Overview of Linear Cryptanalysis on S-DES and Block Ciphers using Hill Cipher Method

Rajashekarappa
Department of CSE,
JSSATE, Mauritius.
(Research Scholar, Jain University, Bangalore, India)

K M Sunjiv Soyjaudah,
PhD.
Department of Electrical & Electronic Engineering,
University of Mauritius, Reduit, Mauritius

Sumithra Devi K A, PhD.
Department of Master of Computer Applications,
R V College of Engineering,
Bangalore-59, India.

ABSTRACT
In this paper presents the Linear Cryptanalysis on S-DES and Symmetric Block Ciphers Using Hill Cipher Method. As a vehicle of demonstration of this concept, choose simple yet representative block ciphers such as computationally tractable versions of S-DES, for the studies. The attack presented in this paper is applicable to block structure independently of the key scheduling. The attack needs distinct known plaintexts which are a more realistic attack model in comparison with impossible differential cryptanalysis which uses chosen plaintext pairs. Moreover, linear cryptanalysis on simplified data encryption standard performed simulations on a small variant block and present the experimental results on the theoretical model of the multidimensional linear cryptanalysis using Hill Cipher method.

General Terms
Simplified Data Encryption Standard, Symmetric Block Cipher.

Keywords
Block Cipher, Data Encryption Standard, Linear Cryptanalysis.

1. INTRODUCTION
The prime concern with DES has been its vulnerability to brute-force attack because of its relatively short (56 bits) key length. However, there has also been interest in finding cryptanalytic attacks on DES. With the increasing popularity of block ciphers with longer key lengths, including triple DES, brute-force attacks have become increasingly impractical. Linear cryptanalysis is one of the most prominent cryptanalysis methods against block ciphers. Thus, there has been increased emphasis on cryptanalytic attacks on DES and other symmetric block ciphers. In this section provides a brief overview of the one of the most powerful and promising approach linear cryptanalysis.

Simplified DES, developed by Edward Schaefer of Santa Clara University, is an educational rather than a secure encryption algorithm. This paper considers cryptanalysis of S-DES ciphers. Though S-DES is a much simplified version of DES, Cryptanalysis of S-DES will give a better insight into the attack of DES and other block ciphers [1]. In the brute force attack, the attacker tries every possible key on the piece of cipher text until an intelligible translation of the cipher text into plaintext is obtained. Cryptographic algorithms are designed to make the brute force attack almost infeasible. Generally, the key space considered by any secret key based algorithm is large enough so that it is not possible for an attacker to try every possible key.

The rest of the paper is organized as follows: Section 2 discusses the earlier studies and works done in this area. Section 3 presents a brief overview of linear cryptanalysis and Section 4 gives the overview of block cipher design procedure. Section 5 gives a brief overview of Hill Cipher. Section 6 presents a brief overview of SDES. Experimental results are discussed in Section 7. Finally, Conclusion and Future work are presented in section 8.

2. RELATED WORK
The proposed work will require an in depth understanding of the area of cryptography and enable the development of general as well as specific algorithms for cryptanalysis [1]. Moreover, the enciphering algorithms developed in this work will find many real time applications in military, banking and other sectors where secure transmission is essential. A cipher takes a message text and some secret keying data (known as the key) as its input and produces an encrypted version of the original message, (known as the cipher text). An attack on a cipher can make use of the cipher text alone or it can make use of some plaintext and its corresponding cipher text (referred to as a known plaintext attack) (Andrew John Clark, 1998). Cryptanalysis is the process of recovering the plaintext and/or key from a cipher. Many cryptographic systems have a finite key space and, hence, are vulnerable to an exhaustive key search attack. Yet, these systems remain secure from such an attack because the size of the key space is such that the time and resources required for a search are prohibitive. A Linear Cryptanalysis Method for DES Cipher was explained by Matsui in 1993[4]. Differential cryptanalysis was not reported in the open literature until 1990. The first published effort appears to have been the cryptanalysis of a block cipher called FEAL by Murphy. This was followed by a number of papers by Biham and Shamir, who demonstrated this form of attack on a variety of encryption algorithms and hash functions; their results are summarized in this paper [2]. The most publicized results for this approach have been those that have application to DES. Differential cryptanalysis is the first published attack that is capable of breaking DES in less than 255 complexity [5]. The scheme, as reported in Biham [2], can successfully cryptanalyze DES with an effort on the order of 247 encryptions, requiring 247 chosen plaintexts. Although 247 is certainly significantly less than 255 the need for the adversary to find 247 chosen plaintexts makes this attack of only theoretical interest. In [11], has been proposed the "Heuristic Search Procedures for Cryptanalysis and Development of Enhanced Cryptographic Techniques" and demonstrated with comparative results.

Although differential cryptanalysis is a powerful tool, it does not do very well against DES. The reason, according to a

member of the IBM team that designed DES [3], is that differential cryptanalysis was known to the team as early as 1974. The need to strengthen DES against attacks using differential cryptanalysis played a large part in the design of the S-boxes and the permutation P. As evidence of the impact of these changes, consider these comparable results reported by Biham [2]. Differential cryptanalysis of an eight-round LUCIFER algorithm requires only 256 chosen plaintexts, whereas an attack on an eight-round version of DES requires 214 chosen plaintexts [6]. In [17], has been described the Comparative Cryptanalysis of Simplified-Data Encryption Standard Using Tabu Search and Simulated Annealing Methods in International Journal of Engineering Research and Development.

3. LINEAR CRYPTANALYSIS

In this paper explained with more recent development is linear cryptanalysis and referred the Matsui paper[4]. The linear cryptanalysis attack is based on finding linear approximations to describe the transformations performed in Data Encryption Standard. This method can find a Data Encryption Standard key given 243 known plaintexts, as compared to 247 chosen plaintexts for differential cryptanalysis[13]. Even this is a minor improvement, because it may be easier to acquire known plaintext rather than chosen plaintext, it still leaves linear cryptanalysis infeasible as an attack on Data Encryption Standard. In this case now it gives a brief summary of the principle on which linear cryptanalysis is based. For a cipher with n-bit plaintext and ciphertext blocks and an *m*-bit key, considering the plaintext block be labeled with P[1], ... P[*n*], the cipher text block C[1], ... C[*n*], and the key K[1], ... K[*m*]. Then define

$A[i, j, ..., k] = A[i] \oplus A[j] \oplus ... \oplus A[k]$

The objective of linear cryptanalysis is to find an effective *linear* equation of the form:

$P[\alpha 1, \alpha 2, ..., \alpha a] \oplus C[\beta 1, \beta 2, ..., \beta b] = K[\gamma 1, \gamma 2, ..., \gamma c]$

(where $x = 0$ or 1; $1 \leq a, b \leq n$, $1 \leq c \leq m$, and where the α, β and γ terms represent fixed, unique bit locations) that holds with probability $p \neq 0.5$. The further p is from 0.5, the more effective the equation. Once a proposed relation is determined, the procedure is to compute the results of the left hand side of the preceding equation for a large number of plaintext-ciphertext pairs. If the result is 0 more than half the time, assume $K[\gamma 1, \gamma 2, ..., \gamma c] = 0$. If it is 1 most of the time, assume $K[\gamma 1, \gamma 2, ..., \gamma c] = 1$. This gives us a linear equation on the key bits. Try to get more such relations so that it can solve for the key bits. Because in this paper dealing with linear equations, the problem can be approached one round of the cipher at a time, with the results combined.

4. BLOCK CIPHER DESIGN PRINCIPLES

Although much progress has been made in designing block ciphers that are cryptographically strong, the basic principles have not changed all that much since the work of Feistel and the Data Encryption Standard design team in the early 1970s as referred in this literature survey. It is useful to begin this discussion by looking at the published design criteria used in the Data Encryption Standard effort [10]. Then look at three critical aspects of block cipher design: the number of rounds, design of the function F, and key scheduling [14][15].

DES Design Criteria:

In this paper the criteria used in the design of Data Encryption Standard, as reported by Coppersmith in 1994 [16] in literature review section 2, focused on the design of the S-boxes and on the P function that takes the output of the S boxes (Figure 1 shown). The criteria for the S-boxes are as follows:

1. No output bit of any S-box should be too close a linear function of the input bits. Specifically, if we select any output bit and any subset of the six input bits, the fraction of inputs for which this output bit equals the XOR of these input bits should not be close to 0 or 1, but rather should be near 1/2.

2. Each row of an S-box (determined by a fixed value of the leftmost and rightmost input bits)should include all 16 possible output bit combinations.

3. If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.

4. If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.

5. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.

6. For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference.

7. This is a criterion similar to the previous one, but for the case of three S-boxes.

5. HILL CIPHER

In this paper referred with literature survey by Coppersmith pointed out that the first criterion in the preceding list was needed because the S-boxes are the only nonlinear part of Data Encryption Standard. If the S-boxes were linear then the entire algorithm would be linear and easily broken. In this paper described this phenomenon with the Hill cipher, which is linear. In this case the remaining criteria were primarily aimed at thwarting differential cryptanalysis and at providing good confusion properties [7].

Hill Cipher is developed by the mathematician Lester Hill in 1929. The core of Hill cipher is matrix manipulations. For encryption, algorithgm takes *m* successive plaintext letters and instead of that substitutes *m* cipher letters. In Hill cipher, each character is assigned a numerical value like $a = 0$, $b = 1$, ... , $z = 25$ [4]. The substitution of ciphertext letters in the place of plaintext letters leads to m linear equation.

In this paper mentioned in terms of column vectors and matrices: simply write as $C = KP$, where C and P are column vectors of length 3, representing the plaintext and ciphertext respectively, and K is a 3 X 3 matrix, which is the encryption key. All operations are performed mod 26 here. Decryption requires using the inverse of the matrix K.

The inverse matrix K^{-1} of a matrix K is defined by the equation $KK^{-1} = K^{-1}K = I$, where I is the Identity matrix. But the inverse of the matrix does not always exist, and when it does, it satisfies the preceding equation. K^{-1} is applied to the ciphertext, and then the plaintext is recovered. In general term we can write as follows:

For encryption: $C = E_k(P) = K_p$

For decryption : P =D_k (C) = $K^{-1}C$ = K-1 K_p =P

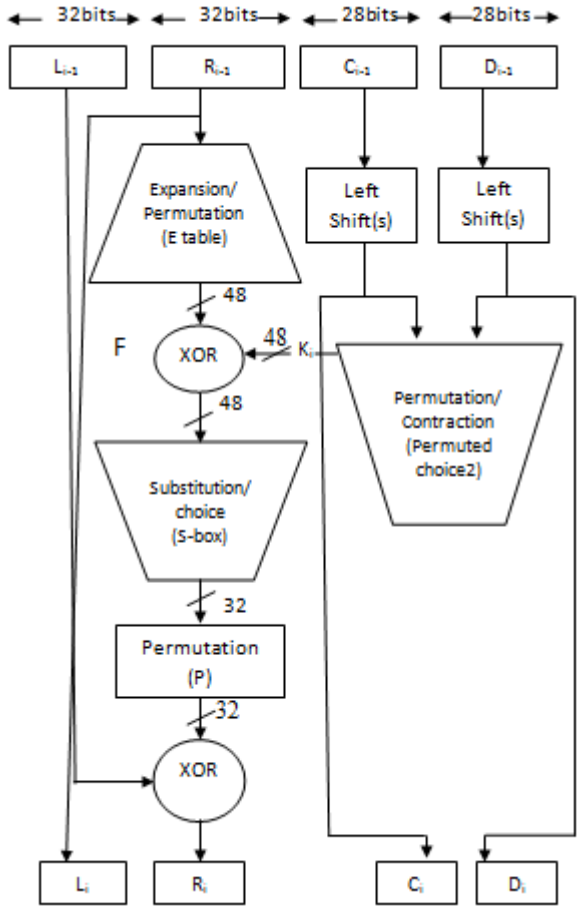

Fig 1 : Calculation of F(R, K)

6. S-DES ALGORITHM

In this paper briefly gives the overview of Simplified Data Encryption Standard Algorithm [1]. In this algorithm realising that studies on the attack of practical complex cryptosystems using evolutionary techniques have not been reported, in this paper described the study on the cryptanalysis of Simplified Data Encryption Standard (SDES), a modified version of DES (Data Encryption Standard) to make the implementation effort tractable. This example shows linear cryptanalysis by doing it on a modified version of S-DES[8]. The function f takes an 8-bit input (x) and an 8-bit subkey (k) as input and produces an 8 bit output (y). Hence write this as y = f(x, k) (mod 2)[9]. What if SDES had been designed in such a way that could write the function f as a linear combination of x and k modulo 2? That is, what if the function f were designed as y = f(x, k) = Mx + Dk (mod 2) where M and D are constant 8 x 8 matrices. The function f would look like this:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} M_{0,0} & M_{0,1} & M_{0,2} & M_{0,3} & M_{0,4} & M_{0,5} & M_{0,6} & M_{0,7} \\ M_{1,0} & M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} & M_{1,5} & M_{1,6} & M_{1,7} \\ M_{2,0} & M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} & M_{2,5} & M_{2,6} & M_{2,7} \\ M_{3,0} & M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} & M_{3,5} & M_{3,6} & M_{3,7} \\ M_{4,0} & M_{4,1} & M_{4,2} & M_{4,3} & M_{4,4} & M_{4,5} & M_{4,6} & M_{4,7} \\ M_{5,0} & M_{5,1} & M_{5,2} & M_{5,3} & M_{5,4} & M_{5,5} & M_{5,6} & M_{5,7} \\ M_{6,0} & M_{6,1} & M_{6,2} & M_{6,3} & M_{6,4} & M_{6,5} & M_{6,6} & M_{6,7} \\ M_{7,0} & M_{7,1} & M_{7,2} & M_{7,3} & M_{7,4} & M_{7,5} & M_{7,6} & M_{7,7} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$+ \begin{bmatrix} D_{0,0} & D_{0,1} & D_{0,2} & D_{0,3} & D_{0,4} & D_{0,5} & D_{0,6} & D_{0,7} \\ D_{1,0} & D_{1,1} & D_{1,2} & D_{1,3} & D_{1,4} & D_{1,5} & D_{1,6} & D_{1,7} \\ D_{2,0} & D_{2,1} & D_{2,2} & D_{2,3} & D_{2,4} & D_{2,5} & D_{2,6} & D_{2,7} \\ D_{3,0} & D_{3,1} & D_{3,2} & D_{3,3} & D_{3,4} & D_{3,5} & D_{3,6} & D_{3,7} \\ D_{4,0} & D_{4,1} & D_{4,2} & D_{4,3} & D_{4,4} & D_{4,5} & D_{4,6} & D_{4,7} \\ D_{5,0} & D_{5,1} & D_{5,2} & D_{5,3} & D_{5,4} & D_{5,5} & D_{5,6} & D_{5,7} \\ D_{6,0} & D_{6,1} & D_{6,2} & D_{6,3} & D_{6,4} & D_{6,5} & D_{6,6} & D_{6,7} \\ D_{7,0} & D_{7,1} & D_{7,2} & D_{7,3} & D_{7,4} & D_{7,5} & D_{7,6} & D_{7,7} \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \\ k_7 \end{bmatrix} \pmod 2$$

To do this we would only have to change the S-boxes to linear functions. All permutations and XORs are already linear functions. For example, P4 can be written as something like y = h(x):

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \pmod 2$$

and XOR4 can be written like z = i(x, y):

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \pmod 2$$

So if the S-boxes were linear equations then could easily find a linear function y = f(x, k) (mod 2). The SW switch function is just a permutation so it is also a linear function, so written as y = g(x) = Ex where E is a constant 8x8 matrix:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \pmod 2$$

Simplified Data Encryption Standard has two rounds of the function f (one for each subkey K1 and K2) with a switch function SW in the middle. So if P is the plaintext and C is the ciphertext then ignore the initial and final permutations, then:

$C = f(g(f(P, K1)), K2)$
$=M \times g(f(P, K1)) + D \times K2$
$=E \times M \times f(P, K1) + D \times K2$
$=E \times M \times (M \times P + D \times K1) + D \times K2$
$=E \times M^2 \times P + E \times M \times D \times K1 + D \times K2 \pmod 2$

Now define three new constant 8x8 matrices: $R= E \times M^2$, $S= E \times M \times D$, and $T=D$. Even if we use independent subkeys for K1 and K2, we have a linear equation:

$$C = R \times P + S \times K1 + T \times K2 \pmod 2$$

Furthermore getting a linear equation like this for any number of rounds, for example let's try four rounds where K1, K2, K3, and K4 are all independently chosen subkeys and not generated off of the same encryption key:

$$C = R \times P + S \times K1 + T \times K2 + U \times K3 + V \times K4 \pmod 2$$

Every element in R, S, T, U, and V is either a 0 or a 1. It means that the respective bit in P, K1, K2, K3, or K4 is either present in the equation for the ciphertext bit or not. So it might get something like

$$C_0 = P_3 + P_4 + P_5 + K1_0 + K1_2 + K1_3 + K1_5 + \ldots + K4_5 + K4_7$$

This says that bit 0 of the ciphertext is equal to bit 3 of the plaintext XORed with bit 4 of the plaintext XORed with bit 5 of the plaintext XORed with bit 0 of K1, etc. In this paper has similar equations for bits 1 through 7 of the ciphertext. So for a known ciphertext attack with one plaintext-ciphertext pair has 8 equations and 32 unknowns, which doesn't do much good.

But with 4 plaintext-ciphertext pairs have 32 equations and 32 unknowns. Using Gaussian elimination or Cramer's rule it is easy to see that could solve this system with something on the order of 32 calculations. In this case it found that two or more of 32 equations were not linearly independent then it could just add another plaintext-ciphertext pair. So a key-size of 32-bits in this case gives us a cryptanalytic effort of 32, instead of 2^{32}. Fortunately, Simplified Data Encryption Standard and Data Encryption Standard use S-boxes which are non-linear. But this doesn't mean that a linear equation for the function f won't hold for some pairs of inputs and outputs. In a perfect world, a linear equation modulo 2 on any input-output combination would hold exactly 50% of the time. This is because if we just take a bunch of randomly chosen bits and put them in a linear equation modulo 2 we would expect to get a 0 half the time and a 1 half the time.

But sometimes in S-DES and DES we can find linear functions for the S-boxes that occur with a probability ≠ 50%. For example, consider S-box 0 of S-DES with input $X_0X_1X_2X_3$ and output Y_0Y_1. The probability that $X_0+X_1+X_3+Y_0 = 0$ is 81.25%. The probability that $X_2+Y_1 = 0$ is 18.75%. Let's rewrite this one as the probability that $X_2+Y_1 = 1$ is 100% - 18.75% = 81.25%. So what we can do is come up with a bunch of equations that that will be satisfied by the S-boxes with a probability of more than 50%, regardless of what the subkey does to mash up the input. The bits of the subkey in the XOR8 will either make our linear approximations hold often or prevent them from holding often, but in either case they should push the likelihood of our equations holding away from 50%.

Let's do step-by-step linear cryptanalysis on a modified version of S-DES which has three rounds instead of two, uses three independent subkeys (K1, K2, and K3), and has no initial or final permutation as shown in Figure 2.

Step 1: We need about 16 good linear approximations for each S-box. They should hold for more than 50% of the possible inputs to the S-box. Each equation just takes a few bits from the input and also from the output of the S-box and adds them together modulo 2 (equivalent to XORing them all together) to get an answer of either 0 or 1. We'll need about eight equations for S0 and eight for S1.

Step 2: We need about 100 plaintext/ciphertext pairs. They don't have to be chosen plaintext, this is a known plaintext attack.

Step 3: Do step 4 for every possible subkey K3

Step 4: For each plaintext/ciphertext pair that we have

Step 4.1: Find Q, the output of the S-boxes for the second round. In our 3-round modified S-DES example this should be pretty easy. The XOR4 in the second round took the right half of the plaintext as input (since the first round didn't touch the right half of the plaintext) and produced the right half of the ciphertext as output (since the third round didn't touch the right half of the ciphertext). So if we XOR these two we should get the other input into the second round's XOR4. Then the inverse of a P4 is all we have to do and we know what the output of the S-boxes for the second round was.

Step 4.2: Find S, the input to the XOR8 of the second round. We can do this by taking the right half of the ciphertext and using it as input to E/P in the third round. We can just use the subkey K3 that we are trying and do the XOR8, S-boxes, and P4 for the third round. The output of P4 and the left half of the ciphertext are one input and the output of the XOR4. We just XOR these two together to get the other input to the XOR4 in the third round. We can trace this input back and observe that it is the same as the input to the E/P in the second round. So we plug it into E/P for the second round and we have S.

Step 4.3: See if each of our linear approximations hold for S as input and Q as output. We don't really care what the subkey K2 is because any bit in K2 has a 50% chance of being a 0 and not touching anything. If it were a 1 then it would change one of our S bits but we wouldn't really care because our linear approximations are still biased away from

50%. If S was chosen well (meaning our guess for K3 was good) then we would expect S and Q to show bias with the linear approximations. If S is not anything like the input that was really used when the ciphertext/plaintext pair was generated then we're just plugging in random bits for S and Q

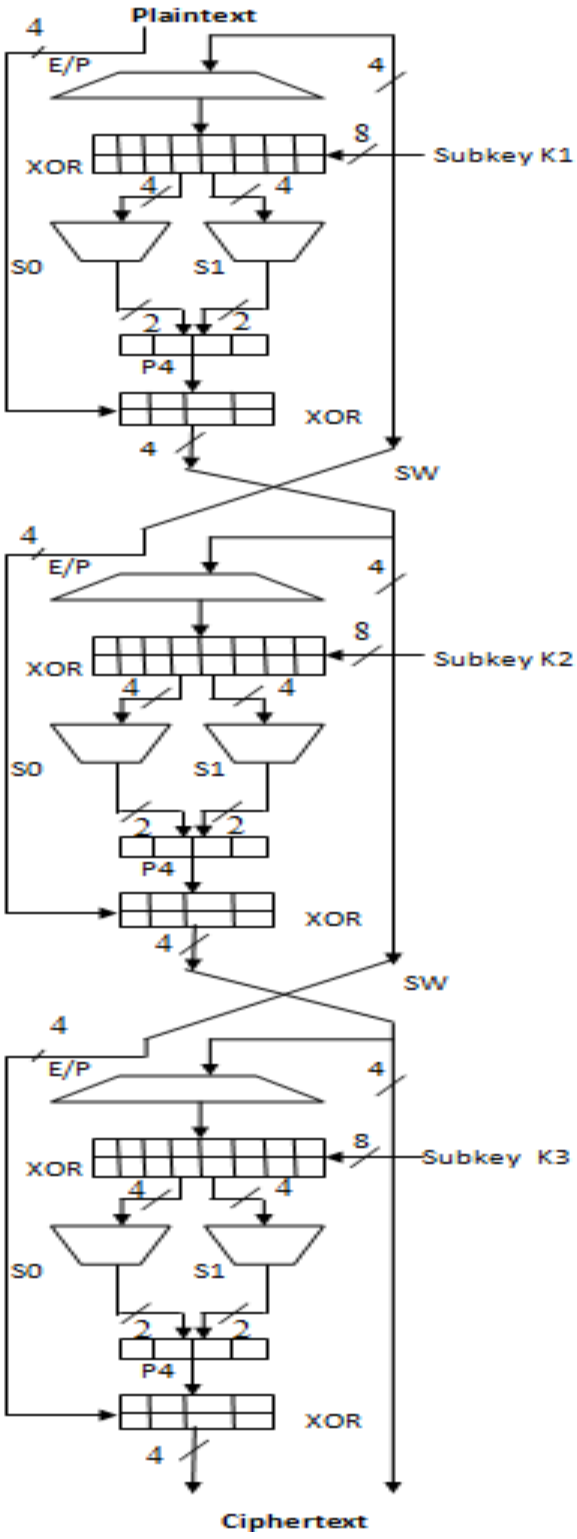

Fig 2: Simplified DES Scheme Encryption Details

and we would expect our linear approximations to hold about 50% of the time. The subkey that we guess that shows the highest deviation away from 50% is the one most likely to be the real K3. If not then the real K3 is definitely second or third on the list. The important thing to notice is that once we know K3 we can attack K2 the same way and then K1. The level of effort to break the cipher then comes from the size of the subkey and not the key size. $O(2^8)$ to break K3 + $O(2^8)$ to break K2 + $O(2^8)$ to break K1 means an overall effort of $O(2^8)$. This is a lot easier than a brute force attack on the 24-bit key which would be $O(2^{24})$. Linear cryptanalysis won't produce such dramatic results on real DES, though, because DES uses 16 rounds and a much better S-box design. The level of effort to do linear cryptanalysis on DES is still dependent on the size of the subkey, but we need a lot of plaintext/ciphertext pairs which makes it pretty much infeasible. The math is a lot harder, too, because we end up trying to find linear equations for 15 rounds of DES instead of just a single round.

7. EXPERIMENTAL RESULTS

Number of experiments is carried out to outline the effectiveness of Linear Cryptanalysis. The Linear Cryptanalysis is coded in MATLAB 7, and tested on more than 1000 benchmark data sets adapted. We consider 30 different sets of distinct known plaintexts with different secret keys. In each experiment the behavior of the statistic test is studied for the right key and also for one wrong key. As predicted by the theoretical model, when more than $2^{30.2}$ distinct known plaintexts are used, the correct key is very likely to pass the test, while the wrong keys would fail. Access to the full codebook leads to the key recovery with negligible error probability. When using 2^{28} distinct known plaintexts, the right key survives with high probability.

Table. 1 The number of bits recovered from the key using Linear cryptanalysis.

Amount of Cipertext	Time(Minute)	Number of bits matched in the Key
100	60	6
200	55	2
300	51.3	8
400	47.1	6
500	44	8
600	40	9
700	30	4
800	28.5	2
900	25	9
1000	20	7

8. CONCLUSIONS

In this paper we showed how to use the matrix method to establish linear approximations automatically. We used this method to obtain several linear approximations over 14 rounds of Block. We believe that the described method will be useful for analysis of other block ciphers, too. Based on the 14-round distinguisher we present an attack on 22 rounds of Block. While the previous attack, which can break the same number of rounds, uses chosen plaintext pairs, our attack

assumes only that the plaintexts are distinct. Finally, we implement the attack for a small variant of Block and run simulations to experimentally validate the statistical model of linear cryptanalysis presented.

9. REFERENCES

[1] Rajashekarappa, Dr K M Sunjiv Soyjaudah, 2012, ICIP, Cryptanalysis of Simplified-Data Encryption Standard Using Tabu Search Method, pp. 561-568, ©Springer-Verlag Berlin Heidelberg.

[2] Biham, E., and Shamir, A, 1993, Differential Cryptanalysis of the Data Encryption Standard. NewYork: Springer-Verlag.

[3] Comppersmith, D, May, 1994, "The Data Encryption Standard (DES) and Its Strength Against Attacks", IBM Journal of Research and Development.

[4] Matsui, M,1993, "Linear Cryptanalysis Method for DES Cipher." *Proceedings, EUROCRYPT '93*, published by Springer-Verlag.

[5] Barker, W, 1991,Introduction to the Analysis of the Data Encryption Standard *(DES)*. Laguna Hills, CA: Aegean Park Press.

[6] Electronic Frontier Foundation, 1998, Cracking DES: Secrets of Encryption Research,Wiretap Politics, and Chip Design. Sebastopol, CA: O'Reilly.

[7] BibhudendraAcharya, GirijaSankarRath, Sarat Kumar Patra, Saroj Kumar Panigrahy, 2007, "Novel Methods of Generating Self-Invertible Matrix for Hill Cipher Algorithm", *International Journal of Security*, Vol 1, Issue 1, pp. 14-21.

[8] William Stallings, 1989, "Cryptography and Network Security Principles and Practices", Fourth edition, McGraw- Hill, 2003.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science.

[9] Behrouz A. Forouzan, 2006, "Cryptography and Network Security", Firstedition, McGraw- Hill.

[10] Atul Kahate, 2003, "Cryptography and Network Security", TMH.

[11] Rajashekarappa and Dr. K M S Soyjaudah, 2012, "Heuristic Search Procedures for Cryptanalysis and Development of Enhanced Cryptographic Techniques" Published at International Journal of Modern Engineering Research (IJMER), Vol.2, Issue.3, pp-949-954.

[12] Poonam, G, 2009, "Crypatanalysis of SDES via Evolutionary Computation Techniques". International Journal of Computer Science and Information Security 1(1).

[13] Chanas, S., Kobylanski, P, 1996, A New Heuristic Algorithm Solving the Linear Ordering Problem. Computational Optimization and Applications 6, 191–205.

[14] Ayman M. B. Albassal, Abdel-Moneim A. Wahdan, 2004, Genetic Algorithm Cryptanalysis of a Fiestel Type Block Cipher, International Conference on Electrical, Electronic and Computer Engineering, Egypt, PP. 217–221.

[15] Garg, P, 2010, Evolutionary Computation Algorithms for Cryptanalysis: A Study, (IJCSIS) International Journal of Computer Science and Information Security,Vol. 7, No. 1.

[16] Coppersmith, D, 1994, "The Data Encryption Standard (DES) and Its Strength Against Attacks." IBM Journal of Research and Development.

[17] Rajashekarappa, Dr K M Sunjiv Soyjaudah, 2012, Comparative Cryptanalysis of Simplified-Data Encryption Standard Using Tabu Search and Simulated Annealing Methods, International Journal of Engineering Research and Development, pp. 07-12.