Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improved Cryptosystem Using SDES Algorithm with Substitution Ciphers

G.Suman * , Ch. Krishna
M. Tech (SE) & SNIST.
India

Abstract— Security is playing a very important and crucial role in the field of network communication system and Internet. Simplified Data encryption standard (SDES) is a private key cryptography system that provides the security in communication system but now a days the advancement in the computational power the SDES seems to be weak against the brute force attacks. To improve the security of SDES algorithm the substitution techniques are added before the SDES algorithm to perform its process. By using an Improved Cryptosystem the security has been improved which is very crucial in the communication and field of Internet. If the substitution techniques are used before the original SDES algorithm then the intruder required first to break the original SDES algorithm and then substitution techniques. So the security is more as compared to a SDES algorithm.

Keywords— Cipher text, Decryption, SDES, Encryption, Plain text, OPBFT,OPBTS ,Substitution.

I. Introduction

The process of encoding the plaintext into cipher text is called Encryption and the process of decoding ciphers text to plaintext is called Decryption. This can be done by two techniques symmetric-key cryptography and asymmetric key cryptography. Symmetric key cryptography involves the usage of the same key for encryption and decryption. But the Asymmetric key cryptography involves the usage of one key for encryption and another different key for decryption. Secret key cryptography includes DES, AES, 3DES, IDEA, Blowfish algorithms etc. and public key cryptography includes RSA, Digital Signature and Message Digest algorithms . For each algorithm there are two key aspects used: Algorithm type (define size of plain text should be encrypted per step) and algorithm mode (define cryptographic Algorithm mode). Algorithm mode is a combination of a series of the basic algorithm and some block cipher and some feedback from previous steps.

II. Simplified Data Encryption Standard

a. S-DES Key Generation:

S-DES depends on the use of a 10-bit key shared between sender and receiver. From this key,two 8-bit subkeys are produced for use in particular stages of the encryption and decryption algorithm.

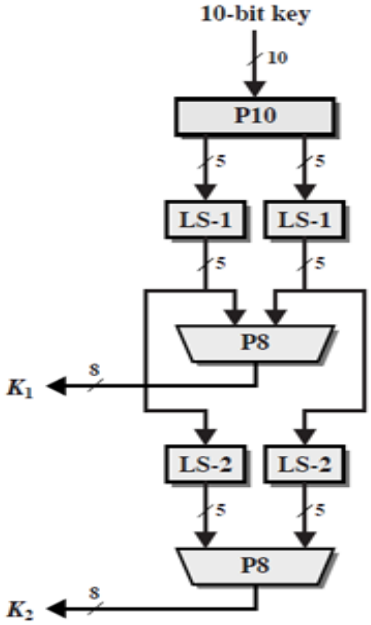

Figure1: Key Generation for Simplified DES

First, permute the key in the following fashion. Let the 10-bit key be designated as (*k*1, *k*2,*k*3, *k*4, *k*5, *k*6, *k*7, *k*8, *k*9,*k*10). Then the permutation P10 is defined as:

P10 (*k*1, *k*2, *k*3, *k*4, *k*5, *k*6, *k*7, *k*8, *k*9, *k*10) = (*k*3, *k*5, *k*2, *k*7, *k*4, *k*10, *k*1, *k*9, *k*8, *k*6)

P10 can be concisely defined by the display:

P10 (3 5 2 7 4 10 1 9 8 6).

This P10 (10-bit key) is read from left to right; each position in it gives the identity of the input bit that produces the output bit in that position. So the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on. For example, the key (1010000010) is permuted to (1000001100). Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits. In our example, the result is (00001 11000). Next we apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:

P8 (6 3 7 4 8 5 10 9)

The result is subkey 1 (*K*1). In our example, this yields (10100100).We then go back to the pair of 5-bit strings produced by the two LS-1 functions and perform a circular left shift of 2 bit positions on each string. In our example, the value (0000111000) becomes (00100 00011). Finally, P8 is applied again to produce *K*2. In our example, the result is (01000011).

b. Initial and Final Permutations:

The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function: IP (2 6 3 1 4 8 5 7).This retains all 8 bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is used:IP^{-1} (4 1 3 5 7 2 8 6).It is easy to show by example that the second permutation is indeed the reverse of the first;that is, $IP^{-1}(IP(X)) = X$.

***c.* The Function f*K*:**

The most complex component of S-DES is the function f*K*, which consists of a combination of permutation and substitution functions. The functions can be expressed as follows. Let *L* and *R* be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to f*K*, and let F be a mapping (not necessarily one to one) from 4-bit strings to 4-bit strings. Then we let f*K*(*L*, *R*) = (*L (X-OR)* F(*R*, *SK*), *R*) where *SK* is a subkey and X-OR is the bit-by-bit exclusive-OR function. For example, suppose the output of the IP stage in Figure3 is (10111101) and F(1101, *SK*) = (1110) for some key *SK*. Then f*K*(10111101) = (01011101) because (1011) X-OR (1110) = (0101). We now describe the mapping F. The input is a 4-bit number . The first operation is an expansion/permutation operation:E/P (4 1 2 3 2 3 4 1). The first 4 bits (first row of the preceding matrix) are fed into the S-box S0 to produce a 2-bit output, and the remaining 4 bits (second row) are fed into S1 to produce another 2-bit output.These two boxes are defined as follows:

S0=

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

S1=

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

The S-boxes operate as follows. The first and fourth input bits are treated as a 2-bit number that specify a row of the S-box, and the second and third input bits specify a column of the Sbox. The entry in that row and column, in base 2, is the 2-bit output. For example, if (*s*0.1,*s0.4*) =(00) and (*s0.2*,*s0.3*) = (10), then the output is from row 0, column 2 of S0, which is 3, or (11) in binary. Similarly, (*s1.1*,*s1.4*) and (*s1.2*,*s1.3*) are used to index into a row and column of S1 to produce an additional 2 bits. Next, the 4 bits produced by S0 and S1 undergo a further permutation as follows:

P4 (2 4 3 1)

The output of P4 is the output of the function F.

d. The Switch Function:

The function f*K* only alters the leftmost 4 bits of the input. The switch function (SW) interchanges the left and right 4 bits so that the second instance of f*K* operates on a different 4 bits. In this second instance, the E/P, S0, S1, and P4 functions are the same. The key input is K2. A brute-force attack on simplified DES is certainly feasible. With a 10-bit key, there are only 2^{10} = 1024 possibilities. Given a ciphertext, an attacker can try each possibility and analyze the result to determine if it is reasonable plaintext.

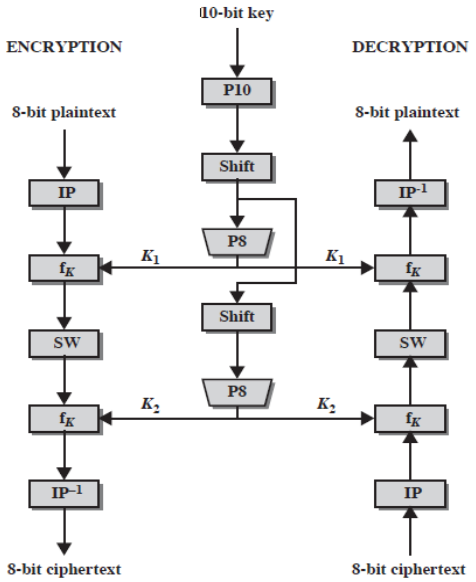

Figure2: Simplified DES Scheme

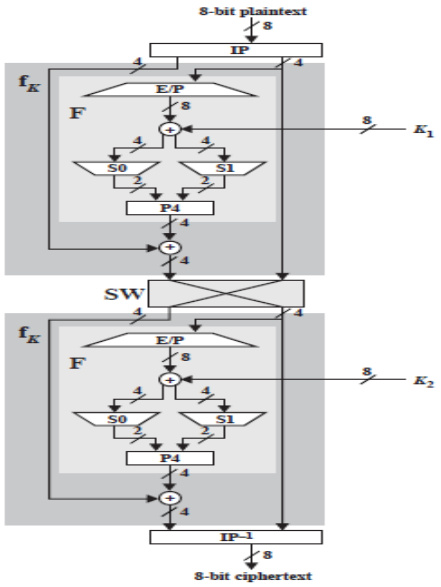

Figure3: Simplified DES Encryption Detail

III. **Four-Square Cipher (Fsc)**

The four-square cipher encrypts pairs of letters (digraphs) and is thus less susceptible to frequency analysis attacks.
The four-square cipher uses four 5 by 5 matrices arranged in a square. Each of the 5 by 5 matrices contains the letters of the alphabet (usually omitting "Q" or putting both "I" and "J" in the same location to reduce the alphabet to fit). In general, the upper-left and lower-right matrices are the "plaintext squares" and each contain a standard alphabet. The upper-right and lower-left squares are the "ciphertext squares" and contain a mixed alphabetic sequence. To generate the ciphertext squares, one would first fill in the spaces in the matrix with the letters of a keyword or phrase (dropping any duplicate letters), then fill the remaining spaces with the rest of the letters of the alphabet in order. The four-square algorithm allows for two separate keys, one for each of the two ciphertext matrices. In the example to the right, "EXAMPLE" and "KEYWORD" have been used as keywords.
To encrypt a message you would first split the message into digraphs. "This is a secret message" would become:
TH IS IS AS EC RE TM ES SA GE
a b c d e E X A M P
f g h i j L B C D F
k l m n o G H I J K
p r s t u N O R S T
v w x y z U V W Y Z

K E Y W O a b c d e
R D A B C f g h i j
F G H I J k l m n o
L M N P S p r s t u
T U V X Z v w x y z
Once that was complete, you would take the first pair of letters and find the first letter in the upper left square and the second letter in the lower right square. In this example we are enciphering TH, so we locate T and H in the grid below (see blue characters). Now, we find the intersections of the rows and columns of the plain text letters. In this example, they have been highlighted in red (R and B). These new letters are the enciphered digraph (RB).

You continue enciphering digraphs in this way until you reach the end of the message. To continue our example, "This is a secret message" would be enciphered as:
RB CP CP AL AO TE RI AS NY FE

IV.**Trifid Cipher (Tc)**

In this we use a 3x3x3 cube.

Layer 2

	1	2	3
1	J	K	L
2	M	N	O
3	P	Q	R

Layer 1

	1	2	3
1	A	B	C
2	D	E	F
3	G	H	I

Layer 3

	1	2	3
1	S	T	U
2	V	W	X
3	Y	Z	

The first step is to use the cube to convert the letters into numbers. We will be writing the numbers vertically below the message in the order of Layer, Column, Row.
Example:
secret message
311213 2133111
123322 1211112
121321 2211132

The numbers are now read off horizontally and grouped into triplets.
311 213 213 311 112 332 212 111 121 213 212 211 132

The cube is used again to convert the numbers back into letters which gives us our ciphertext. Sppsdxmabpmjf To decipher a Trifid encrypted message, you first convert each letter into its corresponding number via the cube. Now, divide the long string of numbers into three equal rows. Now, read off each column and use the cube to convert the three numbers into the plaintext letter.

V.**The Proposed Work**

The proposed system takes plain text as input. This plain text is fed into Four-Square Cipher(FSC) which converts the plaint text into OPBFT(**Output Between Four-Square Cipher and Trifid Cipher** in encrypted form).The OPBFT is given as input to Trifid Cipher that inturn converts OPBFT into OPBTS(**Output Between Trifid Cipher and SDES** in encrypted form).Atlast OPBTS is given as input **(**in which each letter consider as 8bit binary number for eg: letter **J** that is equal to **01001010 converted into 00010111)** to SDES Encryption algorithm which converts it into cipher text.In Decryption we convert cipher text into plain text(figure5)
Example:
Plain Text: THIS IS A SECRET MESSAGE (input to FSC)
$K_{10\ bit}$ = 1010000010 (for SDES algorithm)

then,
OPBFT: RBCPCPALAOTERIASNYFE

OPBTS: JMBLJZGLSYQSPLCSBRDW

Cipher Text: 00010111 11101010 00011001 10100101 00010111 10110100 00000001 10100101 01001111 10101010 10000100 01001111 10100011 10100101 01000100 01001111 10011111 01011111 00001010

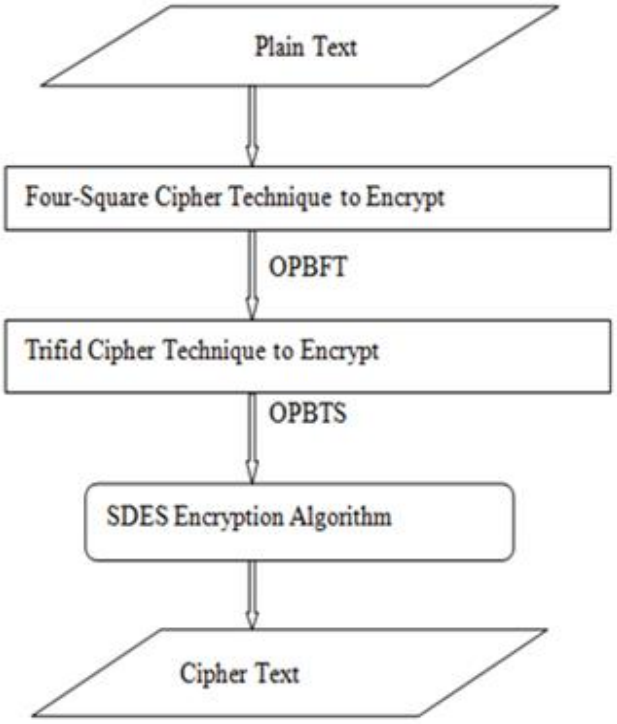

Figure4: Improved Cryptosystem Encryption Process

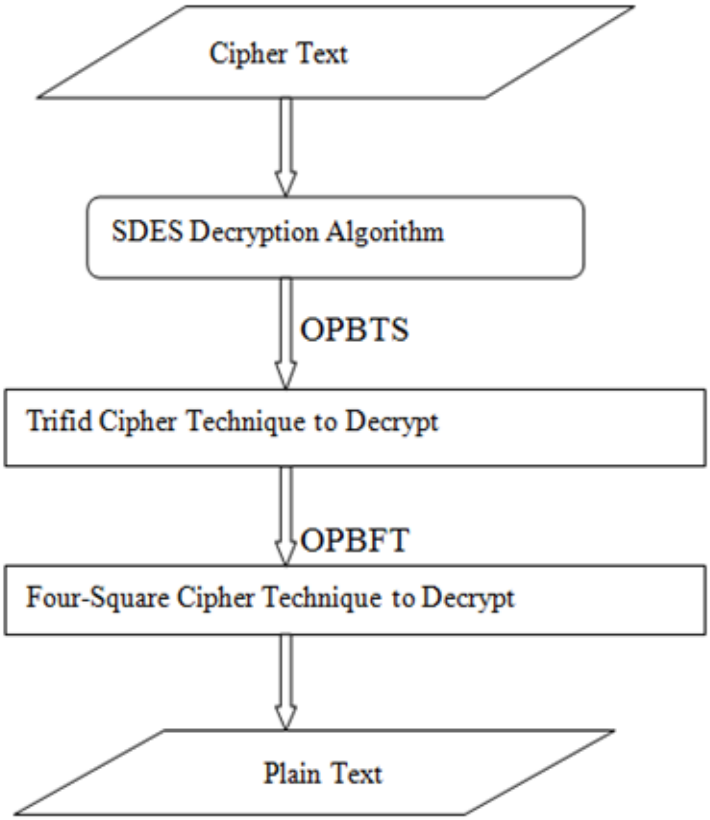

Figure5: Improved Cryptosystem Decryption Process

VI. **Properties of The Improved Cryptosystem**

The Improved Cryptosystem has the following advantages over SDES:

a. The security of the algorithm is increased. The Four-Square Cipher and Trifid Cipher substitution Techniques are used before SDES.

b. The Brute Force attack is weak against the Improved Cryptosystem because the intruder required to break the SDES , Four-Square Cipher and Trifid Cipher . He required extra time to hack the Improved Cryptosystem.

c. If the intruder is success to hack the key of SDES in any way then he required crack the Four-Square, Trifid Cipher approaches to reach the plain text The Improved Cryptosystem has the following disadvantage over SDES:

a. The main disadvantage of using this approach is the extra computation is required to perform the operation. But this is not so crucial because our main aim is to provide tighter security.

VII. **Conclusions**

In todays time,the security is playing a very important and powerful role in the field of networking, Internet and various communication system .The electronic communication system is used in banking, reservation system and marketing which required a very tight security system. The original SDES implementation has some weaknesses, to overcome the most of weakness the Improved Cryptosystem is designed. The Designed system improved the security power of original SDES. The only drawback of Improved Cryptosystem is extra computation is needed but the todays computer have parallel and high speed computation power so the drawback of the Improved Cryptosystem is neglected because our main aim is to enhance the security of a system. By using the Improved Cryptosystem the security is very tight and approximately impossible to crack and break the Improved Cryptosystem.

Acknowledgment

We would like to give our sincere gratitude to our guide Ms.G.Ravi (*Asstt. Prof. in CSE Dept)* who encouraged and guided us throughout this paper.

References

[1] M. E. Hellman,"DES will be totally insecure within ten years" lEEE Spectrum, Vo1.16, N0.7, pp32-39, July 1979.

[2] Manikandan. G, Rajendiran.P, Chakarapani.K, Krishnan.G, Sundarganesh.G,"A Modified Crypto Scheme for Enhancing Data Security", Journal of Theoretical and Advanced Information Technology, Jan 2012.

[3] Shah Kruti R., Bhavika Gambhava,"New Approach of Data Encryption Standard Algorithm", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-1, March 2012.
[4] William Stallings, " Cryptography and Network Security Principles and Practices", Prentice Hall, November 16, 2005.
[5] A. Nadeem, "A performance comparison of data encryption algorithms", IEEE information and communication technologies, pp.84-89, 2006.Bn
[6] Atul Kahte, "Cryptography and Network Security",Tata Mcgraw Hill, 2007.
[7] Shasi Mehlrotra seth, Rajan Mishra,"Comparative Analysis of Encryption Algorithms For Data Communication", IJCST Vol. 2, Issue 2, June 2011.
[8] Charels Connell, "An Analysis of New DES: A Modified Version of DES", Locust Street Burlington, USA, Boston MA 02215 USA.
[9] D. Coppersmith, "The Data Encryption Standard (DES) and Its strength Against attacks", IBM J. RES. Develop. VOL.38 NO.3 MAY 1994.
[10] Gaurav Shrivastava, "Analysis Improved Cryptosystem Using DES with RSA" VSRD-IJCSIT, Vol. 1 (7), 465-470, 2011.
[11] Gurdev Singh, Jimmy Singla and Shivdev Singh, "Message Encryption and Decryption" VSRD-IJCSIT, Vol. 2 (7), 2012, 668- 671.

