Genetic Algorithm for DES Cryptanalysis

Tarek Tadros†, Abd El Fatah Hegazy††, and Amr Badr†††

† Collage of Computing& I.T., Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt
††Prof. Dr., Collage of Computing& I.T., Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt
††† Prof. Dr., Faculty of Computers & Information, Cairo University, Cairo, Egypt

Summary

This paper discusses the use of Genetic Algorithm (G.A.) for DES cryptanalysis for the most two famous attacks Differential Attack & Linear Attack in order to enhance the attack overall performance. This paper presents a new proposed fitness function for G.A. Linear Attack & an enhanced technique for G.A. Differential Attack in order to enhance performance of previous G.A. Differential Attacks [1][8].G.A. Linear Attack was implemented on DES-4 Rounds while G.A. Differential Attack was implemented on DES 6-rounds. Both attacks were capable to break DES and extract key bits in less than one second. Experimental results show that G.A. enhanced the overall performance & memory consumption. Therefore G.A. can be used to enhance Differential Attack & Linear Attack for various DES-like cryptosystems.

Key words:
DES, Differential Cryptanalysis, Linear Cryptanalysis, Genetic Algorithm, Evolutionary Cryptanalysis.

1. Introduction

During the seventies of the last century I.B.M. invented the Data Encryption Standard (D.E.S), adopted by National Bureau of Standards (N.B.S.) [14].Since then DES has been a worldwide standard and become the most widely used cryptosystem for many commercial applications. For many years, DES withstood all known attacks successfully until the beginning of the 90's when Adi Shamir & Eli Biham invented a new chosen plain text attack called Differential Attack which was capable to break DES in time less than exhaustive search[5]. Differential Attack opened the door for other new innovative attacks such as Linear Attack which is a known plain text attack developed on 1993 by Matsuru Matsui and was also capable to break DES in time less than exhaustive search [13]. Both attacks be implemented to other various DES-like cryptosystems [4, 7, 10, 12], knowing that Differential Attack was capable to break N-Hash function as well [6]. An interesting study in 1994 by K. Langford and E. Hellman showed that both Differential Attack & Linear Attack can be combined together so as to be used for DES Cryptanalysis [15]. However, both attacks Differential Attack & Linear Attack consume memory and time.

Genetic Algorithm (G.A.) is an effective search algorithm inspired by Darwinian Law "Survival of the fittest"[9].G.A. is an incremental solution. It starts with an initial solution then increments guided enhancements in order to reach the optimal solution. However, the problem is cryptosystems do not work this way because a one-bit change in key leads to a total different output. Therefore if G.A. can be used in DES cryptanalysis guided by either Differential Attack or Linear Attack this might enhance the performance and reduce memory.

This paper presents a new proposed fitness function for G.A. guided by Linear Attack so as to break DES 4-rounds and extract S-Box 1 sub-key in less than 1 second.

For G.A. Differential Attack an enhanced technique is been presented based on 2 major improvement:-
First: by applying a new filtering rule which can be used to decline wrong pairs for Differential Attack.
Second: by using an enhanced fitness function for key evaluation in order to increase the efficiency of Genetic progressive search.

This paper is not limited to theoretical hypothesis. It provides a practical attack implemented by C++ language for both G.A. Linear Attack &G.A Differential Attack. Genetic Algorithm part was done through an integration with another 3rd party ready-made genetic library (GALIB)[11] in order to provide a generic model which can be used with any other ready-made genetic library without the need for a customized Genetic Algorithm. Both attacks are supported by fitness function, algorithm & results.

Results show that G.A. enhanced performance and provide less memory consumption for both attacks. Therefore G.A. Can be used to replace both Linear Attack & Differential Attack for various DES-Like cryptosystems

Manuscript received May 5, 2010
Manuscript revised May 20, 2010

2. DES Background

DES operates on 64 bit-size block starting with an initial permutation (IP). Then block is broken into right half (R) and left half (L). Each half has a fixed length of 32 bit. Right half (R) is expanded into 48 bits using an expansion Function (E), then XORed with the round key and sent to 8 S-Boxes producing 32 new bits. Those 32 bits are permuted then XORed with the left half becoming the new right half, while the new left half is assigned with the old right half.

$L_i = R_{i-1}$.

$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$.

Where $f(R_{i-1}, K_i) = P(S(E(R_{i-1}, K_i)))$.

The above operation is repeated 16 times forming 16 round of DES, ending by a final permutation (IP^{-1}) performed (Figure 1).

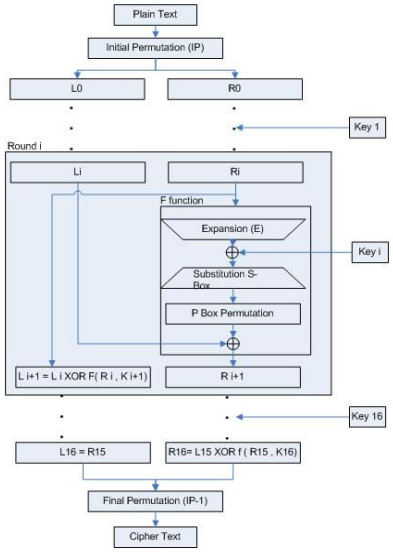

Figure1 DES Algorithm.

3. Key Schedule

DES key is originally 64 bit size key. The key is permuted (PC^{-1}) according to fixed permutation then reduced to 56 bits after discarding every eight bit since the eighth bit is mainly used for parity check. The remaining 56 key bits are spitted into 2 halves each of fixed size 28 bit. Cyclic left shift is performed by one or two bits shift depending on round number. Finally, the round sub-key is generated by selecting 48 bits according to a compressed permutation (Figure 2).

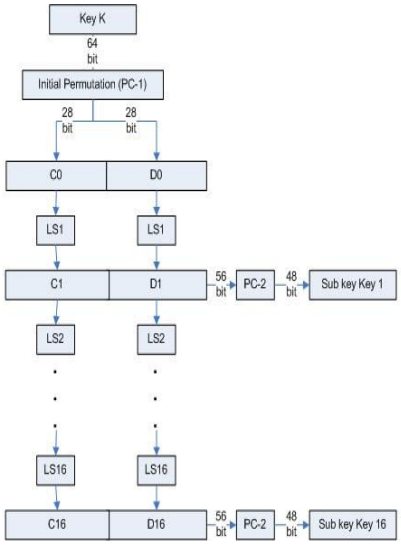

Figure2 Key Schedule.

4. Differential Cryptanalysis

Differential Cryptanalysis is a chosen plain text attack, where the attacker choose plain text pairs with specific XOR difference and try to analyze the effect on the differences of the resultant cipher text pair. This difference can be used later to assign probabilities for possible keys and locate the most probable key bits. This type of attack is based on n-round characteristic which allows pushing the knowledge of a plain text XOR to the knowledge of intermediate XOR after as many rounds as possible. This is called an n-round characteristic. Every round has a particular plain text XOR P, a particular XOR of the data in the n^{th} round T and a probability p from which the XOR of the data in the n^{th} round is T.

Right pairs: Any pair whose plain text XOR is P and XOR of data in the n^{th} round (using a particular key) is T, is called a right pair with respect to an n-round characteristic. Any other pair is called wrong pair [4].

Signal to noise ratio: Signal to noise ratio is an engineering concept used to determine how much signal has been corrupted by noise. However, in Differential Attack according to E. Biham & A. Sahmir, signal to noise is the ratio between the number of right pairs and the average count of the incorrect sub-keys in a counting scheme. The signal to noise ratio of the counting scheme is donated by S/N.

$$S/N = \frac{m_1 P m_2}{m_1 \frac{\alpha\beta}{2^k} m_2} = \frac{2^k P}{\alpha\beta} \quad [5]$$

Where m = number of created pairs, P=probability of characteristics, α = average number of suggested keys per pair, m.β= number of analyzed pairs among all pairs, k=number of key bits counted & 2^k=number of occurrence of possible keys. Signal to noise ratio is used to determine the number of times the right key is counted over the number of times a random key is counted.

5. Linear Cryptanalysis

One of the main advantages of the Linear Attack is that it is a known plain text attack where the attacker needs to know plain text with their cipher output without the need to access specific chosen plain text as in case of Differential Attack. Also Linear Attack can be applied to only cipher attack in special cases (if plain text consists of natural English sentences represented by ASCII codes)[13]. Hence Linear Attack is more practical than Differential Attack. The main principle of Linear Attack for n-round DES cipher is to find an effective linear expression relating plaintext, cipher-text, round function and key for a given cipher algorithm which holds with probability $p \neq ½$ as follows:-

$P[i_0, i_1, i_2 \dots i_a] \oplus C[j_0, j_1, j_2 \dots j_b] \oplus F_n(C_L, K_n)[\ell_0, \ell_1, \ell_2 \dots \ell_d] = K[k_0, k_1, k_2 \dots k_c]$

Where P=Plain-text, C=cipher-text, F=round function, K=sub-key, $i_0, i_1, i_2 \dots i_a, j_0, j_1, j_2 \dots j_b$, $\ell_0, \ell_1, \ell_2 \dots \ell_d$ & $k_0, k_1, k_2 \dots k_c$ denotes fixed bit locations. While $|p - ½|$ represent the magnitude of the above equation.

Mitsuru presented an algorithm based on maximum likelihood method & piling up lemma in order to construct linear approximation. Once succeeded in reaching an effective linear expression, it is possible to determine some key bits [13].It is worthy to study the S-Box trying to find a linear expression since it is the only non linear part of the DES algorithm. Mitsuru defined a rule which can be used in order to get probability of the S-Box XOR input bits coincides with S-Box XOR output bits as follows:-

Definition: for a given S-Box S_a (a = 1, 2… 8), $1 \leq \alpha \leq 64$ and $1 \leq \beta \leq 15$,
Define N S_a (α, β) as number of times out of 64 input patterns of S_a, such that an XORed value of the input bits masked by α coincides with an XORed value of the output bits masked by β as follow:-

$N\ S_a\ (\alpha, \beta) \neq \{x|0 \leq x \leq 64;\ (\oplus^5_{l=0}(x\ [S] \bullet \alpha\ [S])) = (\oplus^3_{t=0}(S_a(x)\ [t] \bullet \beta[t]))\}$ [13]

Where • Denotes bit wise AND operator

6. Genetic Algorithm Background

Genetic Algorithm is incremental optimization algorithm that is inspired by both natural selection and natural genetics [2], It simulates biological evolution including matting, cross-over, and mutation and selection according to the fittest.

Holland shows that fitness increases exponentially over generation according to Holland's schema theorem [9] as follows:-

$$m(H, t+1) \geq \left(\frac{m(H,t) f(H)}{a_t}\right) |1 - p|$$

Where m (H, t + 1)=number of strings belonging to schema "H" at generation "t", *f (H)* is the fitness of schema "H", a_t =average fitness at generation "t". p=probability that cross-over or mutation will destroy schema "H" as follows.

$$p = \frac{\delta(H)}{\ell - 1} P^c + o(H)\, P_m$$

Where $\delta(H)$ = length of H defined as the distance between the first and last fixed bit of schema "H", ℓ=chromosome length, P_c=probability of cross-over, o(H) =order of schema "H" defined by the number of fixed string positions of Schema "H", P_m=probability of mutation.

7. Genetic Algorithm Differential Attack

Previous G.A. Attacks used below fitness function in order to evaluate each chromosome however, an enhanced fitness function will be presented later

$C_r = \frac{n_{sr}}{n_p}$ [1][8].

Where C_r = Chromosome correctness, n_{sr}=number of right pairs generated by the chromosome & n_p=total number of right pairs.

Above fitness function will always have a value less than 1 except when $n_{sr}=n_p$. Since n_p is always fixed, therefore C_r value is directly proportional to n_{sr} value and will be incremented with each correct right pair. Hence Holland's schema theorem will be satisfied and G.A. will perform progressive search producing the correct key. Knowing that Right key will always generate the max number of right pairs, therefore right key will obtain the max fitness score among all other keys.

Genetic Algorithm will be used for DES 6-rounds cryptanalysis based on 3-round characteristics with
ΩP = 40800000 04000000x trying to deduce 30 bit from the

original key. A 30-bit chromosome will be used holding 5 sub-keys for S-Box S2, S5…S8 each sub-key is 6 bit size. Initial permutation and final permutation will be omitted since they do not have any cryptanalysis effect. According to DES 3-round characteristic (figure 3), if we added a fourth round having 'd ='b ⊕ 'C with an input to the fourth round = 40800000x. According 5 S-Boxes S2,S5…S8 in the fourth round will have their output XOR = zero therefore the 5 S-Boxes mentioned will have zero XOR output as well.

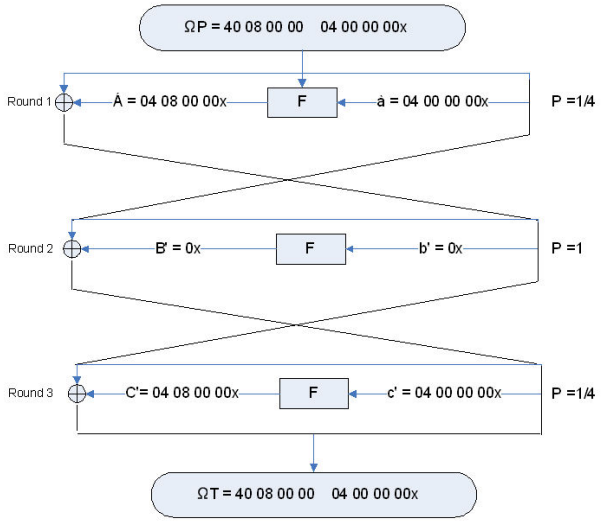

Figure3 DES 3-rounds characteristics

Corresponding 6 round can be found by 'F = 'c ⊕ 'D ⊕ '[illegible] [4]. Expected right pair's probability equal to 1/16 with respect to 3 rounds characteristics. In order to improve the performance of the above genetic attack 2 enhancements required to be done.

First Enhancement:- D. Stinson suggested an additional filtering rule which can be used at the beginning of the attack so as to increase the percentage of the right pairs from 1/16 to 1/3 as follows:-

$$\prod_{j \in \{2,5,6,7,8\}} |Test\ j| \quad [3]$$

Such that |test j| should be > 0 for j = 2, 5, 6, 7, 8.
In other words right pair should generate number of suggested keys for S-Box i where i ∈ {2, 5, 6, 7, 8} otherwise this pair should be discarded and marked as a wrong pair.

Second Enhancement: - Enhanced fitness function to evaluate the average fitness chromosome rather than evaluating the whole 30 bit chromosome by breaking the 30 bit chromosome into 5 sub-keys and evaluate each key separately.

Pervious G.A. Attack fitness function evaluates the whole 30 bit chromosome whether it fails or succeeds[1][8],the problem is that the chromosome may carry some correct genes (sub-keys) along with other wrong genes (sub-keys) in this case the whole 30 bit chromosome will fail and therefore correct genes(right sub-keys) might be lost. While in above enhanced fitness function the average fitness score will be considered, therefore such chromosome will be eligible to compete within the next generation hence correct genes (sub-keys) will not be lost among next generation. Of course this will guarantee incremental fitness and will enhance the performance.

Algorithm:

Input: Number of right pairs with respect to round characteristics and filtering condition.
Output: 30 bit of the last round sub-key.

Applying Genetic Algorithm Differential Attack as follows:

1-Generate a pool of random pairs with fixed difference where P=40800000 04000000x and encrypt those pairs using the same key.
2-Store right pairs &discard wrong pairs according to round characteristics
3-Apply filtering rule to eliminate 2/3 of the wrong pairs according to first Enhancement presented before.
4-Create an initial population of chromosomes each of size 30-bit (each chromosome carries 5 S-box sub-keys S2, S5 …S8.
5-Apply cross-over, mutation according to cross-over, mutation percentage.
6-Evaluate each chromosome according to enhanced fitness function which was presented before (chromosome which carries right S-box sub-keys S2, S5…S8) should get maximum fitness score).
7-Repeat above steps from step 4 till maximum generation is reached.
8- If maximum fitness is less than target fitness therefore G.A. failed or at least chromosome may contain wrong sub-key(s)
Else select chromosome with the maximum fitness score which contains 5 sub-keys for (S2, S5 …S8).
9- Adjust 30-bit key in their position, then apply exhaustive search to get the remaining 26 bits of the key.

Note: above algorithm can be used with different 3-round characteristics (00200008 00000400x) so as to gain additional 12 bit from the original key bits (for S1 & S4) so as to reduce exhaustive search from the remaining key bits from 26 bits to 14 [3].

8. Genetic Algorithm Linear Attack

Linear cryptanalysis tries to find a probabilistic parity relation between selected bits of plain text, cipher and key. These parity relation derive from parity relation within S-Boxes that differ from the uniform 50-50 distribution and which can be connected through different rounds[15],key bits that generate higher bias value are suspected to be correct key bits.

Therefore G.A. target is to find key bits having max bias value genetically.

G.A. attack was implemented on 4 round DES. Linear expression for DES 4-rounds is

PH[7, 18, 24, 29] + CH[7, 18, 24, 29] + PL[15] + CL[15] = K_1[22] + K_3[22]

Where PH=plain-text right 32 bits, PL=plain-text left 32 bits, CH=cipher-text right 32 bits, CL = cipher-text left 32 bits, K_n is sub-key K at round n.

Following fitness function can be used:-

$$Cf = ABS\left(\frac{\sum 1..p_n\ \beta - \frac{P_n}{2}}{P_n}\right)$$

Where Cf is Chromosome fitness, P_n= Total number of pairs, β = Bias value generated for each pair,
Target is to find S-Box 1 sub-key which generates the max. Cf value genetically.

Algorithm:

Input: Number of random plain text pairs with their cipher text pairs all encrypted with same key.
Output: 6 bit represent S-Box 1 sub-key.

Applying Genetic Algorithm Linear Attack as follows:

1-Generate a pool of random pairs and encrypt those pairs using same key.
3-Create an initial population of chromosomes each of size 6-bit (each Chromosome represent S-Box 1 sub-key).
4-Apply cross-over, mutation according to cross-over, mutation percentage.
5-Evaluate each chromosome according to fitness function presented above (chromosome which carries right S-Box 1 sub-key should get maximum fitness score).
6-Repeat above steps from step 4 till maximum generation is reached.
7- If maximum fitness is less than target fitness therefore G.A. failed
Else select chromosome with the maximum fitness score which represent S-Box 1 sub-key
8- Adjust 6-bit key in their position. Then apply exhaustive search to get the remaining key bits.

9. Implementation

Implementation was done with C++ in order to stick to a bit level so as to achieve maximum performance.
Genetic part was done through (GALIB) [11] in order to prove that G.A. attack does not require a tolerated or specific genetic implementation. Genetic Attack was done for the 2 most famous DES attacks Differential Attack & Linear Attack in order to prove the G.A. can be applied to any attack and not limited to specific attack.
G.A. Differential Attack implemented on DES 6-round while G.A. Linear Attack implemented on DES 4-round.

10. Results

Attack was implemented many times in order to guarantee success of this model in order to break DES encryption algorithm.

G.A. Differential Attack was able to break DES 6 rounds successfully and extracted following S-Boxes sub-keys S2, S5, S6, S7 & S8 with 100 right pairs having specific XOR difference in less than 1 second.

While G.A. Linear Attack was able to break DES 4 rounds successfully and extracted the first S-Box sub-key S1 with 100 random pairs in less than 1 second.

G.A. Differential Attack was implemented with the following parameters:-
Number of right pairs = 100
Chromosome size = 30 bits binary string
Population size = 50
No. of generation = 100
Probability of cross-over = 0.6
Probability of mutation = 0.05
Selection method is Roulette Wheel

G.A. Linear Attack was implemented with the following parameters:-
Number of random pairs = 100
Chromosome size = 6 bits binary string
Population size = 10
No. of generation = 20
Probability of cross-over = 0.5
Probability of mutation = 0.05
Selection method is Roulette Wheel
Statistics for G.A. Differential Attack:-

Number of selections since initialization# = 10000.
Number of cross-over since initialization# = 2870.
Number of mutations since initialization# =10000.
Number of replacements since initialization# =10000.
No. of population evaluations since initialization# =101.
Maximum score since initialization# = 0.404.
Minimum score since initialization# =0.072.
Avg. of all scores ('on-line' performance)# =0.327973.
Avg. of max. scores ('off-line' performance)# =0.366.
Avg. of min. scores ('off-line' performance)# =0.31464.
Mean score in initial population# = 0.105.
Maximum score in initial population# =0.192.
Minimum score in initial population# =0.072.
Standard deviation of initial population# =0.0224508.
Mean score in last population# = 0.38228.
Maximum score in last population# =0.404.
Minimum score in last population# =0.372.
Standard deviation of last population# =0.0120505.

Statistics for G.A. Linear Attack:-
Number of selections since initialization# = 200.
Number of cross-over since initialization# = 98.
Number of mutations since initialization# =61.
Number of replacements since initialization# =200.
No. of population evaluations since initialization# =21.
Maximum score since initialization# = 0.27.
Minimum score since initialization# =0.
Avg. of all scores ('on-line' performance)# =0.20105.
Avg. of max. scores ('off-line' performance)# =0.2275.
Avg. of min. scores ('off--line' performance)# =0.184.
Mean score in initial population# = 0.052.
Maximum score in initial population# =0.11.
Minimum score in initial population# =0.
Standard deviation of initial population# =0.0379473.
Mean score in last population# = 0.27.
Maximum score in last population# =0.27.
Minimum score in last population# =0.27.
Standard deviation of last population# =0.

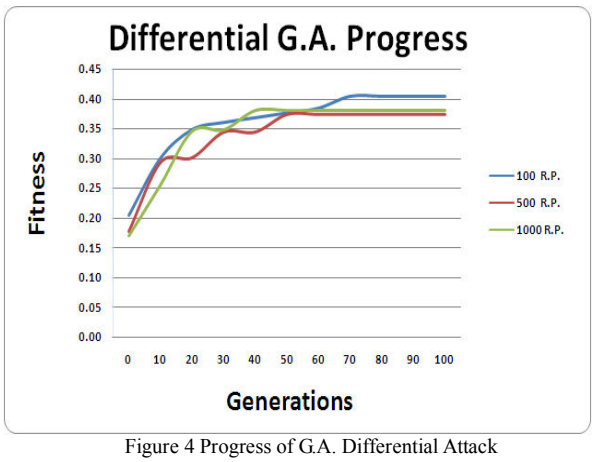

Figure 4 Progress of G.A. Differential Attack

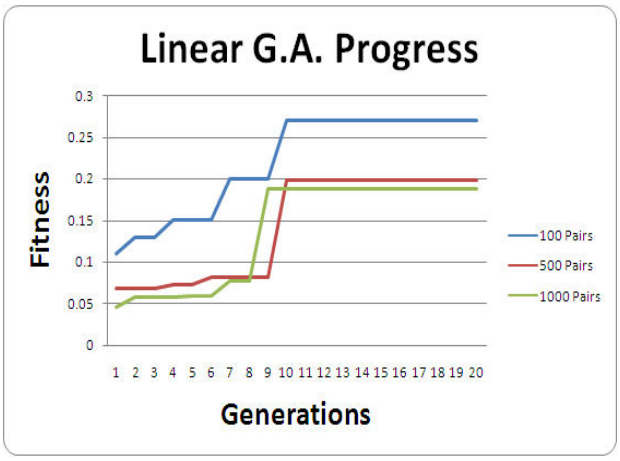

Figure 5 Progress of G.A. Linear Attack

Conclusion

The motivation of this work is to present a new proposed attack for G.A. Linear Attack and a new enhancement for G.A. Differential Attack.

The proposed approach presents a new fitness function for G.A. Linear Attack. G.A. Linear Attack was capable of breaking DES-4 rounds in less than 1 second.

A new enhancement was also proposed for G.A. Differential Attack based on 2 major enhancements (filtering rule & enhanced fitness function). Experimental results show that the above enhancement increased the overall performance compared to previous G.A. Differential Attack methods. G.A. Differential Attack was capable of breaking DES-6 rounds in less than1 second.

Although G.A. complexity tends to $O(n^3)$ where n is number of pairs, still G.A. enhances performance and provides less time & memory consumption for both Differential Attack& Linear Attack for the following reasons.

- G.A. evaluates each key (chromosome) separately without a need to keep counter for all keys then compares them. This shows enhancements in memory consumption & performance.
- Chromosome which gets maximum fitness score is eligible to compete in the next generation therefore G.A. guarantee incremental progressive search.
- Chromosome evaluation according to average chromosome fitness guarantees that good genes (Sub-keys) are not lost among next generations.

For the above mentioned reasons G.A. Attack can be implemented to various DES-Like cryptosystem in order to enhance attack for both Differential Attack & Linear Attack.

Authors are aiming to extend their future work so as to include:-

- Applying G.A. Differential Attack & G.A. Linear Attack to higher DES rounds like DES 8-round.
- Applying G.A. Differential Attack & G.A. Linear Attack to other DES-like cryptosystems subjected to Differential Attack & Linear Attack like FEAL encryption algorithm.
- Applying G.A. for combine Differential-Linear Attack.

Acknowledgments

I would like to express my deep thanks to Allah, my family, my supervisors, my colleagues and to all who supported me so as to present this work.

References

[1] Ayman Bahaa Albassal, "Intelligent Systems for Information Security", Ph.D. Thesis, Ain Shams University,2004, PP.82.
[2] David A Coley ,"An Introduction to Genetic Algorithms for Scientists and Engineers", World Scientific, 1999, pp1.
[3] Douglas Stinson, Cryptography Theory and Practice, Second edition, CRC Press, 1995, PP.98-103.
[4] Eli Biham and Adi Shamir, "Differential Cryptanalysis of DES-like cryptosystems", Journal of Cryptology, Vol.4, No.1, PP.3-72, 1991.
[5] Eli Biham and Adi Shamir, "Differential Cryptanalysis of Data Encryption Standard", Springer-Verlag, 1993.
[6] Eli Biham and Adi Shamir, "Differential Cryptanalysis of FEAL and N-Hash", technical report CS91-17, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, 1991.
[7] Eli Biham and Adi Shamir, "Differential Cryptanalysis Of Snefru, Khafre, REDOC-II,LOKI and Lucifer", Technical report CS91-18,Departement of Applied Mathematics and Computer Science, The Weizmann Institute of Science,1991.
[8] Hassan M. Hassan, Bayoumi I. Bayoumi, Fathy S. Holail, Bahaa Eldin M.Hassan and Mohamed Z. Abd El Mageed ,"A Genetic Algorithm for Cryptanalysis with Application to DES-like Systems", International Journal of Network Security,Vol.8,PP.177-186 , March 2009.
[9] John Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.
[10] Kazuo Ohta and Kazumaro Aoki, "Linear Cryptanalysis of the Fast Data Encipherment Algorithm", Advances in Cryptology-CRYPTO'94, Springer-Verlag, 1994.
[11] Matthew Wall, "GAlib version 2.4", http://lancet.mit.edu/ga, 1996 Matthew Wall
[12] Mistsuru Matsui, "A New Method for Known Plain text Attack of FEAL Cipher", Advances in Cryptology - EUROCRYPT'92, 1992.
[13] Mistsuru Matsui, "Linear Cryptanalysis Method for DES Cipher", Advances in Cryptology-EUROCRYPT'93, 1994.
[14] National Bureau of Standards, "Data Encryption Standard, Federal Information Processing Standard (FIPS)", Publication 46, National Bureau of Standards, U.S. Department of Commerce, January 1977.
[15] Susan K. Langford and Martin E.Hellman,"Differential-Linear Cryptanalysis", Advances In Cryptology-CRYPTO'94, 1994.

Tarek Tadros B.Sc. of Information Technology, Sadat Academy 1997, This paper was presented to obtain Master degree in computer science.

Abd El Fatah Hegazy, Prof. Dr. ,Dean Assistant for Post Graduate Studies, Collage of Computing and Information Technology, Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt.

Amr Badr, Prof. Dr. of Artificial Intelligence, Faculty of Computers & Information ,Computer Science Department, Cairo University, Cairo, Egypt