Effective Comparison of Enhanced DES Algorithm

[1]Payal Patel, [2]Khushbu Shah
[1]M.E. Computer Engineering, [2]Assistant Professor
L.J. Institute of Engineering & Technology, Ahmedabad, India
[1]Payal.5886@gmail.com, [2]khushburana1@gmail.com

***Abstract*— The principal goal to design any encryption algorithm must be secure against unauthorized attacks. Data Encryption Standard algorithm is a symmetric key algorithm and it is used to secure the data. DES works on 64 bit data and 56 bit key. Different enhancements of DES algorithms are available. From the enhanced algorithm, few algorithm works on increasing the key length and few has complex S-BOX design, while other has increased the number of states in which the information is to be represented. In DES-96 improved DES security algorithm has used 84-bit key instead of the original 56-bit key, to resist brute-force attack. This would give $2^{84} \approx 1.934*10^{25}$ trials instead of $2^{56} \approx 7.205* 10^{16.}$ By increasing the key length, the number of combinations for key is increases which is hard for the intruder to do the brute force attack. As the S-BOX design will become the complex, there will be a good avalanche effect. As the number of states increases in which the information is represented instead of binary representation, it is hard for the intruder to crack the actual information. Block encryption standard for transfer of data algorithms have minimized the memory requirements and execution time complexity. The total number of combinations required to decipher a 4 byte text is: $2^{32} * 2^{10} * 2^{24} * 2^{6} = 2^{72}$ units.**

Key Words – Cryptography, Encryption, Xor, DES

I. INTRODUCTION

Cryptography is the study of information hiding and verification. It includes the protocols, algorithms and strategies to securely and consistently prevent or delay unauthorized access to sensitive information and enable verifiability of every component in a communication. Cryptography is derived from the Greek words: kryptós, "hidden", and gráphein, "to write" - or "hidden writing". People who study and develop cryptography are called cryptographers. The study of how to circumvent the use of cryptography for unintended recipients is called cryptanalysis, or code breaking. An example of the sub-fields of cryptography is steganography — the study of hiding the very existence of a message, and not necessarily the contents of the message itself (for example, microdots, or invisible ink).

When information is transformed from a useful form of understanding to an opaque form of understanding, this is called encryption. When the information is reverted back into a useful form, it is called decryption. Intended recipients or authorized use of the information is determined by whether the user has a certain piece of secret knowledge. Only users with the secret knowledge can transform the opaque information back into its useful form. The secret knowledge is called the key, though the secret knowledge may include the entire process or algorithm that is used in the encryption/decryption. The information in its useful form is called plaintext (or cleartext); in its encrypted form it is called ciphertext. The algorithm used for encryption and decryption is called a cipher (or cypher).

A. Cryptography Goals

1. *Confidentiality:* Information in computer transmitted information is accessible only for reading by authorized parties.[6]
2. *Authentication:* Origin of message is correctly identified with an assurance that identity is not false.[6]
3. *Integrity:* Only authorized parties are able to modify transmitted or stored information.[6]
4. *Non Repudiation:* Requires that neither the sender, nor the receiver of message be able to deny the transmission.[6]
5. *Access Control:* Requires access may be controlled by or for the target system. [6]
6. *Availability:* Computer system assets are available to authorized parties when needed. [6]

B. Types of Algorithm

1) Symmetric key encryption Algorithm

In Symmetric key encryption algorithm there is only single secret key and that is used for encryption as well as for decryption.
There are two types of symmetric encryption algorithms:

a. Stream cipher: which perform bit by bit encryption.
b. Block cipher: which perform block of data.

2) Asymmetric key encryption Algorithm

In Asymmetric key encryption algorithm there are two different key. One Private key and another is Public key. The data can be encrypted by sender's private key and it is decrypted by receiver's public key. Using this mechanism authentication is achieved.

If data is encrypted by sender's public key and decrypted by receiver's private key then confidentiality is achieved.

II. DATA ENCRYPTION STANDARD ALGORITHM

Data encryption standard is a symmetric key algorithm. *DES* takes on input a 64-bit plaintext data block and 56-bit key (with 8 bits of parity) and outputs a 64-bit cipher text block.

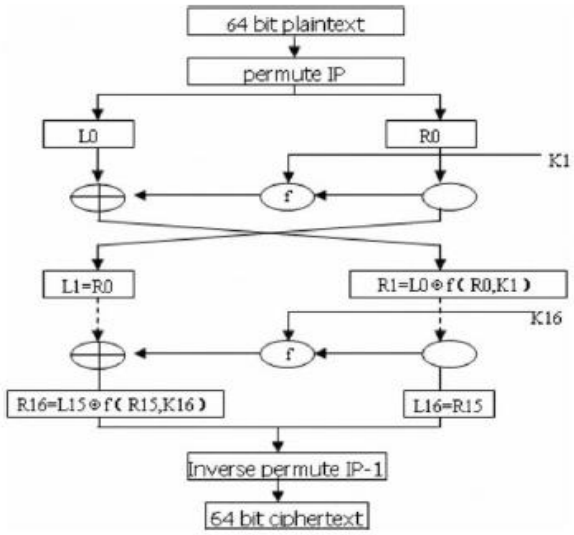

Fig. 1. DES Algorithm[1]

A. *Steps*

1) The plaintext block is subject to an Initial Permutation to shift the bits around.
2) The 8 parity bits are removed from the key by subjecting the key to its Key Permutation.
3) The plaintext and key are processed in 16 rounds consisting of:
   h) The key is split into two 28-bit halves.
   i) Each half of the key is shifted (rotated) by one or two bits, depending on the round.
   j) The halves are recombined and subject to a Compression Permutation to reduce the key from 56 bits to 48 bits. This Compressed Key is used to encrypt this round's plaintext block.
   k) The rotated key halves from step 2 are used in next round.
   l) The data block is split into two 32-bit halves.
   m) One half is subject to an Expansion Permutation to increase its size to 48 bits.
   n) Output of step 6 is exclusive-OR'ed with the 48-bit compressed key from step 3.
   o) Output of step 7 is fed into an S-box, which substitutes key bits and reduces the 48-bit block back down to 32-bits.
   p) Output of step 8 is subject to a P-box to permute (scramble) the bits.
   q) The output from the P-box is exclusive-OR'ed with the other half of the data block.
   r) The two data halves are swapped and become the next round's input.
4) After 16 rounds, the resultant ciphertext is subject to a Reverse Initial Permutation. The output is the ciphertext block.

B. *Possible Aattacks against DES*

- Brute force is a known-plaintext attack and requires testing, on average, 2^{55} keys.
- Differential cryptanalysis is a chosen plaintext attack where the attacker encrypts two chosen plaintext blocks and uses the differences between the ciphertext to deduce the key. This attack requires 2^{43}plaintext/ciphertext pairs and $2^{55.1}$ encryption operations, making it less efficient than a brute force attack. Apparently DES was designed to be resistant to differential cryptanalysis.
- Linear cryptanalysis is a more recent development; DES was not specifically designed to resist this attack. However, linear cryptanalysis of DES has not been fully developed.

III. MODIFIED DES ALGORITHM

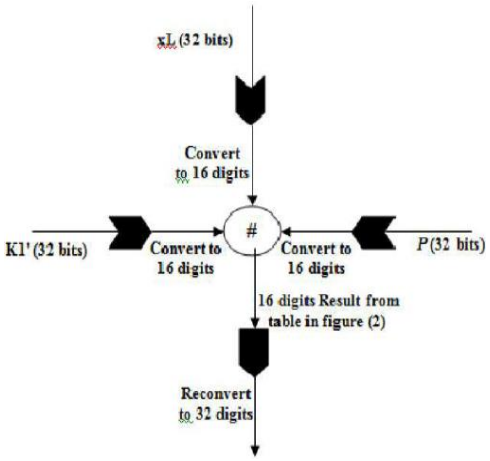

Fig. 2. Design of Modified DES Algorithm[1]

A. *Steps:*

Input: plaintext m1 . . . m64; 64-bit two keys K=k1 . . . k64 and K'=k1' . . . k64' (includes 8 parity bits).
Output: 64-bit ciphertext block C=c1 . . .c64.

1) (key schedule) Compute sixteen 48-bit round keys Ki, from K.
   a. (key schedule) compute sixteen 32-bit round keys Ki', from K'
2) (L0, R0) ← IP (m1, m2,. . .m64) (Use IP Table to permute bits; split the result into left and right 32-bit halves L0=m58m50 . . . m8,R0=m57m49 . . . m7)
3) (16 rounds) for i from 1 to 16, compute Li and Ri as follows:
   a) Li=Ri-1
   b) Ri = Li-1 # *f* (R i-1, Ki)
   c) where *f*(Ri-1, Ki) = P(S(E(Ri-1) Å Ki)), computed as follows:
   d) Expand Ri-1 = r1r2 . . . r32 from 32 to 48 bits. T ← E(Ri-1). (Thus T= r32r1r2 . . . r32r1.)
   e) T' ← T XOR Ki . Represent T ' as eight 6-bit character strings: T ' = (B1 . . . B8)
   f) T " ← F where Function F = (((((((S1+S2) mod 2 ^ 32) XOR S3) + S4) mod 2^32) XOR S5) +S6)mod 2^32 Here, Si(Bi) maps to the 8 bit entry in row r and column c of Si
   g) T'" ← P(T"). (Use P per table to permute the 32 bits of T"=t1t2 .. ... …. t32,yielding t16t7 . . . t25.) and the operation # in Ri = Li-1 # *f* (R i-1, Ki) is computed as follows:

      I. Convert the 32 bits resulted from *f* (R i-1, Ki) into 4-states 16 digits call it ***f*** '
      II. Convert the 32 bits of Li-1 to 4-states 16 digits call it Li-1'
      III. Convert the 32 bits of Ki' to 4-states 16 digits call it Ki"
      IV. Compute Ri by applying the # operation on *f* ', Li-1', and Ki" according to truth tables shown in Table.

4) b1b2 . . . b64 ← (R16, L16). (Exchange final blocks L16, R16.)
5) C ← IP^{-1} (b1b2 . . . b64). (Transpose using IP^{-1} C = b40b8 . . . b25.)
6) End.

TABLE I. TRUTH TABLE[1]

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

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

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

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

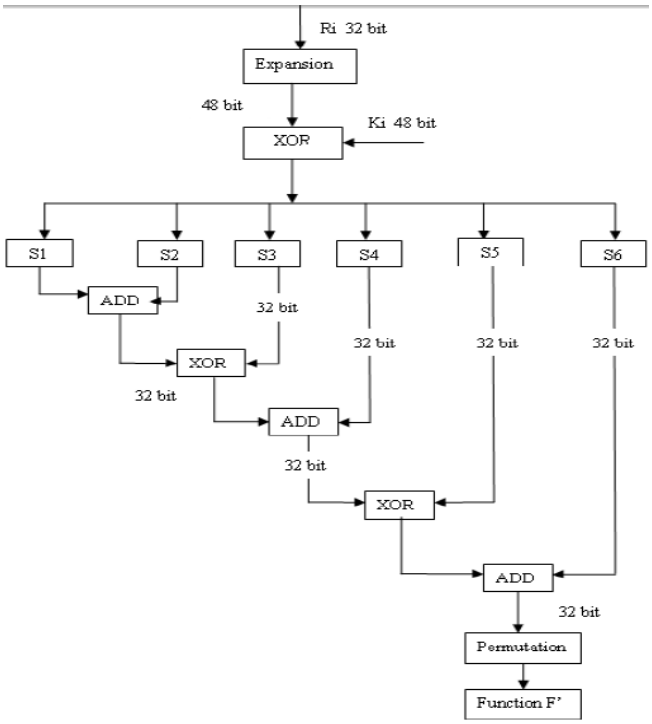

Fig. 3. Function "F" Design[1]

A new method to enhance the performance of the Data Encryption Standard (DES) algorithm. This is done by replacing the predefined XOR operation applied during the 16 round of the standard algorithm by a new operation depends on using two keys, each key consists of a combination of 4 states (0, 1, 2, 3) instead of the ordinary 2 state key (0, 1). This replacement adds a new level of protection strength and more robustness against breaking methods.

IV. MODIFIED DATA ENCRYPTION STANDARD WITH ADDITION MODULO OPERATION[5]

A. *Steps:*

Input: plaintext p1 . . . p64; 64-bit key K=k1 . . . k64 (includes 8 parity bits).
Output: 64-bit cipher text block C=c1 . . .c64.

1) (key schedule) Compute sixteen 48-bit round keys Ki, from K.
2) (L0, R0) ←IP (p1, p2,. . .p64) (Use IP Table to permute bits; split the result into left and right 32-bit halves L0=p58p50 . . . p8,R0=p57p49 . . . p7)
3) (16 rounds) for i from 1 to 16, compute Li and Ri as follows:
   a) Li=Ri-1
   b) Ri = Li-1 additionmodulo2^{32} *f* (R i-1, Ki)
   Where *f*(Ri-1, Ki) = P(S(E(Ri-1) XOR Ki)), computed as follows:
   i. Expand Ri-1 = r32r1r2 . . . r32 r1 from 32 to 48 bits,
   M←E(Ri-1).
   ii. M' ←M XOR Ki. Represent M ' as eight 6-bit character strings:
   M'= (B1 . . . B8)
   iii. M " ← F' where function F'= ((((s1Λs2) XOR s3) Λ s4)XOR s5) Λ s6). Here si(Bi) maps to the 8/32 S-Box that consist of 256 entries.
   iv. M"' ←P (M"). (Use P per table to permute the 32 bits of
   M"=m1m2 . . . m32, yielding m16m7 . . . m25)
4) b1b2 . . . b64 ← (R16, L16). (Exchange final blocks L16,R16.)
5) C ←IP^{-1} (b1b2 . . . b64).
6) End.

A new improvement to the DES algorithm which makes the use of the new operation known as addition modulo (+). It takes two inputs and performs Addition and resulting output assume like x. later perform x mod 2w Where w is the number of bits that depends on given input.

Example: x and y are the Inputs
X=1100 1000
Y=1000 1111

X1 is obtained by performing x+y
X1= 1 0101 0111
Carry can be thrown off (or) perform modulo 2^8
X1 is converted to decimal number
X1= 343 mod 2^8 = 87
Binary equivalent of x1 is 0101 0111
To find original x value perform following operation
X=x1+(-y)
To obtain (-y) = 2^8-y =>256-143=113
Perform X1+ (-y) which results
Original x
0111 0001
0101 0111
1100 1000 original x value

By adding modified S-Box design, modifies function implementation and replacing the old XOR by a new operation give more robustness to DES algorithm and make it stronger against any kind of intruding. This new algorithm gives avalanche effect than the original DES algorithm.

V. BLOCK ENCRYPTION STANDARD FOR TRANSFER OF DATA [3]

In this algorithm, there are two predefined stacks along with a logic based lookup concept. The first stack holds some specially chosen symbols, where other stack contains a random number from a preselected range by a predefined method to make the code sequence more secure. The encryption process does a variety of binary operations like Shift Left Operation on the message for protecting it against unauthorized attacks.

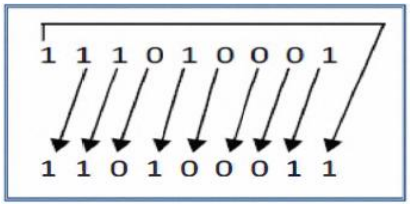

Fig. 4. Left Shift Operation[3]

Bits are shifted left to one place and the Most Significant Bit (MSB) is placed to Least Significant Bit (LSB).

1) The plain text in the block size of 32 bits is read from input file.
2) The plaintext is transformed into ASCII code and then modified into binary form.
3) Then shift-left operation is performed on this 32-bit data 10 times.
4) The modified plain text is then X-ORed with a secondary key of 32 bits and it is made sure the result is also of 32 bits.
5) A random number is chosen from a given range and converted into 16-bit binary number.
6) A sequence symbol is randomly selected from a preselected range.
7) The selected symbol is converted into ASCII code and then finally into binary number of 8 bits.
8) The 8-bit binary code is then appended to the 16-bit binary number resulted from random number and the result is stored as the Base Key or Primary Key.
9) Then the key is applied on the modified plaintext with the help of a binary operation.
10) In the next step, a new key is generated from a different random number and different sequence symbol.
11) Every time a new key is generated, it is applied using a different binary operation on resulted cipher text of previous step and a modified cipher text is obtained.
12) This process is repeated 10 times i.e. ten times a different key is produced and ten times this key is applied on the plaintext or cipher text of previous round.
13) The encryption process is continued for next characters of file until end of file is reached.

BEST algorithm keeps changing the key based on randomly selected integer number and sequence symbol. This feature makes BEST algorithm immune to the "Replay attacks", making it more safe and sound.

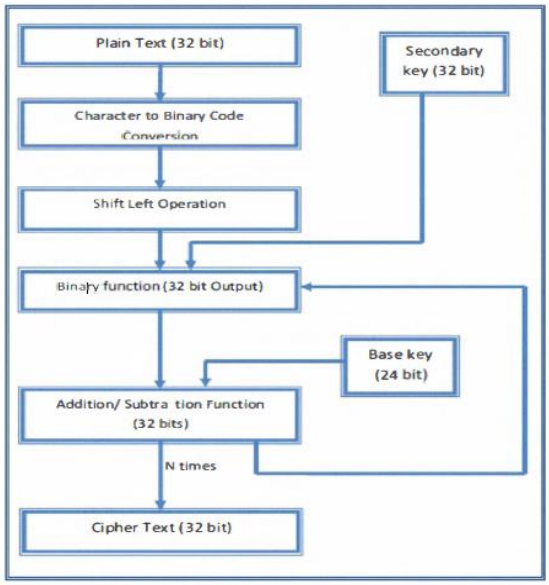

Fig. 5. Block Diagram of BEST Encryption Algorithm[3]

A. Timing Analysis

The core advantage of this cryptographic algorithm is the speed of encoding and decoding of data.

Encryption Algorithms	Input Size (Bytes)	Encryption Time (sec)	Decryption Time (sec)	Total Execution Time (sec)
DES	45911	5	41	46
AES	45911	8	125	133
X-MODES	45911	257	2	259
BEST	45911	1	0.8	1.8

TABLE II. TIMING ANALYSIS OF BEST [3]

B. Memory Requirements

Memory required by the BEST is half as compare to DES encryption algorithm and one-fourth of AES encryption algorithm.

Encryption Algorithms	Key Length (Bits)	Plain Text Size (Bits)	Cipher Text (Bits)
DES	56	64	64
AES	128	128	128
X-MODES	32	32	32
BEST	24	32	32

TABLE III. MEMORY ANALYSIS OF BEST [3]

- Possible number of attempts to break the Secondary Key: 2^{32}
- Probable number of attempts to break the Primary Key: $2^{16} * 2^{8} = 2^{24}$
- There are total 10 cycles like this and each contains the 2 different functionalities. So multiply it by 2^{10} also. So Total effort for the primary key = $2^{10}*2^{24}$.
- Potential number of attempts to reverse the Shift left Operation;. As the data is 32 bits long and shift left operation is performed on this data, hence there is total 2 * 32 (2^{6}) cases possible to break the code.

The total number of combinations required to decipher a 4 byte text is: $2^{32} * 2^{10} * 2^{24} * 2^{6} = 2^{72}$ units.

VI. DES96 - IMPROVED DES SECURITY[2]

The proposed key generation algorithm has 96-bit key length from which only 84 bits are used after removing the parity bits. A 7-bit left shift takes place in each round. The system also has a part to indicate the arrangement of the S-Boxes of each round, a stage of S-Boxes inside the key generation algorithm itself, and more linear permutations and Permuted Choice to provide more diffusion. This is the key generation algorithm.

A. Steps:

1) The 96-bit key enters an initial permutation that discards the 12 parity bits to give an 84-bit key.
2) The 84 bits are now divided into three parts:
   a) 48 bits enters the S-Boxes to produce a 32-bit output.
   b) 28 bits enter a permuted choice to produce a 16 bit output. This permuted choice is shown in figure 3.
   c) 8 bits are processed as the following: each two adjacent bits are XORed together to produce 4 bits.

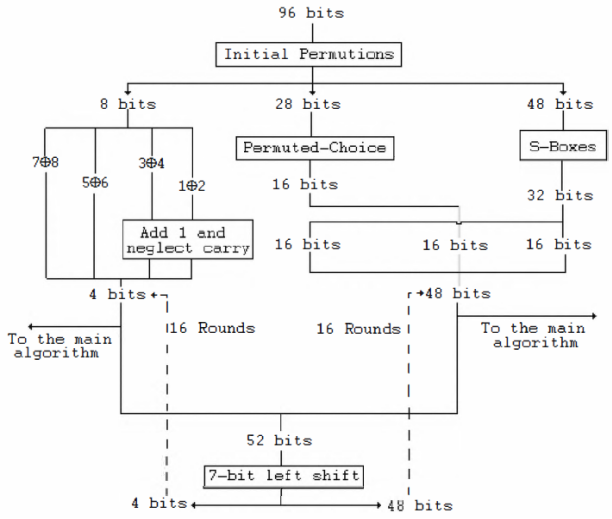

Fig. 6. Proposed Key Generation Algorithm [2]

87	57	43	47	92	25	62	2	61	66	45	4
34	92	70	30	77	22	81	5	76	20	42	55
95	75	38	41	50	85	52	79	23	67	51	21
73	19	53	89	17	9	84	44	13	74	69	27
29	90	1	31	35	14	65	33	11	28	93	6
3	83	58	12	78	7	91	37	18	15	63	26
86	54	49	71	36	59	86	60	68	39	10	46

TABLE IV. INITIAL PERMUTATION [2]

9	22	5	12	27	1	13	16
14	19	21	3	8	7	11	28

TABLE V. PERMUTED CHOICE [2]

3) The leftmost 16 bits of the 32-bit output of Step (2,a) are swapped with the 16-bit output of Step (2,b) and all these outputs are combined to produce a 48-bit block to be sent to the main algorithm as Kl (the first sub key).

4) The 4-bit output from Step (2,c)is used twice after adding 1 to the two least significant bits and discarding the carry. First, the 4 bits are sent to the main algorithm to control the arrangement of the S-Boxes. The first bit determines whether to swap boxes 2 and 3, the second bit is used to control the swapping of boxes 1 and 7, the third controls boxes 4 and 6, and the fourth controls boxes 5 and 8. The 4 bits are then recombined with the 48 bits to prepare the sub-key of the next round.
5) For the next round, a shift of 7 bits to the left takes place and the rightmost 48 bits are sent to the main algorithm and the leftmost 4 bits are dealt with as the output of Step (2,c), and so on for 16 rounds. The only change to the main algorithm was the 4 bits sent with each sub-key to determine the arrangement of the S-Boxes for each round.

B. Advantages over DES:

1) The 84-bit key instead of the original 56-bit key is aimed to resist brute-force attack. This would give $2^{84} \approx 1.934*10^{25}$ trials instead of $2^{56} \approx 7.205* 10^{16}$
2) The S-Boxes inside the key generation algorithm are aimed to reduce linearity. This is to resist linear cryptanalysis providing a non-linear operation. The non-linear operation was chosen to be the same S-Boxes of the main algorithm in order to reduce memory requirements (for software implementation) and the components needed (in hardware implementation). And it is done only once to reduce the time required for sub-key generation which is convenient for key generation.
3) Good randomness.
4) Preserving good avalanche effect.(small change in plaintext or key larger change cipher text).

VII. Comparison of Algorithm

Algorithm Name	Key Length (Bits)	No. Of Combination for decipher	Plain/ Cipher Text Length	Security
Simple DES Algorithm	Key – 56	2^{56}	64	Too Low
Modified DES Algorithm	Key 1 – 56 Key 2 – 32	$2^{56} * 2^{32}$	64	High
Modified data encryption standard with addition modulo operation	Key 1 – 56 Key 2 – 32	$2^{56} * 2^{32}$	64	High
Block encryption standard for transfer of data	Secondary Key – 32 Base Key – 24	2^{72}	64	Low
DES96- Improved DES Security	Key – 84	2^{84}	64	Moderate

TABLE VI. Algorithm comparison

VIII. CONCLUSION

Automated information resources are increased day by day and cryptography will continue to increase in importance as a security mechanism. Electronic networks for banking, shopping, inventory control, benefit and service delivery, information storage and retrieval, distributed processing, and government applications will need improved methods for access control and data security. The information security can be easily achieved by using Cryptography technique. DES is now considered to be insecure for some applications like banking system. There are also some analytical results which demonstrate theoretical weaknesses in the cipher. So it becomes very important to augment this algorithm by adding new levels of security to make it applicable. By adding additional key, modified S-Box design, and replacing the old XOR by a new operation, to give more robustness to DES algorithm and make it stronger against any kind of intruding. DES Encryption with two keys instead of one key already will increase the efficiency of cryptography. Creating the S-BOX design as complex as possible so it will create the good avalanche effect. By increasing the key length, it is hard for intruder to perform the brute force attack.

IX. References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.