Artificial Neural Networks for Cryptanalysis of DES

Bahubali Akiwate
Department of Computer Science and Engineering
KLECET, Chikodi, Karnataka, India

Veena Desai
Department of Electronics and Communication Engineering
GIT, Belgaum, Karnataka, India

Abstract- Cryptanalysis of block ciphers has witnessed many changes during the last two decades and many new cryptanalytic techniques were introduced. On the other side, Neural Networks have shown great deal of reliability and applicability as its applications spread in different areas. Furthermore the efficient ways introduced to implement its hardware and software became even faster and easier. The enhancement of neural networks using genetic algorithms has proved to be a good method for developing better and easily trained neural networks.
A new cryptanalytic technique that is based on neural networks evolved by genetic algorithms is introduced in this work. This cryptanalysis technique is considered to be a known-plaintext attack. The proposed technique was implemented in software and applied to the Data Encryption Standard (DES). The DES as a complex block cipher with block size of 64 bits. The software implementation of the proposed cryptanalytic system has shown noticeable privileges over the other cryptanalytic techniques, in terms of time, computational abilities required, and the number of ciphertext-plaintext pairs needed for cryptanalysis.
This paper concern with the learning capabilities of neural networks and its application in cryptanalysis.

Keywords – Cryptanalysis,Artificial Neural Networks.

I. INTRODUCTION

Cryptography is a method of storing and transmitting data in a form that only those it is intended for can read and process. It is a science of protecting information by encoding it into an unreadable format. Cryptography is an effective way of protecting sensitive information as it is stored on media or transmitted through network communication paths. Artificial neural networks have been developed as generalizations of mathematical models of biological nervous systems. A first wave of interest in neural networks (also known as connectionist models or parallel distributed processing) emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943. Artificial neural networks are non-linear mapping structures based on the function of human brain. They are powerful tools for modeling, especially when the underlying data relationship is unknown [18]. Artificial neural networks can identify and learn correlated patterns between input data sets and corresponding target values. After training they can be used to predict the outcome of new independent input data. Neural networks exhibit nonlinearity property which is desired in cryptography.

The ability of neural networks to explore the solution space could also be used in the field of Cryptanalysis. It also could be possible to generate new kind of attacks on existing algorithms based on the idea that any function could be reproduced by a neural network, so it will be possible to find the exact solution, at least theoretically breaking the algorithm. In this paper, we started from the motivation for block cipher design and implemented DES algorithm in MATLAB. They possess the most popular design criteria for block ciphers. The motivation for this proposal is the ability of neural networks to perform complex mapping functions from one domain to another.

Therefore the paper aims at analyzing the nonlinearity property of neural networks by performing cryptanalysis on DES.

II. BACKGROUND

To achieve the goal a variety of security algorithms mechanisms have been designed and these mechanism are collectively known as ciphers, which are used for encryption and decryption and are difficult to break. In recent years modern ciphers apply key(s) for controlling encryption and decryption and messages are decrypted by the

receiver only if the key matches the encryption key. Research on cryptographic mechanism had proved that symmetric algorithm is quicker to execute on a computer than asymmetric algorithm because of the use of one key for both operations [11]. However in practice both keys (algorithms) are used together to encrypt and decrypt.
Data Encryption Standard (DES) which uses 64 bits block and key was the main encryption standard for applications developments when it first began in 1977. However this algorithm is susceptible to brute force attacks so more block ciphers had been developed to enhance security of data namely Advanced Encryption Standard (AES), MARS, RC6, Serpent, Twofish. Among these algorithm AES had been chosen as the standard algorithm for internet and hardware security because the algorithm is user friendly to computer computational power. But one thing of great interest is that these algorithms apply Key schedule which is an algorithm that, given the key, calculates the sub-keys for each round. Whenever a key generation is weak it becomes easy for differential and linear cryptanalysis that's why research had already been done on the application of classical Artificial Neural Networks (ANN) to address these security concerns. However, Spiking Neural Networks (SNN) which uses simulated neurons for each basic unit of operations and timing of outputs are encoded in 1's and 0's, with 1 meaning it is occurring and 0 being it is not happening. SNN works by using a sign function as activation function and then project the weights and filter co-efficient to be adjusted so allowing for additional independence than ANN.
To rectify the private key issues and applied it in SNN, a prototype symmetric cipher block which had a block length of 8 bits but which in practice can be expanded to either 64 bits, 128 bits or to any variable desired length. Simple-Data Encryption Standard (S-DES), the block cipher chosen for the research as it is the short version of DES and recognized for most internet security purposes, and moreover the algorithm makes it possible for quicker changes in encryption keys and network level encryption at a much higher speed without the concern of factorizations.

III. IMPLEMENTATION

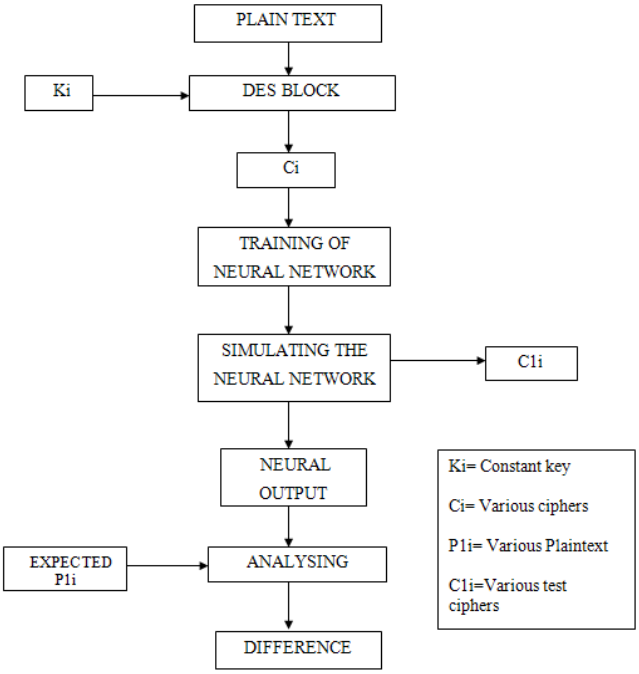

Figure 1. Implementation of neural network

We considered DES using neural networks. We considering feedforward neural network. For DES implementation here considering neural network of 4 layers. The neural network input layers consists of 64 neurons, 2 intermediate layers, each consisting of 100 neurons, output layer consisting of 64 neurons.

Using DES block with a variable plain text and constant key different ciphers are generated, which are then fed as a input to the training neural block along with the variable plaintexts. This neural network is then simulated with different set of cipher and a neural output is obtained as shown in figure 1. This neural output is compared with expected plaintexts for the cryptanalysis purpose.

Training and testing neural networks

The best training procedure is to compile a wide range of examples (for more complex problems, more examples are required), which exhibit all the different characteristics of the problem. To create a robust and reliable network, in some cases, some noise or other randomness is added to the training data to get the network familiarized with noise and natural variability in real data [4].

Poor training data inevitably leads to an unreliable and unpredictable network. Usually, the network is trained for a prefixed number of epochs or when the output error decreases below a particular error threshold. Special care is to be taken not to overtrain the network. By overtraining, the network may become too adapted in learning the samples from the training set, and thus may be unable to accurately classify samples outside of the training set. Figure 2 illustrates the classification results of an overtrained network. The task is to correctly classify two patterns X and Y. The test patterns were not shown during the training phase. As shown in Figure 2 (left side), each class of test data has been classified correctly, even though they were not seen during training. The trained network is said to have good generalization performance.

Figure 2 (right side) illustrates some misclassification of the test data. The network initially learns to detect the global features of the input and, as a consequence, generalizes very well. But after prolonged training, the network starts to recognize individual input/output pairs rather than settling for weights that generally describe the mapping for the whole training set.

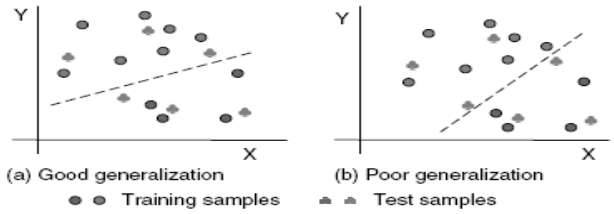

Figure 2. Illustration of generalization performance

Choosing the number of neurons

The number of hidden neurons affects how well the network is able to separate the data. A large number of hidden neurons will ensure correct learning, and the network is able to correctly predict the data it has been trained on, but its performance on new data, its ability to generalize, is compromised. With too few hidden neurons, the network may be unable to learn the relationships amongst the data and the error will fail to fall below an acceptable level. Thus, selection of the number of hidden neurons is a crucial decision.

Choosing the initial weights

The learning algorithm uses a steepest descent technique, which rolls straight downhill in weight space until the first valley is reached. This makes the choice of initial starting point in the multidimensional weight space critical. However, there are no recommended rules for this selection except trying several different starting weight values to see if the network results are improved.

Choosing the learning rate

Learning rate effectively controls the size of the step that is taken in multidimensional weight space when each weight is modified. If the selected learning rate is too large, then the local minimum may be overstepped constantly, resulting in oscillations and slow convergence to the lower error state. If the learning rate is too low, the number of iterations required may be too large, resulting in slow performance.

IV.RESULTS

In this paper, Data Encryption Standard algorithm is used to generate ciphertext, with plaintext as an input along with a constant key. Here a file consisting set of characters is read by programmer and another file equivalent to this, which is in binary form is to be written. Here the file named 'ppp.txt' which consists a set of characters as shown in below figure 3 is read and an another file named 'bininput.txt' is written, which is binary equivalent to the file named 'ppp.txt', as shown in figure 4.

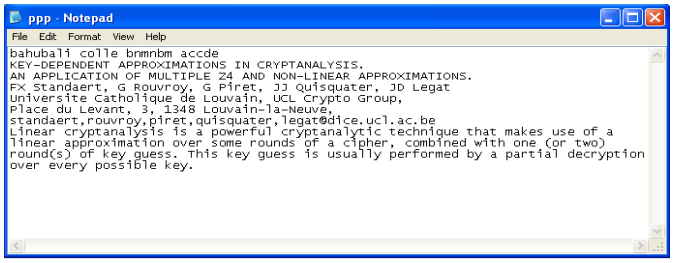

Figure 3. File consisting a set of characters

There can be 8 characters can be taken at a time from the text file while converting it into binary equivalent file. A single character can generate 8 binary bits and hence 8 characters combining to form 64 bits. While doing an encryption and decryption processes these 64 bits are used at a time along with a constant key of 64 bits as a user entry. Here in this project the constant key E326D1475398000(in hexadecimal) is used for both encryption and decryption.

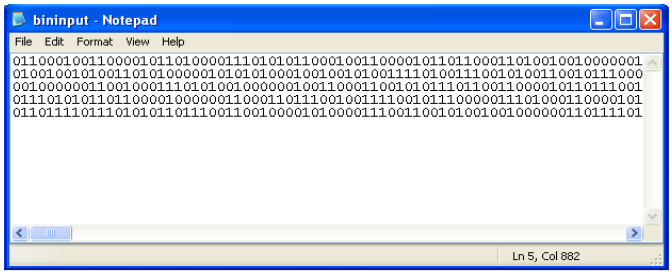

Figure 4. Binary equivalent file

After writing binary equivalent file named 'bininput.txt', it can be read again and one more file named 'ccc.txt' can be written, which consisting the encrypted binary bits known as ciphertext bits, which uses the encryption function along with plaintext bits and key, as shown in below figure 5. This file consists of a same number of bits as ‘bininput.txt’ file.

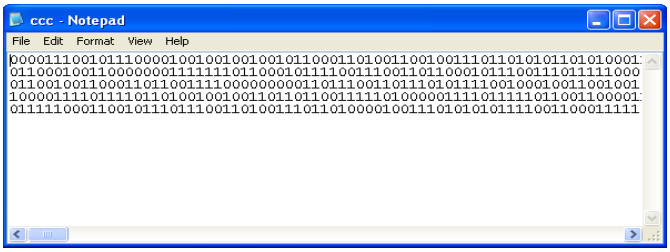

Figure 5. Ciphertext file

Then a file named ‘xxx.txt’ can be written by reading the file 'ccc.txt' as a decrypted file, which consists of decrypted binary bits. This uses the decryption function along with ciphertext bits and key. This file consists same bits as a file named ‘bininput.txt’, which is shown in figure 6.

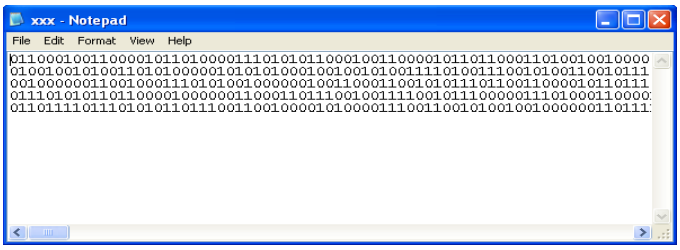

Figure 6. Decrypted file

The plaintext and ciphertext files were used for training neural network program by taking bits from these files, as inputs. Here I used about half of the bits among all bits produced in these files.
Among these chosen ciphertext bits and plaintext bits some samples were taken for simulation. For different runs of this training neural network program, different samples were used. There should be a corresponding ciphertexts and plaintexts were chosen as samples for each run. Here the variables p1 and t1 represents ciphertext and plaintext respectively, as an input and target values. Some examples are as shown below.

Example 1((Here p1 and t1 matrices are in dimension 8×64)

p1=[1 0 1 0 1 1 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 1
1 0 1 0 ;
0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0
0 1;
0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0
0 1 ;
0 0 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0
0 0 ;
1 0 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 1 1 0
0 1 ;
0 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1
0 1;
0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0
0 0 ;
0 1 0 0 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 1 0 1 1 1 1 0 0 0 1 0 1 0 1 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 1 1
1 0];

t1=[0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0
1 1 1 0 ;
0 1 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1
0 1 ;
0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0 0 1 1 0 1 0
0 1 ;
0 1 1 1 0 0 0 1 0 1 1 1 0 1 0 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 1
0 0 ;
0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0
0 0 ;
0 1 0 1 0 1 0 1 0 1 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 1 0 0 1 0 1 1 1 0 0
0 0 ;
0 1 1 1 0 1 0 0 0 1 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 1 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 0
0 0 ;
0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1
0 1] ;

After simulation, the network outputs were generated along with following graph shown in figure 7, here the value of mean calculated is found to be as 11.7500.Finally, the performance error can be calculated by using the equation given below:

Performance Error =(mean/64)×100

Hence the performance error established here is 18.3594. Activation functions used here are tansig, logsig, purelin and satlin respectively. This is one of the combination of activation functions used in this neural network training program in which the performance error is more.

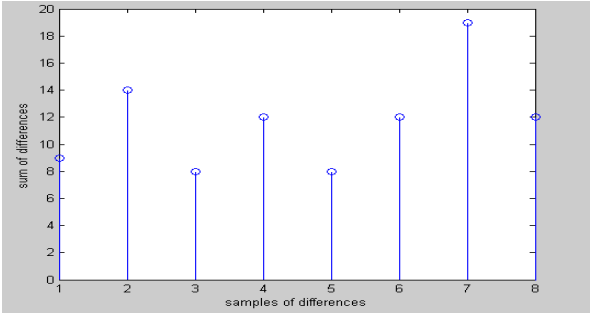

Figure 7. Representing sum of differences

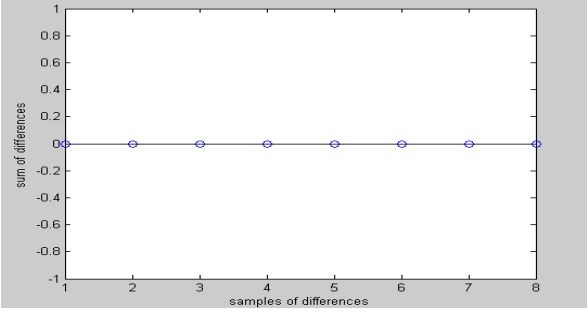

Figure 8. Representing sum of differences

Let us consider the figure 8, here the value of mean calculated is found to be as 0. Hence the performance error established here is 0. Activation functions used here are logsig, tansig, satlin and purelin respectively. This is one of the combination of activation functions used in this neural network training program in which the performance error is less.

V. CONCLUSION

As we move toward a society where automated information resources are increasingly shared, 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 DES algorithm has been a successful effort in the early development of security mechanisms. It is the most widely analyzed, tested, and used cryptoalgorithm and it will continue to be for some time yet to come. But perhaps the most important contribution of the DES is that it has led us to other security considerations, beyond the algorithm itself that must be made in order to have secure computer systems and networks.

In this project, I presented an idea of application of artificial neural networks for Cryptanalysis. I performed efficient cryptanalysis on the DES, by using plaintexts and key. Here, the performance error established is used to analyze data. And hence efficiency can be achieved. The efficiency can be increased further by increasing the number of samples used for training the neural network and correspondingly adjusting the weights and biases of the neurons in each layer.

A lot of future works on neural network weights selection and adaption (training) is still to be done. Implementation possibilities especially by hardware are still an open area. Cryptanalysis of the proposed cipher and enhancing its security are another area of future work.There may be different types of neural networks are used for cryptanalysis.

REFERENCES

[1] William Stallings, "Cryptography and Network Security**",** 3rd edition, Pearson Education (Asia) Pvt.Ltd./Prentice Hall of India, 2003.
[2] Dr. K. S. Ooi and Brain Chin Vito, "Cryptanalysis of S-DES", 2002.
[3] Carlos Gershenson, "Artificial Neural Networks for Beginners"
[4] Ajith Abraham, "Artificial Neural Networks "

[5] Fiona Nielsen, "Neural Networks – algorithms and applications"
[6] Rudra Pratap, "Getting started with Mat Lab 6.0"
[7] K.V .Srinivas Rao , M. Rama Krishna and D. Bujji Babu, "Cryptanalysis of a feistel Type Block Cipher by Feed Forward Neural Network Using Right Sigmoidal Signals",2009
[8] Howard M. Heys ,"A Tutorial on Linear and Differential Cryptanalysis" Memorial University of Newfoundland, email: howard@engr.mun.ca
[9] Anders Krogh, "What are artificial neural networks? "
[10] Simon Haykin , "Feedforward Neural Networks : An Introduction"
[11] Craig Smith ,"Basic Cryptanalysis Techniques" , November 17th, 2001.
[12] Khaled M. Alallayah , Waiel F. Abd El-Wahed, Mohamed Amin and Alaa H. Alhamami , "Attack of Against Simplified Data Encryption Standard Cipher System Using Neural Networks", Journal of Computer Science 6 (1): 29-35, 2010
[13] RC Chakraborty , "Fundamentals of Neural Networks", June 01 ,2010
[14] Subariah Ibrahim,Mohd Aizaini Maarof, "A Review On Biological Inspired Computation In Cryptology", Universiti Teknologi Malaysia, Skudai 81300, Johore, Malaysia.
[15] www.infosecuritymag.com/articles/july01/columns_logoff.shtml
[16] Kaisa Nyberg , "Methods for Symmetric Key Cryptography and Cryptanalysis", 2009.
[17] S N Sivanandam, S Sumathi, S N Deepa, "Introduction To Neural Networks Using MATLAB 6.0",2006
[18] Girish Kumar Jha, "Artificial Neural Networks And Its Applications", I.A.R.I, New Delhi, girish_iasri@rediffmail.com
[19] Reaffirmed,"Data Encryption Standard (DES)", U.S. Department Of Commerce/National Institute of Standards and Technology, 1999 October 25
[20] Biham, E. and Shamir, A.:"A Differential Cryptanalysis of the Data Encryption Standard", Springer Berlin Heidelberg New York, 1993.
[21] H. Feistel, "Cryptography and computer privacy," Sci. Amer., vol. 228, no. 5,pp. 15-23, May 1973.
[22] W. Diffie, M. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," Computer, pp. 74-78, June 1977.
[23] Martin Albrecht and Carlos Cid, "Algebraic Techniques in Differential Cryptanalysis" Information Security Group, Royal Holloway, University of London Egham, Surrey TW20 0EX, United Kingdom