A Genetic Algorithm for Cryptanalysis of DES-8

Hasan Mohammed Hasan Husein[1], Bayoumi I. Bayoumi[2], Fathy Saad Holail[3], Bahaa Eldin M. Hasan[4], and Mohammed Z. Abd El-Mageed[5]

(Corresponding author: Hasan Mohammed Hasan Husein)

Research Development Center National Defense Council[1]

Department of Mathematics, Faculty of Science, Ain Shams University[2]

Head of C.R. Division, Research Development Center, National Defense Council[3]

C.R. Division, Research Development Center, National Defense Council[4]

Department of Computer Science, Faculty of Engineering, Al-Azhar University, Nasr City, Cairo, Egypt[5]

(Received Dec. 10, 2005; revised and accepted Jan. 10, 2006)

Abstract

Various cryptosystems exploit exhaustive techniques to search, missing-the-mark, key space. Such search techniques should be guided in order to be computationally adequate. Here, a Genetic Algorithm, GA, is proposed for the cryptanalysis of DES-like systems to find out the underlying key. The genetic algorithm approach is adopted, for obtaining the exact key by forming an initial population of keys that belong to the key subspace. The premature convergence (local minimum) could be avoided by dynamic variation of control parameters that can affect the fitness function. In this paper a new method has been developed for the first time to break DES with eight rounds. Its performance is considerably faster than exhaustive search and differential cryptanalysis, DC. The new method can be applied to a variety of DES-like systems instead of the available DC techniques.

Keywords: DES-like, differential cryptanalysis, genetic algorithm, fitness function

1 Introduction

Genetic Algorithms (GA's) had been explained by Holland [4] as an adaptive heuristic search method that depends on the evolutionary ideas of natural selection and genetics. The basic goal of a genetic algorithm is to simulate the process of natural evolution, taking into consideration the principle of survival of the fittest. It is generally used in situations where the search space is relatively large and cannot be traversed efficiently by classical search methods. This is mostly the case with problems whose solution requires evaluation of many apparently unrelated variables. As GA's represent an intelligent mapping of a random search space to a guided search space in which the problem solution could be found. The algorithm performs the following steps:

1) Generate an initial population, randomly.
2) Compute the fitness for each individual in the current population.
3) Define selection probability of each individual so that it is proportional to its fitness.
4) Generate the next current population by probabilistically selecting the individuals from the previous current population, in order to produce offspring via the genetic operators represented by: selection, crossover and mutation.
5) Repeat Step 2 until satisfactory solution is obtained.

Holland [3] has analyzed the influence of GA operators (selection, crossover and mutation) on the expected number $m(H,t)$ of schemata H when going from one generation t to the next $t+1$. A good discussion can be found in [3]. Holland's schemata theorem can be expressed as:

$$m(H,t+1) \geq m(H,t)\frac{f_H(t)}{\bar{f}(t)}[1 - P_c\frac{\delta(H)}{l-1} - o(H)P_m],$$

where $f_H(t)$ denotes the fitness value of the string representing schema H; $\bar{f}(t)$ denotes the average fitness value over all strings in the population; P_c and P_m denote probabilities of crossover and mutation, respectively; l denotes the schema length; $\delta(H)$ denotes length of schema H measured as the distance between the first and the last fixed string positions of schema H; $o(H)$ denotes order of schema H, defined by the number of fixed string positions of schema H.

This implies that the fitness function will grow up when better offspring's are used. This fact as well as the ability of Genetic Algorithms to search efficiently huge spaces, would afford GA's as natural candidate for use in cryptanalysis.

Actually, they have been recently successfully applied to the cryptanalysis of simple substitution, transposition, and knapsack ciphers [7, 10, 11], respectively.

As DES-like systems have a large key space and it is impossible to find out the encryption key using traditional search algorithms then an evolutionary approach, based on GA's, should be examined.

2 Background of DES Cryptanalysis

In a r-round iterated block cipher such as DES, the ciphertext is computed by iteratively applying a round function g to the plaintext such that

$$C_i = g(C_i - 1, K_i), i = 1, 2, \ldots, r,$$

where C_0 is the plaintext, K_i is a round key and C_r is the ciphertext. The round function is usually based on using S boxes, arithmetic operations, and bitwise XORing [1].

A Feistel cipher with block length of $2n$ and r rounds is defined as follows.
The round function is:

$$g: \quad \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n \times \mathrm{GF}(2)^m \times \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n$$
$$g: \quad (X, Y, Z) = (Y, F(Y, Z) + X).$$

Thus, Given plaintext $P = (P^L, P^R)$ and r round keys $K_1, K_2, \ldots, K_r$, the ciphertext $C = (C^L, C^R)$ is computed in each round as follows:

1) Set $C_0^L = P^L, C_0^R = P^R$;
2) Compute $(C_i^L, R_i^L) = (C_{i-1}^R, F(C_{i-1}^R, K_i) + C_{i-1}^L)$ for $i = 1, 2, \ldots, r$;
3) Set, $C^L = C_r^L, C^R = C_r^R$, where the round key $K_i \epsilon \mathrm{GF}(2)^m$.

Definition 1. *A DES-like iterated cipher is a Feistel cipher, where F is defined as [5]:*
$F : GF(2)^m \rightarrow GF(2)^n$; $F(X, K_i) = P(f(E(X) \oplus K_i))$, *where* $K_i \epsilon GF(2)^m$; $f : GF(2)^m \rightarrow GF(2)^n, m \geq n$ *be a weak round function;* $E : GF(2)^n \rightarrow GF(2)^m$ *be an affine expansion mapping; And* $P : GF(2)^n \rightarrow GF(2)^n$ *be a permutation.*

Here a specialization of Definition 1 for DES [9] has been used as follows:
Let $M_n = m_n | m_n = b_0 b_1 \ldots b_{n-1}, b_i \epsilon \Sigma_2$ where, m_n is an n-bit string; $\Sigma 2 = 0, 1 = \mathrm{GF}(2).b_i$ is the i^{th} block and $m_n \in \mathrm{GF}(2)^n$ is a message consisting of n-bit blocks. Then, the DES cryptosystem χ_{DES} is given by:

$$\chi_{DES} =< m_{56}, m_{64}, m_{64}, T_{DES} >$$

where, m_{56} is a 56-bit key, m_{64} is a 64-bit block of either plain or cipher text and T_{DES} is the encryption/decryption transformation such that

$$t_k(m_{64}) = T_{DES}(m_{56}, m_{64}); k = 1, \cdots, 10,$$

and t_k is a composition given by:

$$t_k = IP^{-1} \circ T8 \circ T7 \circ \ldots \circ T2 \circ T1 \circ IP,$$

where IP is an initial permutation of the 64 bits and each of the $T_j, j = 1, \cdots, 10$ is a variation of an encryption theme.

In 1990 Biham and Shamir [1] have developed a type of cryptanalitic attack that can break DES-like cryptosystems, and known as differential cryptanalysis, DC. They described an n-round characteristic which allowed them to push the knowledge of the plaintext by making use of an XOR operation, to a knowledge of an intermediate round. Every round characteristic has a particular plaintext difference Ω_P, a particular XOR of the data in the n^{th} round Ω_T and a probability p^Ω (in which Ω_T when random pairs whose plaintext difference is Ω_P are used). Any pair whose plaintext difference is Ω_P and whose XOR of the data in the n^{th} round, using a particular key, is Ω_T is called a right pair with respect to that key and the n-round characteristic. Any other pair is a wrong pair. Therefore, the right pairs form a fraction p^Ω of all possible pairs.

DC attempts to find out the round key K_n. Then for two plaintext P, P^* of difference Ω_P the cryptanalyst can solve the following equation for K_n:

$$F^{-1}(C_n, K_n) \oplus F^{-1}(C_n, K_n)^{-1} = \Omega_T.$$

The solutions are candidate round keys. The method of DC can be summarized as follows [5]:

Step 1. Find a proper round characteristic with high probability.

Step 2. Uniformly select a plaintext pair P, P^* with difference Ω_P and get the encryption of this pair. Determine candidate round keys such that each of them could have caused the observed output difference. Increment a counter of each candidate round key.

Step 3. repeat Step 2 until one round key is distinguished as being counted significantly more often than other round keys. Take this key to be the actual candidate round key.

Biham and Shamir found that, from experiments on restricted versions of DES, the complexity of the attack was approximately c/p^Ω, where p^Ω is the probability of the characteristic being used, and c is a constant bounded as $2 < c < 8$. They used the signal to noise ratio S/N to measure the efficiency of DC. Assume that m pairs of chosen plaintexts are used in DC and that p^Ω is the probability of the characteristic used. Then about $m \times p^\Omega$ pairs are right pairs, each of which actually can suggest the right key value among other values. In some cases the attacker can classify pairs for the plaintext as wrong pairs using the intercepted ciphertexts. In this case such pairs are discarded and should not be used in the analysis.

Let k be the number of possible values of the key we are looking for, γ is the number of keys suggested by each

non-discarded pair of plaintexts and λ is the ratio of non-discarded pairs to all pairs. The average number of times a random key is suggested can be given by:

$$S/N = \frac{m \times \gamma \times \lambda}{k}.$$

Thus S/N determine the number of times the right key is counted over the number of times a random key is counted, i.e.,

$$S/N = \frac{k \times p^{\Omega}}{\lambda \times \gamma}.$$

A necessary condition for the success of a DC attack is $S/N > 1$ and the expected success of the attack increases with that ratio. Actually, DC attacks need a large number of right pairs that consume memory and time to suggest the encrypted key. On the other hand, GA's cannot be directly applied on the population of keys represented in the form of chromosomes. Therefore, DC is needed to determine the right pairs. This is accomplished by examining - in each round - the input difference which causes the correct output difference of each S-box. Such pairs are needed to obtain the subkeys of the key.

In what follows GA's have been exploited to calculate the key of some DES-8 cryptosystems by two methods.

1) Using a number of DC generated right pairs, which stored in order to be implemented with a proper characteristic.
2) Generating right pairs genetically.

3 The Method of Stored Right Pairs

First, the proper number of right pairs, with respect to the key, along with the proposed characteristic are stored for future processing. For each one of these right pairs there exist a number of expected keys, for every S-box. The GA is used to find out the output bits for each S-box, in the last subkey. In any iteration, the S-box output bits constitute the current chromosome of the GA. The chromosome correctness is determined by making use of the following theorem:

Theorem 1. *The chromosome correctness $C_r = \frac{n_{sr}}{n_p}$ (where n_{sr} is the number of right pairs for the current chromosome r and n_P is the total number of stored right pairs) can be successfully used as a fitness function of a genetic algorithm.*

Proof. Since $C_r = \frac{n_{sr}}{n_p}$, then it monotonically increases with the increase of n_{sr} and

$$\lim_{n_{sr} \to n_p} C_r \to 1.$$

Taking $f(S) = C_r$, for the schema S, then the fitness $f(S)$ monotonically increases with the increase of n_{sr} and

$$\lim_{n_{sr} \to n_p} f(S) = 1.$$

That is, $f(S)$ is always less than 1 except when $n_{sr} = n_P$. This guarantees that Holland's schemata theorem is satisfied and the expected number $m(S, t+1)$ of representative schema S at time $t+1$ is always greater than or equal the number $m(S, t)$ of S at the previous time t. Then $m(S, t+1) \geq m(S, t)$, which means that the number of schemata is growing up and proves this theorem □

Since the fittest chromosome is the one that satisfies the entire number of right pairs n_P, then the fittest chromosome will make C_r reach as 1.

In average, $\overline{C_r} = \frac{1}{s}\Sigma_{i=1}^{s} C_r$, where s is the chromosome length that represents the schema S. the population size should be greater than or equal s. The stored right pairs, which have been prepared by DC are used to obtain some key bits using algorithm SPCA, emphasized below.

3.1 The Algorithm SPCA (Stored Pair Cryptanalysis)

Input: number of right pairs with respect to the expected key along with the proper characteristic.
Output: some bits of that key.
Procedure:

1) Read the stored right pairs n_P;
2) For each S-box do
   a. Create an initial population in which each individual (chromosome) has number of bits equal to the key input of the current S-box.
   b. Evaluate the fitness $C_r = \frac{n_{sr}}{n_p}$ for each individual r of the population in the current generation.
   c. Apply crossover operation
   d. Apply mutation operation, if needed.
   e. Upon convergence take the fittest chromosome, which may be an expected key in the current S-box.
3) Put the correct bits in their positions in the last subkey.
4) Calculate the position of the unknown bits of the key.
5) Apply the exhaustive search on one pair to get the remainder bits of the key.

3.2 Application of SPCA to DES-8

Here the cryptanalysis DES-8 is considered. For each one of the eight S-boxes, the genetic algorithm, SPCA, is used to find out a 6-bit chromosome. The emphasize is on the first 8 rounds while the initial and final permutation are omitted, since they are not important for the attack analysis. Such analysis is based on using a number of right pairs, which were generated differentially and stored in working area. By making use of the 5-round characteristic, Figure 1, $\Omega_P = 405C000004000000_x$, with probability

$\frac{1}{10485.76}$ the analysis proceeds. Particularly, these pairs are generated and stored, by satisfying the causing condition for S2, S5, S6, S7, and S8 S-boxes for the subkey K8. Thus one can calculate correct 36 bits in K8, for S-boxes: S1, S2, S5, S6, S7, and S8 using GA. Figure 2 shows the reminder three rounds of the 5-round characteristic that complete the eight rounds of DES-8 system. The right pairs have been satisfied the condition

$$R' = h', L' = H' \oplus P(x0xx0000x)04000000x,$$

Where P is the permutation round.

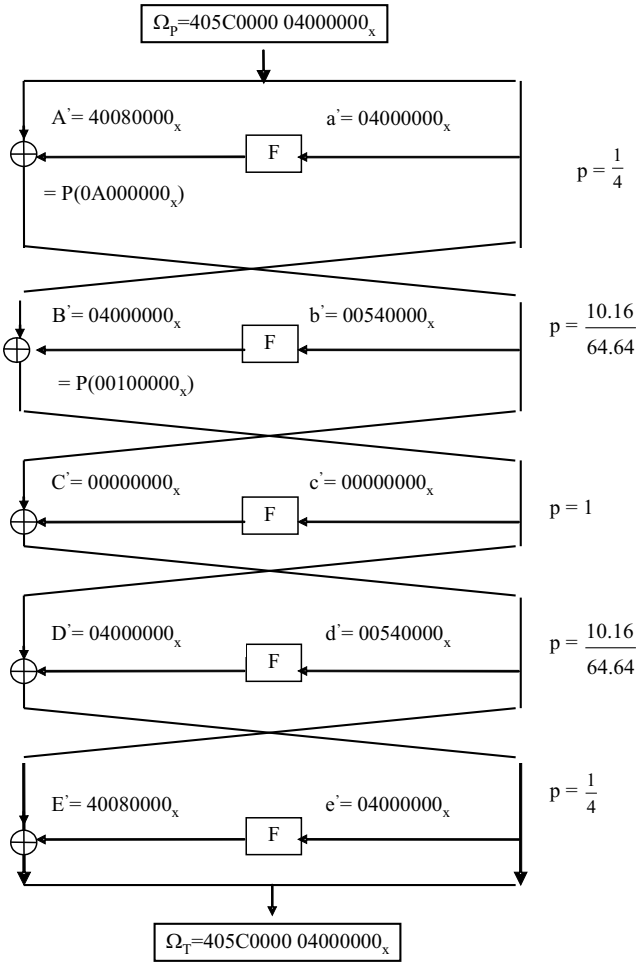

Figure 1: The 5-round characteristic with probability $\frac{1}{10485.76}$

For each right pair, $\Omega_P = P \oplus P^*$ there is a corresponding ciphertext pair T and T*, with the difference $T' = T \oplus T^*$. The right half of T' is R', and the left is L' in Figure 2. For every S-box there is a corresponding 6 bits SK in K8 satisfying the causing condition.

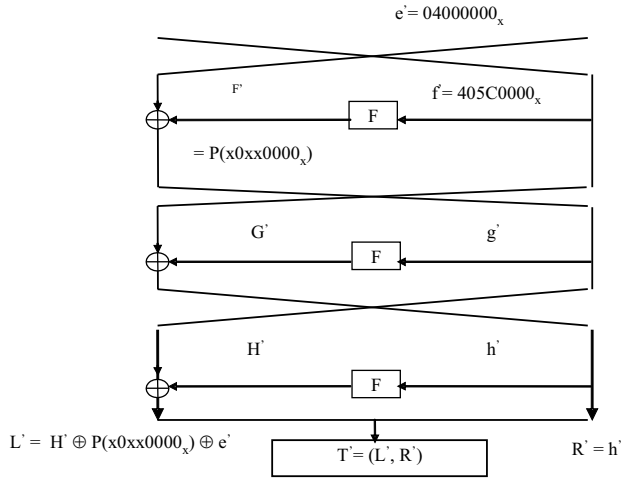

Figure 2: The last 3 rounds of DES-8 system

4 The Method of Generated Right Pairs

This method is based on a memoryless approach and it exploits the idea that without storing any pair, the fitness function can be used to generate right pairs that satisfy a proper characteristic. After generating a proper number of pairs we get a number of subkeys. According to the following theorem the most frequent subkey will be the target subkey.

Theorem 2. *Let Ω_P and Ω_T be the input and output pair of the underlying characteristic, respectively. Then any pair P, P^* such that $\Omega_P = P \oplus P^*$, enciphered by a DES-8, to $T' = T \oplus T^*$ is a right pair. Accordingly, such right pair maximizes the fitness function given by:*

$$Fitness(\Omega_T, T') = 1 - \frac{H_d(\Omega_T, T')}{n},$$

where $H_d(\Omega_T, T')$ is the Hamming distance between Ω_T and T' whereas n is the block length. This function can be successfully used as fitness function for genetic algorithms to break down DES-like systems.

Proof. If$P, P^*$is right pair then $\Omega_T = T'$. Hence $H_d(\Omega_T, T') = 0$. Also, when $H_d(\Omega_T, T')$ decreases, the expectation of the right pair increases.

Thus $\lim_{T' \to Q} Fitness(\Omega_T, T')$ and the fitness, as such, increases monotonically with the distances decrease. Then, as in Theorem 1, it means that the number of schemata is growing up with the fitness increase, and this proves the theorem □

The following algorithm, that can be used to generate the required right pairs genetically.

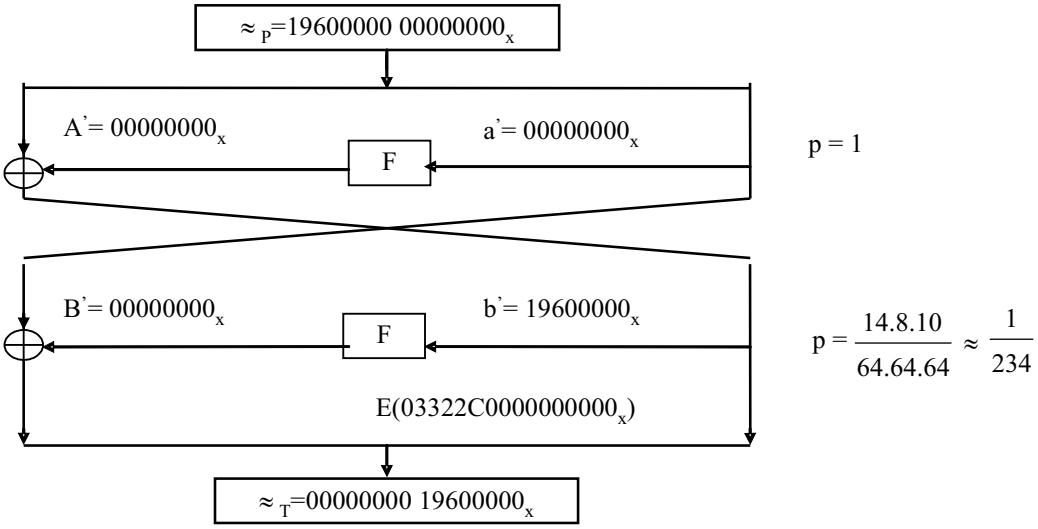

Figure 3: The 2-round characteristic with probability $\approx \frac{1}{234}$

4.1 The Algorithm GPCA (Generated Pair Cryptanalysis)

Input: difference of two plaintexts with respect to a proper characteristic.
Output: some bits of that key. Procedure:

1) Create an initial population in which each individual (chromosome) as the first plaintext.P.
2) For each chromosome do
   a. Evaluate the second plaintext $P^* = \Omega_P \oplus P$, where ΩP is the characteristic difference.
   b. Obtain the ciphertext pair $T' = T \oplus T^*$.
   c. Evaluate the Hamming distance $H_d(\Omega_T, T')$.
   d. Form a two dimensional table $\tau =< \varepsilon S, \varsigma S >$, where εS is an expected subkey and ςS is the corresponding counter for it. Set all counters to zero.
   e. Compute the fitness function $Fitness(\Omega_T, T') = 1 - \frac{H_d(\Omega_T, T')}{n}$, where n is the system block length, for each individual in the current population.
   f. If the fitness value is greater than 0.5 then test the causing condition with respect to all S-boxes.
   g. If the causing condition is satisfied then
      i. Produce from table τ subkeys for each S-box.
      ii. Generate all possible bits that may appear in the last round subkey (associated with all S-boxes) by choosing one subkey, εS for the underlying S-box from tableτ Denote such bits by σ.
      iii. For each σ, increment the corresponding counter ςS.
3) Apply crossover operation.
4) Apply mutation operation, if needed.
5) Generate the next population.
6) Repeat step ii to obtain the counter of the maximum value ς_{opt}. Such counter is associated with σopt that is probably a correct expectation for the last round subkey.

4.2 Application of GPCA to DES-8

Here the cryptanalysis DES-8 is considered. For each one of the eight S-boxes, the genetic algorithm, GPCA, is used to find out a 64-bit chromosome. By making use the 2-round characteristic, $\Omega_P = 1960000000000000_x$ with probability $\approx \frac{1}{234}$ (refer to Figure 3). Thus one can calculate correct correct 18 bits in K_8 for S-boxes S1, S2, and S3. $H_d(\Omega_T, T')$ measures Ω_P and $FP^{-1}(T')$. The causing condition is satisfied for the first three S-boxes.

5 Implementation and Performance Evaluation

These results emphasize the effect of using GA in the process of cryptanalysis. Such effect is indicated by examining the performance of SPCA and GPCA.

5.1 Effect of SPCA

The algorithm SPCA has been applied to break down DES-8 by using the chosen plaintexts/ciphertexts attack. In this case 1000 right pairs are computed "differentially"

and stored in a particular list structure. These pairs are used as the input of SPCA which has been carried out with the following parameters:

Number of right pairs = 100;
Population size = 5;
Chromosome length = 8;
Probability of crossover = 0.6;
Probability of mutation = 0.2;
Maximum generation = 100;
Seed of randomly = 0.8.

In this algorithm the value of the fitness function C_r should satisfy the condition $C_r \geq 0.15$. Otherwise the underlying S-box is by-passed and the next S-box is considered. The algorithm performance is reported in Figure 4, which shows that increasing the number of right pairs reduces the number of runs and consequently the time required to obtain the correct key.

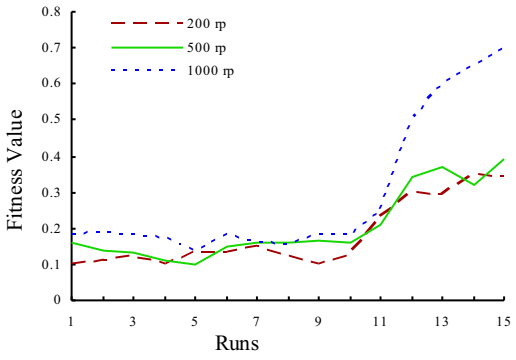

Figure 4: The computational results of SPCA when 200, 500 and 1000 right pairs are used for finding S1 genetically.

5.2 Effect of GPCA

Actually algorithm GPCA represents a "deepening" of role of GA's in cryptanalysis. In this case 1000 pairs are generated "genetically" and employed as input to the algorithm GPCA. The algorithm is executed with the following parameters:

Number of right pairs = 100;
Population size = 5;
Chromosome length = 8;
Probability of crossover = 0.6;
Probability of mutation = 0.2;
Maximum generation = 100;
Seed of randomly = 0.8.

Figure 5 shows the change of $Fitness(\Omega_T, T')$, with increasing the number of generations.

6 Conclusion

Actually the use of genetic algorithms can improve the cryptanalysis of DES-like cryptosystems. For convenience

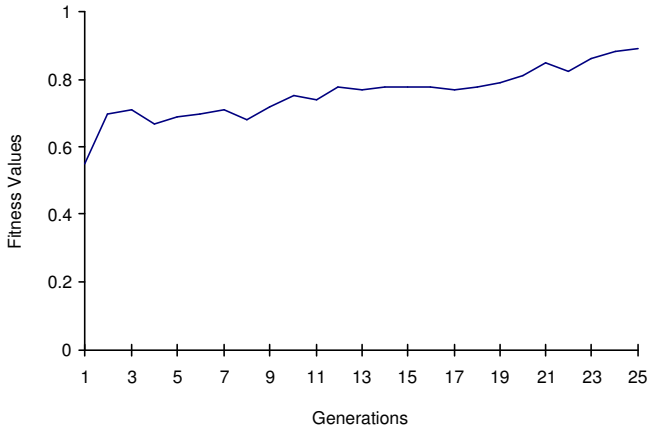

Figure 5: The results of GPCA when 200 right pairs are used for finding S1 genetically.

these algorithms are applied on DES-8. In this case, the following concluding remarks are pointed out.

1) GA's can be either combined with differential crypt-analysis methods or relied upon solely to break down block-ciphered texts.

2) The problem of using a huge number of right pairs can be solved by generating the right pairs genetically. Such generation process is carried out by exploiting the relation $Y = \Omega_P \oplus X$. thus if Ω_P is available, then Y can be obtained when X is genetically generated. The pair that satisfies the causing of the underlying S-boxes may be an expected key.

3) Despite the fact that the time complexity of GA's is $O(n^3)$, where n is the input size, computing the right pairs (needed for estimating the key) genetically is faster than differentially. This because of the fact that no right pair are stored and examined for the former technique.

4) The mutation operation is used to accelerate the break down process. In our experiments the best value of mutation rate is about 0.25. Actually, other improvements such as elitism can be added to accomplish the required cryptanalysis.

5) The performance evaluation of SPCA and GPCA indicates that genetic algorithms can successfully replace the available cryptanalysis methods of DES-like systems.

References

[1] E. Biham and A. Shamir, "Differential crypt analysis of data encryption standard," pp. 2-21, Springer-Verlag, New York, 1993.

[2] H. Feistel, "Cryptography and data security," em Scientific American, vol. 223, no. 5, PP. 15-23, May 1973.
[3] D. E. Goldberg, *Genetic Algorithms in Search*, Optimization and Machine Learning, Addison-Wesley Publishing Company Inc., Reading, Massachusetts, 1989.
[4] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich., 1975.
[5] L. R. Knudsen, *Block Ciphers- Analysis, Design and Applications*, Ph.D. Thesis, DAIMI PB-485, Aarhus University, Denmark, 1994.
[6] L. R. Knudsen, "Truncated and higher order differentials," in *International Workshop on Fast Software Encryption-Second*, LNCS 1008, pp. 196-211, Springer Verlag, 1995.
[7] R. Mtthews, "The use of genetic algorithms in cryptanalysis," *Cryptologia*, vol. 17, no. 2, pp. 187-201, 1993.
[8] National Bureau of Standards, Data encryption standard, Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., Jan. 1977.
[9] J. Seberry and J. Pieprzyk, *Cryptography: An Introduction to Computer Security*, Prentice Hall of Australia Pty Ltd, 1989.
[10] R. Spillman, M. Janssen, B. Nelson, and M. Kepner, "Use of genetic algorithms in the cryptanalysis of simple substitution cipher," *Cryptologia*, vol. 17, no. 1, pp. 31-44, 1993.
[11] R. Spillman, "Cryptanalysis of knapsack cipher using genetic algorithms," *Cryptologia*, vol. 17, no. 1, pp. 367-377, 1993.

Bahaa Hassan was born in Cairo, Egypt on May 29, 1954; He received the B. E. and M. E. degrees in electrical engineering from Zagazig University in 1978 and 1987 respectively. He received the Ph. D. degree in computer and systems engineering from Ain Shams University in 1994. He is currently a General Manager in National Defense Council, Cairo, Egypt. His current research interests include cryptography and their applications, smart cards applications and computer and network security.

Bayoumi Ibrahim Bayoumi was corn in El-Sharkia, Egypt on August 14, 1946. He received his B.Sc. from Faculty of Science, Ain Shams University, Egypt, June, 1967. M.Sc. from Faculty of Science, Ain Shams University, Egypt, June, 1970.Ph.D. from Leningrad state University, USSR, September, 1974. Now he is Professor of Mathematics, Faculty of Science, Ain Shams University, Egypt.

Fathy Saad Holail was born in Cairo, Egypt on November 14, 1946. He received the B. Sc., M. Sc., and Ph.D. degrees in mathematics from Cairo university, in 1970,1983 and 1985 respectively. During 1991 and 1992 he was a visiting scholar at Tsujii Laboratory of Information system security, Department of electrical and electronic Engineering, Tokyo Institute of Technology, Japan. His Current job is Head of C. R. Division, Research Development Center, National Defense Council, Cairo, Egypt. His current research interests includes design and implementations of encryption algorithms also cryptanalysis of cryptographic systems. He is a member of IACR, Mathematical Egyptian society and language Engineering society at Egypt.

Hassan Mohammed Hassan Hussein was born in Sharkia Egypt on May 29, 1951. He received the B. Sc., M. Sc., and Ph. D. degrees in pure mathematics from university of Cairo, Zagazig, and Ain Shams in 1974, 1987, and 2003 respectively. His Current job is one of C. R. Division, Research Development Center, National Defense Council, Cairo, Egypt. His Current research interests includes Design and Implementation of encryption algorithms also cryptanalysis of cryptographic systems.

M. Zaki is the professor of software engineering, Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University at Cairo. He received his B.Sc. and M.Sc. degrees in electrical engineering from Cairo University in 1968 and 1973 respectively. He received his Ph. D. degrees in computer engineering from Warsaw Technical University, Poland in 1977. His fields of interest include artificial intelligence, soft computing, and distributed system. (azhar@mailer.scu.eun.eg)