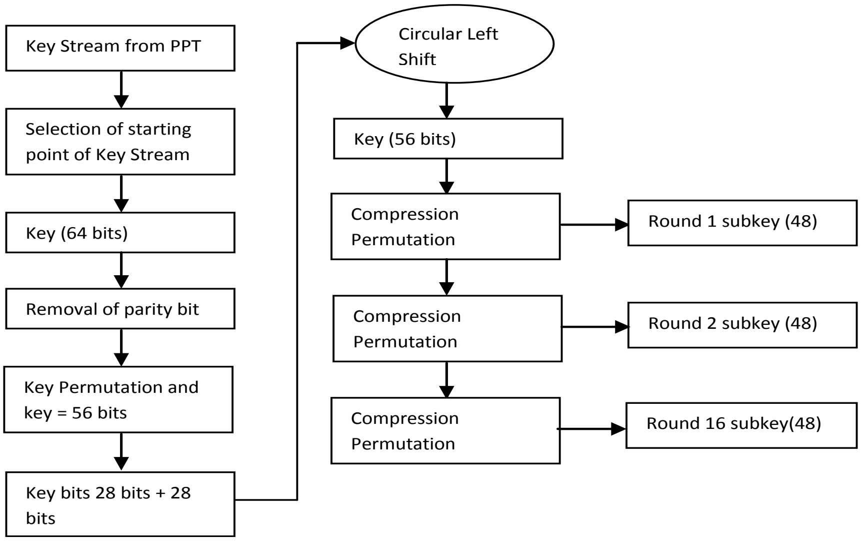

Fig.3. Modified DES

6. Encryption using DES with PPT's Based Keystream - An Example

To show the relevance of the proposed work, let the starting position of class label sequence is 2, then the keystream chosen from the class label by both the sender and receiver is **"BDECEABE"**. While selecting a DES key from the keystream, the ASCII value of each character is taken and then it is converted into binary.

The 8th bit is discarded because it is parity bit and only 7 bit is considered. To fill the value in 8th bit, as per DES key rule, count the total number of 1's up to 7-bit in binary form of the character. If it is odd then zero will be added to the LSB, otherwise one will be added. After obtaining the 64-bits DES key, the 56-bits round key is generated. or the second round, the other eight characters ($8 \times 8 = 64$ bits) are taken by considering randomly taking the starting position from the class label sequence as second keystream. This is because, in DES 64-bit key is taken as input and 56-bits round key is generated from it. Similarly, other keystreams are generated in this manner for the rest of the round. Totally, sixteen different keystreams are used to encrypt the message which is shown in Table3 and Fig3. Table 4 and table 5 show the generation of subkey and its corresponding ciphertext for round 1 using DES algorithm.

Table 3. Random Generation of Sequence from Class Label Sequence

Round	Starting Position	Key Stream
1	2	BDECEABE
2	15	BDBFBCCB
3	20	CCBAFDCB
--	--	-- -- -- -- -- --
15	36	ABAFFDDE
16	72	DAFDABAF

6.1. *Generation of first round key from first keystream*

Let the plaintext is "KANNANBA" and its binary equivalent is

M=0100 1011 0100 0001 0100 1110 0100 1110 0100 0001 0100 1110 0100 0010 0100 0001=64 bits

After applying the Initial Permutation (IP), then

IP(M)=1111 1111 0000 0000 0010 1100 1001 0011 0000 0000 0000 0000 0010 1101 0110 1101
Now, L_0=1111 1111 0000 0000 0010 1100 1001 0011
R_0=0000 0000 0000 0000 0010 1101 0110 1101

Table 4. Generation of First Round Key

R_i	KS_i	$ASKS_i$	$ASKS_iB$	$DESKS_i$	KS_iPC1	C_{i-1}	D_{i-1}	After Shifting		K_i
								C_i	D_i	
1	B	66	01000010	$1000010\underline{1}$	11111111	1111	1011	1111	0111	000011
	D	68	01000100	$1000100\underline{1}$	00000000	1111	1100	1110	1000	110100
	E	69	01000101	$1000101\underline{0}$	00000000	0000	0100	0000	1001	000100
	C	67	01000011	$1000011\underline{0}$	00001011	0000	1001	0000	0011	010001
	E	69	01000101	$1000101\underline{0}$	11000100	0000	1001	0000	0010	001110
	A	65	01000001	$1000010\underline{1}$	10011001	0000	0110	0000	1100	110010
	B	66	01000010	$1000010\underline{1}$	01100000	0000	0000	0001	0001	110100
	E	69	01000101	$1000101\underline{0}$						101001

R_i – ith round KS_i – Keystream for R_i $ASKS_i$ - ASCII value for KS_i $DESKS_i$ – DES KeyStream
C_i – Left order bits for ith round D_i – Right order bits for ith round K_i – ith Round key

6.2. *Generation of cipher text using first round key*

Table 5. Generation of First Round Ciphertext

K_i	L_{i-1}	R_{i-1}	$E(R_{i-1})$	$f(R_i ,K_i)$	$L_i = R_{i-1}$	S-Boxes	$Ri = L_i \oplus f(R_{i-1}, K_i)$
000011	1111 1111	0000 0000	100000	0010 1100	0000 0000	1100 1100	1101 0011
110100	0000 0000	0000 0000	000000	1010 1100	0000 0000	1001 0100	1010 1100
000100	0010 1100	0010 1101	000000	1001 1001	0010 1101	0111 0010	1011 0101
010001	1001 0011	0110 1101	000000	0011 1000	0110 1101	0000 1101	1010 1011
001110			000101				
110010			011010				
110100			101101				
101001			011010				

6.3. Generation of second round key from second keystream

The DES procedure is repeated, the second round subkey and the corresponding cipher text generated are shown in Table 6 and Table 7 respectively.

Table 6. Generation of Second Round Key

R_i	KS_i	$ASKS_i$	$ASKS_iB$	$DESKS_i$	KS_iPC1	C_{i-1}	D_{i-1}	After Shifting		K_i
								C_i	D_i	
2	B	66	01000010	0100001$\underline{1}$	0000001	0000	1111	0000	1111	001000
	D	68	10000100	1000010$\underline{1}$	0111111	0010	1101	0101	1010	001101
	B	66	01000010	0100001$\underline{1}$	0100000	1111	0000	1111	0001	001001
	F	70	01000110	0100011$\underline{0}$	0000000	1101	1010	1010	0100	000010
	B	66	01000010	0100001$\underline{1}$	1111110	0000	0000	0000	0000	001000
	C	67	01000011	0100001$\underline{1}$	1000010	0000	0000	0000	0000	110010
	C	67	01000011	0100001$\underline{1}$	1000000	0000	0000	0000	0001	000100
	B	66	01000010	0100001$\underline{1}$	0000000					010011

6.4. Generation of second round ciphertext

Table 7. Generation of Second Round Ciphertext

K_i	L_{i-1}	R_{i-1}	$E(R_{i-1})$	$f(R_i ,K_i)$	$L_i = R_{i-1}$	S-Boxes	$Ri = L_i \oplus f(R_{i-1}, K_i)$
001000	0000 0000	1101 0011	111010	0010 1001	1101 0011	1100 0100	0010 1001
001101	0000 0000	1010 1100	100111	1100 0001	1010 1100	1110 1010	1100 0001
001001	0010 1101	1011 0101	110101	1000 0100	1011 0101	1001 0000	1010 1001
000010	0110 1101	1010 1011	011001	0101 0101	1010 1011	1010 1000	0011 1000
001000			010110				
110010			101011				
000100			110101				
010011			010111				

6.5. *Generation of second round ciphertext for original DES with single keystream*

Table 8. Generation of Second Round Ciphertext by Using Original DES

K_i	L_{i-1}	R_{i-1}	$E(R_{i-1})$	$f(R_i, K_i)$	$L_i = R_{i-1}$	S-Boxes	$R_i = L_i \oplus f(R_{i-1}, K_i)$
001000	0000 0000	1101 0011	111010	1000 1010	1101 0011	1010 0000	1000 1010
001101	0000 0000	1010 1100	100111	1011 0111	1010 1100	0100 0001	1011 0111
001001	0010 1101	1011 0101	110101	0000 1110	1011 0101	0110 0010	0010 0011
000010	0110 1101	1010 1011	011001	1010 0000	1010 1011	0111 1111	1100 1101
001000			010110				
110010			101011				
000100			110101				
010011			010111				

The process is repeated for all 16 rounds, swapping and inverse permutation is applied to generate cipher text.

Table 9. Hamming Distance for Round1 and Round2

M	Modified C_1	DES C_2	HD $H(M\oplus C_1)$	HD $H(M\oplus C_2)$	Original C_1	DES C_2	HD $H(M\oplus C_1)$	HD $H(M\oplus C_2)$
0100 1011	0000	1101 0011	32	40	0000 0000	1101 0011	32	38
0100 0001	0000	1010 1100			0000 0000	1010 1100		
0100 1110	0000	1011 0101			0010 1101	1011 0101		
0100 1110	0000	1010 1011			0110 1101	1010 1011		
0100 0001	0010	0010 1001			1101 0011	1000 1010		
0100 1110	1101	1100 0001			1010 1100	1011 0111		
0100 0010	0110	1010 1001			1011 0101	0010 0011		
0100 0001	1101	0011 1000			1010 1011	1100 1101		
	1101							
	0011							
	1010							
	1100							
	1011							
	0101							
	1010							
	1011							

C_1- Ciphertext for R1 C_2- Ciphertext for R_2 M-Plain Text HD-Hamming Distance

It is noted that the Hamming distance H(M,C) where M is plaintext and C is ciphertext which is calculated for the first two round and the resultant is shown in Table 9. If H(M,C) is the avalanche effect and is more which results, the adversary may not easily recover M from C.

6.5.1. *Hamming Distance for Modified DES*

M = 0100 1011 0100 0001 0100 1110 0100 1110 0100 0001 0100 1110 0100 0010 0100 0001
C_1= 0000 0000 0000 0000 0010 1101 0110 1101 1101 0011 1010 1100 1011 0101 1010 1011
$H(M\oplus C_1)$ = 1 3 1 1 2 2 1 2 2 1 3 1 4 3 3 2 =32 bits
M = 0100 1011 0100 0001 0100 1110 0100 1110 0100 0001 0100 1110 0100 0010 0100 0001
C_2= 1101 0011 1010 1100 1011 0101 1010 1011 0010 1001 1100 0001 1010 1001 0011 1000

$H(M \oplus C_2)$= 2 1 3 3 4 3 3 2 2 1 1 4 3 3 3 2 =40 bits

6.5.2. Hamming Distance for Original DES

M = 0100 1011 0100 0001 0100 1110 0100 1110 0100 0001 0100 1110 0100 0010 0100 0001
C_1= 0000 0000 0000 0000 0010 1101 0110 1101 1101 0011 1010 1100 1011 0101 1010 1011
$H(M \oplus C_1)$ =1 3 1 1 2 2 1 2 2 1 3 1 4 3 3 2 =32 bits
M = 0100 1011 0100 0001 0100 1110 0100 1110 0100 0001 0100 1110 0100 0010 0100 0001
C_2= 1101 0011 1010 1100 1011 0101 1010 1011 1000 1010 1011 0111 0010 0011 1100 1101
$H(M \oplus C_2)$= 2 1 3 3 4 3 3 2 2 3 4 2 2 1 1 2 =38 bits

7. Conclusion

PPT based keystreams have been thought of in this paper and they are used in generating the subkey for each round of DES. The key generation of modified proposed DES differs in the conventional DES key generation in the sense that in the conventional DES the same 64-bit key value which was initially accepted as input is used for generating the subkey in all round. It is noticed that in each round 64-bits key is taken from the class label assigned to PPT by considering the stating position randomly as determined by both sender and receiver. It provides an additional level of security. Further, to generate the next round key an another eight characters are taken from the stream of class labels but the starting position is determined randomly. In the proposed modified DES, the Hamming distance between plaintext and cipher text is increasing when it is compared with classical DES wherein which the same 64-bits key is always being used in generating the subkey for each round. As the Hamming distance between plaintext and ciphertext is increasing, the adversary may not easily recover the plaintext from the ciphertext. The idea used is unique, simple and innovative and it enhances the security.

References

[1] DeepthiSathidevi, “Hardware Stream Cipher Based on LFSR and Modular Division Circuit”, World Academy of Science Press, vol.2 (10), 2008.

[2] Win Kyaw, “Speech Encryption and Decryption Using LFSR”, World Academy of Science, Engineering and Technology Journal, 2008.

[3] Majid Bakhtiari, Mohd Aizaini Maarof , “An Efficient Stream Cipher Algorithm for Data Encryption”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011 ISSN (Online): 1694-0814 www.IJCSI.org.

[4] Meier, W. and O. Staffelbach, “Nonlinearity Criteria for Cryptographic Functions, Advances in Cryptology”, EUROCRYPT ’89, J-J. Quisquater and J. Vandewalle, Editors. 1990, Springer Berlin /Heidelberg, pp: 549-562.

[5] Charles Pfleeger, Shari Lawrence Pfleeger, “Security in computing”, Fourth Edition 2007, Prentice Hall of India Pvt Ltd, New Delhi.

[6] Scott Fluhrer, ItsikMantin and Adi Shamir, “weaknesses in the Key Scheduling Algorithm of RC4”, (1).Cisco Systems, Inc, 170 West Tasman Drive, SanJose, CA95134.(2).Computer Science department, The WeizemannInstitute, Rehovot 76100, Israel.

[7] O Neugebauer and A sachs, “Mathematical Cuneiform Texts”, New Haven, CT., 1945.

[8] Biham E. and Seberry, “Py (Roo): A Fast and Secure Stream Cipher”, EUROCRYPT'05 Rump Session, at the Symmetric Key Encryption Workshop (SKEW2005), May 2005, pp: 26-27.

[9] HongGeun Kim, JungKyu Han and SeongjeCho, “An efficient implementation of RC4 cipher for

encrypting multimedia files on mobile devices", SAC '07 Proceedings of the ACM symposium on Applied computing, 2007, pp: 1171--1175, NewYork, USA.

[10] Sreelaja.N.K and G.A.VijayalakshmiPai, " Swarm Intelligence based key generation for Text encryption in Cellular Networks", IEEE Proceedings of the Third International Conference on System Software and Middleware and Workshops, 2008, COMSWARE 2008, 6-10 Jan. 2008, pp: 622 – 629.

[11] Minaam, D.S.A.Abdual-Kader, H.M. & Hadhoud M. M. (2010), "Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types", IJ Network Security, Vol.11 (2), 2010.

[12] T.Muthumanickam, "Performance Analysis of Cryptographic VLSI Data", IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250- 3501 Vol. 2, No. 1, 2012.

[13] Mijanur Rahaman, Md. Masudul Islam, "An Overview on Quantum Computing as a Service (QCaaS): Probability or Possibility", International Journal of Mathematical Sciences and Computing (IJMSC), Vol.2, No.1, pp.16-22, 2016.

[14] Erdem.S.S., Yanik, T., Ko¸c, C¸.K., "Fast Finite Field Multiplication. In: C¸.K. Ko¸c(ed.) Cryptographic Engineering", Chapter 5. Springer (2009).

[15] Longa, P., Miri, A., "New Composite Operations and Precomputation Scheme for Elliptic Curve Cryptosystems over Prime Fields. In: PKC 2008", LNCS, vol. 4939, pp. 229-247, Springer, Heidelberg (2008).

Authors' Profiles

Mani. K received his MCA and M.Tech. from the Bharathidasan University, Trichy, India in Computer Applications and Advanced Information Technology respectively. Since 1989, he has been with the Department of Computer Science at the Nehru Memorial College, affiliated to Bharathidasan University where he is currently working as an Associate Professor. He completed his PhD in Cryptography with primary emphasis on evolution of framework for enhancing the security and optimizing the run time in cryptographic algorithms. He published and presented around 15 research papers at international journals and conferences.

Devi. A received her MCA and M.Phil. from Bharathidasan University, Trichy, India in Computer Science Applications. During 2004-2016(April), she had been with the Department of Computer Science at the Lowry Memorial College, affiliated to Bangalore university, Karnataka, India where she was working as an Associate Professor. During 1998-2001, She was working as a programmer in different software companies. She is currently working as a Professor in Cavalier Animation and Media Science, affiliated to Mysore University, Karnataka, India. She is pursuing her PhD in Compressed Cryptosystem, Bharathidasan University, Trichy, India.

How to cite this paper: K.Mani, A.Devi,"Modified DES using Different Keystreams Based On Primitive Pythagorean Triples", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.3, No.1, pp.38-48, 2017.DOI: 10.5815/ijmsc.2017.01.04