Applying Neural Networks for Simplified Data Encryption Standard (SDES) Cipher System Cryptanalysis

Khaled Alallayah[1], Mohamed Amin[2], Waiel AbdElwahed[3], and Alaa Alhamami[4]
[1]Department of Computer Science, IBB University, Yemen
[2]Department of Mathematical and Computer Science, El-Menoufia University, Egypt
[3]Department of Operation Research and Decision, El-Menoufia University, Egypt
[4]Faculty of Computing Studies, Amman Arab University for Graduate Studies, Jordan

Abstract: *The problem in cryptanalysis can be described as an unknown and the neural networks are ideal tools for black-box system identification. In this paper, a mathematical black-box model is developed and system identification techniques are combined with adaptive system techniques, to construct the Neuro-Identifier. The Neuro-Identifier is discussed as a black-box model to attack the target cipher systems. In this paper, a new addition in cryptography has been presented and the methods of block Simplified DES (SDES) crypto systems are discussed. The constructing of Neuro-Identifier mode is to achieve two objectives: The first is to emulator construction Neuro-model for the target cipher system, while the second is to (cryptanalysis) determine the key from given plaintext-ciphertext pair.*

Keywords: *System Identification, artificial neural network, emulation, SDES, cryptanalysis, neuro-identifier.*

Received July 19, 2009; accepted May 20, 2010

1. Introduction

Block cipher systems belong to symmetric cryptographic systems, where the same key is used for encryption and decryption process. The major difference between block ciphers and other symmetric cryptographic systems are that; block ciphers are characterized by the fact that the decipherment of a bit of data depends not only on the key but also on some of the other bits of data. The principles behind the design of most block ciphers are the concepts of diffusion and confusion. The idea of confusion is to make the relation between a cryptogram and the corresponding key a complex one. This aims to make it difficult for the statistics to point out the key as having comes from any particular area of the key space. The concept of diffusion is to spread the statistics of message into statistical structure, which involves long combinations of the letters in the cryptogram, and hence whitening all the statistical feature of the neutral language.

In this paper, a brief discussion of block ciphers background, and techniques are presented. DES cipher is chosen as a case study of block cipher because, it was (and still) the challenge of most of the researchers over the last 25 years. Security of cryptographic systems is directly related to the difficulty associated with inverting encryption transformations of the system. The protection afforded by the encryption procedure can be evaluated by the uncertainty facing an opponent in determining the permissible keys [6]. The cryptanalysis problem can be described as an identification problem, and the goal of the cryptography is to build a cryptographic system that is hard to identify [9, 10]. System identification is concerned with inferring models from observation and studying system behaviour and properties. System identification deals with the problem of building mathematical models of dynamical systems based on observed data from the system [10, 13].

Artificial Neural Networks (ANN) are simplified models of the central nervous system. They are networks of highly interconnected neural computing elements that have the ability to respond to input stimuli. Among the capabilities of ANN, are their ability to learn adaptively from dynamic environments to establish a generalized solution through approximation of the underlying mapping between input and output [7, 14, 16]. Neural networks can be regarded as a black-box that transforms an input vector of m-dimensional space to an output vector in n-dimensional space. This makes them ideal tools for black-box system identification [15, 23].

In this paper, a simplified version of the DES block cipher algorithm has been implemented. Naturally enough, it is called SDES, and it is designed to have the features of the DES algorithm but scaled down so it is more tractable to understand. A survey of previous cryptographic work especially for DES is presented.

The proposed (Emulation and Cryptanalysis) models using Neuro-Identifier (NID) against SDES is described in detail with the results obtained during the paper.

2. System Identification

There are two approaches for system identification [10, 13] depending on the available information, which describe the behaviour of the system. The first approach is the State-Space approach (internal description), which describes the internal state of the system, and is used whenever the system dynamical equations are available. The second approach is the Black-Box approach (input-output description) which is used when no information is available about the system except its input and output [17]. Figure 1 illustrates an unknown system with *xm* input signals and *yn* output signals. The central concept in identification problems is identifiably [13]. The problem is whether the identification procedure will yield a unique value of the parameter (q), and/or whether the resulting model (M) is equal to the true system, i.e., a model structure is globally identified at:

$$(\Theta\ *)\ if:\ \ M\,(\theta) = M\,(\theta\ *), \quad \theta \in DM => \theta = \theta\ * \tag{1}$$

Where *M* is a model structure; θ is a parameter vector, ranging over a set of values *DM* [23].

3. Input-Output Descriptions

The input-output description of a system gives a mathematical relationship between the input and output of the system. In developing this description, the knowledge of the internal structure of a system may be assumed to be unavailable; the only access to the system is by means of the input and output terminals [1, 10]. Under this assumption, a system may be considered a Black-Box as shown in Figure 1. Clearly what one can do to a black box is to apply inputs and measure their corresponding outputs, and then try to abstract key properties of the system from these input-output pairs. An input-output model assumes that the new system output can be predicted by the past inputs and outputs of the system [4, 17].

A Black-Box model of system identification assumes no prior knowledge about the system except it's input and output, i.e., no matter what analysis is used, it always lead to the same input-output description. Moreover, a Black-Box model allows finite-dimensional identification techniques to be applied, which may require in nonlinear system identification. In developing the input-output description, before an input is applied, the system must be assumed to be relaxed or at rest, and that the output is excited solely and uniquely by the input applied thereafter and the system is said to be causal if the output of the system at time k does not depend on the input applied after time *k* [2]. The system can be described as follows:

$$Y\,(k) = H\,x \tag{2}$$

Where *H* is some function that specifies uniquely the output *y* in terms of the input *x* of the system. Although the subject of system identification is well developed for linear systems, the same is not true for the nonlinear case. However, linearization of nonlinear systems can be obtained by several methods, among them is the approximate linearization technique for nonlinear systems [10, 17, 19]. For Single-Input Single-Output (SISO), the input-output model identification problem is to devise a mathematical model which, when excited with the input sequence [*x*(*k*), *k=1,2,…, m*], will produce an estimated output [*y*(*k*), *k=1,2,…, n*], such that:

$$y(k)=f(y(k-1),y(k-2),\dots,y(k-n),x(k-1),x(k-2),\dots,x(k-m) \tag{3}$$

Where [*x* (*k*), *y* (*k*)] representing the input-output pairs of the system at time *k*, *n*, and m are positive integers representing the number of past outputs and the number of past inputs respectively. *F* is a static nonlinear function which maps the past inputs and outputs to a new output. *f* is called describing function. That means; for any discrete-time, unknown nonlinear system there would be suitable positive integers (*m* and *n*) and a multidimensional mapping *f* (.) in such a way that the system output at a given instant could be approximated by equation 3. If a system is linear *f* is a linear function, and equation 3 can be rewritten as [7, 9, 10]:

$$Y\,(k) = \alpha_1\, y\,(k\text{-}1)+\alpha_2\, y(k\text{-}2),\ \dots+\ \alpha_n\, y(k\text{-}n)\ +\ \beta_1\, x\,(k\text{-}1)\ +\beta_2\, x\,(k\text{-}2)\ \dots+\ \beta_n\, x\,(k\text{-}m) \tag{4}$$

Where α_i (*I=1,2,…,n*) and β_1 (*I=1,2,…,m*) are real constants. Equation 4 can be rewritten in matrix notation:

$$y(k) = \sum_{i=0}^{n} \alpha_i\, k(y-1) + \sum_{j=0}^{m} \beta_j\, k(x-j) \tag{5}$$

For Multi-Input Multi-Output (MIMO), *y*(*k*) and *x*(*k*) are of dimensions m and p respectively, equation 5 can be rewritten as [10]:

$$y(k) = \sum_{i=0}^{n} A_i\, k(y-1) + \sum_{j=0}^{m} B_j\, k(x-j) \tag{6}$$

Where A_i and B_j an (*m*×*m*) and (*m*×*p*) matrices, respectively.

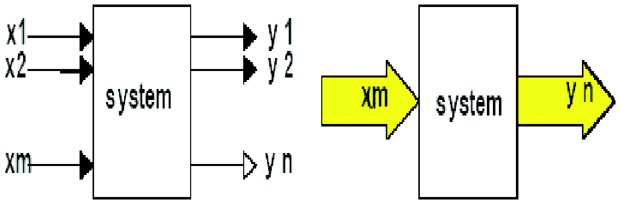

Figure 1. System with m inputs and n outputs.

4. Cryptographic System

An encryption algorithm is a single parameter family of invertible transformations (mappings) of the message space (M) into the cryptogram (ciphertext) space (C) using finite length key k from key space (K). See a reversible encryption algorithm [18, 19] in equation 2:

$$E_k\colon M \rightarrow C$$
$$\text{Such that: } E_k\,(m) = c,\ k \in K,\ m \in M,\ c \in C \tag{7}$$

An inverse decryption algorithm:

$$D_k = E^{-1}{}_k : D_k\colon C \rightarrow M \tag{8}$$
$$\textit{Such that: } \quad D_k(c) = D\,k\,[E_k\,(m)] = m$$

The keys should uniquely define the enciphered message:

$$i.e.,\ E_{k1}\,(m) \neq E_{k2}\,(m) \quad if\ k_1 \neq k_2 \tag{9}$$

According to the previous discussion of the properties of the system, and the definition of a cryptographic system, it might be concluded that: a cryptographic system is, relaxed, causal, time invariant, and nonlinear system.

5. Neuro-Identifier

Identification of a system consists of finding a model relationship. Consider the system described in equation 3. Identification then consists of determining the system orders and approximation of the unknown function by neural network model using a set of input and output data [5, 11, 12].

The procedure begins with the choice of neural model which is defined by its architecture and an associated learning algorithm. This choice can be made through trial and error. Once the neural model is chosen, and system input-output data are available, learning can begin. Different structures are trained and compared using learning set and simulation set of data, and a criterion (error goal) [21, 24]. The optimal structure then, is the one having the fewest units (neurons) for which the criterion is met. Neuro-Identifiers (NID) are basically Multi-Layer Feed-Forward (MLFF). Artificial neural networks with an input layer (buffer layer), a single or multiple nonlinear hidden layer with biases, and a linear/or nonlinear output layer [17, 22]. The results of research have shown that linear identifiers are not capable of identifying nonlinear systems. Hybrid identifiers can identify simple nonlinear systems but not complex ones [8, 20, 22]. Figure 2 illustrates the structure of the multi-layer feed-forward neural network identifier NID, with two nonlinear hidden layers, which is used in this research. The size of the neural network is crucial in designing the whole structure. There is no mathematical formulation to calculate the optimal size of such networks. However, with many free units the NID will learn faster, avoid local minima, and exhibit a better generalization performance [7, 23]. The essential constraint on increasing the size of hidden layers is the limitation of the hardware architecture used in the experimental work.

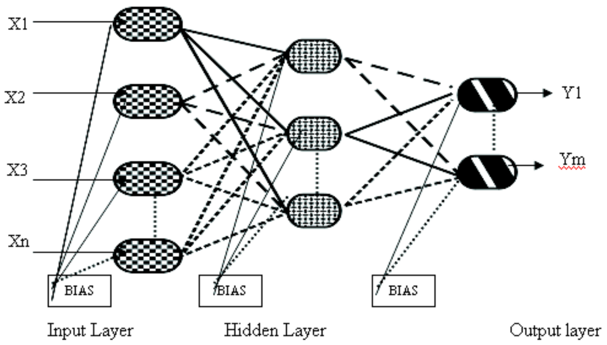

Figure 2. Multi-layer feed forward neuro-identifier architecture.

5.1. Training Algorithm

The Levenberg-Marquardt (LM) algorithm is MLFF, the most idely used optimization algorithm. It outperforms simple gradient descent and other conjugate gradient methods in a wide variety of problems. This document aims to provide an intuitive explanation for this algorithm. The LM algorithm is first shown to be a blend of vanilla gradient descent and Gauss-Newton iteration. Subsequently, another perspective on the algorithm is provided by considering it as a trust-region method [10, 11].

```
Algorithm (LM)
1. Initialize network (Weights and Biases)
2. For each training pair 3-7 until performance criteria
3. Sums weighted input and apply activation function to compute output.
     $h_0i = \sum I=1\ Xi\ Wij + bi$
     $hi = f(h_0j)$
4. Compute output of network
     $yy = bp + \sum i=1\ hi\ Wpi$
     $y = f(yy)$
5. Calculate error term
     $\delta = (y-yd)$
6. Calculate correction term
     $Wb = [w1b1\ w2b2\ \ldots\ wpbp\ ]$
     $\Delta Wb = (J^T.J + \eta I)-1.\ (-J^T.\delta)$
7. Update biases and weights
     $Wij\ (new) = wij\ (old) + \Delta\ Wb$
8. End
```

5.2. Using NID in Cryptanalysis

Cryptographic systems are a 2-input, 1-output systems, it takes a plaintext character (or bit /block of bits), and a key character to produce a ciphertext character. Hence a 2-neurons input layer is used to present the training data to the identifier, while a single neuron output layer is used. The described neural network identifier was used to identify cryptographic systems in two approaches with the following objectives:

1. Emulation Approach: Construct of a neuro-model for the target unknown cipher system [9].

  a. Encryption Cipher: Input data: TP, TK. Desired output data: TC.
  b. Decryption Cipher: Input data: TC, TK. Desired output data: TP.

2. Cryptanalysis Approach: Input data: TP, TC. Desired output data: TK.

The first objective is to construct a neuro-model which imitates the internal (transfer) function of the cryptographic system (hardware or software). After training and on convergence, the constructed model will resemble the target system completely. The construction of such a model will be useful in studying the behaviour of the unknown system and it can be used as a real system in encryption and decryption in cases where the real system cannot be.

The aim of the second objective is to obtain clearly a pure cryptanalysis target (total break). This could be done by introducing plaintext-cipher text as input to the system, which yields the key as output. The training data is built using the target cipher system algorithm by applying selected input signals (characters or bits) and collecting the output response of the system. The resulting data are split into two groups; the first group is used to train the neural network, while the second group is used to test (simulate) the trained network.

6. Block Ciphers

IBM initiated a cryptographic research concentrating on nonlinear block ciphers in the late 1960's, and has produced several important cryptographic systems. In January 1977, the National Bureau of Standard (NBS) adopted one of these as the national data encryption standard (DES). IBM systems have their roots in Shannon's brilliant 1949 paper connecting cryptography with information theory [2]. Shannon suggested using product ciphers to build a strong system out of simple, individually weak components.

He suggested using products of the form $B_1M, b_2M..., B_nM$, where M is a mixing transformation, and Bi is simple cryptographic transformations. High-speed electronic circuitry allows the product system to be implemented almost as economically as single BM pairs. The data are encrypted in number of "rounds" (iterations) each consisting of a single pair B_iM and each using the same hardware. The same key is used in encryption and decryption process. The fundamental building block of DES is a single combination of substitution followed by permutation (diffusion and confusion) on the text based on the key. This is known as a round. DES has 16 rounds; i.e., it applies the same combination of substitution and permutation 16 times [21]. The output of the i th round become the input to the (*i+1*) round. Block ciphers probably of the most important cryptographic primitives.

Although they are used for many different purposes, their essential goal is to ensure confidentiality. This paper is concerned by their quantitative security, that is, by measurable attributes that reflect their ability to guarantee this confidentiality. Well know results. Starting with Shannon's Theory of Secrecy, we move to practical implications for block ciphers, recall the main schemes on which nowadays block ciphers are based, and introduce the Luby-Rackoff security model. We describe distinguishing attacks and key-recovery attacks against block ciphers [3].

The system uses a transformation of the bits within a block for the fixed mixing transformation T, and substitution on four bits groups of the block for the simple cryptographic transformation Si. Any k-bit S-box can be implemented as 2k word memory with k-bit words. The Neuro-Identifier (NID), as described above, has been used in this research in block cryptosystem identification, as a black-box model.

The objective of the attack, is to determine the key from the given plaintext-ciphertext pair. Black-box attack has been applied to SDES. SDES encryption takes a 10 bit raw key (from which two 8 bit keys are generated as described in the handout) and encrypts an 8 bit plaintext to produce an 8 bit ciphertext. Implement the SDES algorithm in a class called SDES.

Definitions:

K = (k0k1……k9) where k1 ∈ {0,1} Key
M = (m0m1……m7) where m1 ∈ {0,1} Message
P4 = (1,3,2,0) Shifting Sequence = (1,2)
P8 = (5,2,6,3,7,4,9,8) P10 = (2,4,1,6,3,9,0,8,7,5)
IP = (1,5,2,0,3,7,4,6) IP-1 = (3,0,2,4,6,1,7,5)

$$SB0 = \begin{pmatrix} 1\,0\,3\,2 \\ 3\,2\,1\,0 \\ 0\,2\,1\,3 \\ 3\,1\,3\,2 \end{pmatrix} \qquad SB1 = \begin{pmatrix} 0\,1\,2\,3 \\ 2\,0\,1\,3 \\ 3\,0\,1\,0 \\ 2\,1\,0\,3 \end{pmatrix}$$

Algorithm (Sdes):

```
1    P10 (K)       ⇒   s = (s0s1s2s3s4) (s5s6s7s8s9)
2.   Shift (s, 1)  ⇒   t =( s1s2s3s4s0s6s7s8s9s5)
3.   P8 (t)        ⇒   k1 = (t5t2t6t3t7t4t9t8)   1st subkey
4.   Shift (t, 2)  ⇒   u =( t2t3t4t0t1t7t8t9t5t6)
5.   P8 (u)        ⇒ k2 = (u5u2u6u3u7u4u9u8)   2st subkey
6.   IP (m)        ⇒   m = (m1m5m2m0m3m7m4m6)
7.   IP-1  (n)     ⇒   n = (n3n0n2n4n6n1n7n5)
8.   T (m)         ⇒   m = (m4m5m6m7m1m2m3)
                                       n7 | n4    n5 |   n6
9.   Arrange n into a diagram D=       n5 | n5    n7 |   n4
10.  D+k1 =
     |n7+k10|n4+k11   n5+k12 | n6+k13  p00 | p01 p02 | p 03
     |n5+k14|n6+k15   n7+k16 | n4+k17  p10 | p11 p12 | p13
11.  SB0 [(p00p03), (p01p02)] = q0q1. SB1 [(p10p13),
     (p11p12)] = q2q3.
12.  P4 (q)        ⇒ q1 q3 q2 q0
13.  S1 (n,q)      ⇒ (n0+q1, n1+q3, n2+q2 , n3+q0,n4 ,n5,n6,n7)
13.  Repeat steps 10-13 Using 2nd subkey k2 instead to form S2
15.  Encrypt (IP-1 ° S2 ° T ° S1 ° IP)
16.  Decrypt (IP-1 ° S1 ° T ° S2 ° IP)
```

6.1. Training of SDES Cipher

During the training, the error goal (sum squared error) is defined as (0.00001 =10^{-5}), which gives 100% accuracy. After the training process has finished and the Neuro-Identifier has converged to the defined error goal, the weights (W) and biases (B) matrices are saved to be used later in the simulation phase. As an experimental result obtained from this research, emulation modes (encryption and decryption modes), a sub set of the training data was sufficient to capture the behaviour of the algorithm. Table 1 illustrates the results of NID training for SDES cipher in both modes (encryption and decryption modes). Table 2 illustrates the results of NID training for SDES cipher in Cryptanalysis modes. Figure 3 illustrates the error curve of NID training for SDES cipher in encryption of emulation mode. Figure 4 illustrates the error curve of NID training for SDES cipher in Cryptanalysis mode.

Table 1. the creation of emulation models in SDES cipher.

Meth.	**Mode**	**Trai Set**	**NN Size**	**No. Epoch**	**No. of Flops**	**Exec. Time sec.**
SDES	Encry.	1024	32*32	1640	4.871 e11	1. 943e 4
	Decry.	1024	32*32	2861	9.735 e11	2.932 e 5

Table 2. the creation of Cryptanalysis models in SDES.

Meth.	**Train Set**	**NN Size**	**No. Epoch.**	**No. of Flops**	**Execution Time sec.**
SDES	1024	32*32	7869	9.4887 e15	8.3243e 11

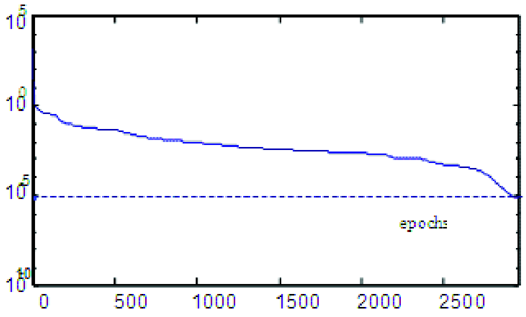

Figure 3. Error curve of the emulation SDES cipher.

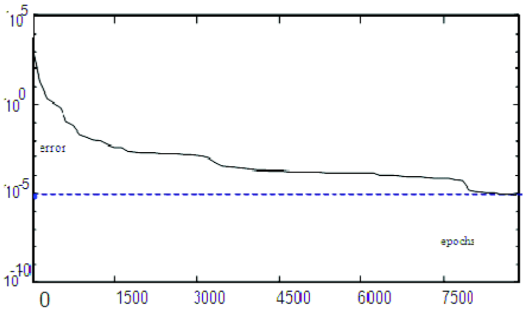

Figure 4. Error curve of the cryptanalysis SDES cipher.

6.2. Simulation of SDES Cipher

The simulation phase includes execution of the trained neural identifier in both approaches (cryptanalysis and emulation) using the saved weights (W) and biases (B), and the simulation data set (SP, SK, SC). Simulation of sdes cipher in both approaches (cryptanalysis and emulation) gives 100% accuracy for any length of key. The possible key of SDES cipher is any combination of lowercase alphabetic characters with maximum length of (1024=32*32) which is the size of the training set. Figure 5 illustrates actual and simulated key of length (300 characters) for SDES cipher, Where the value of the letter a = 0 and the letter z = 25.

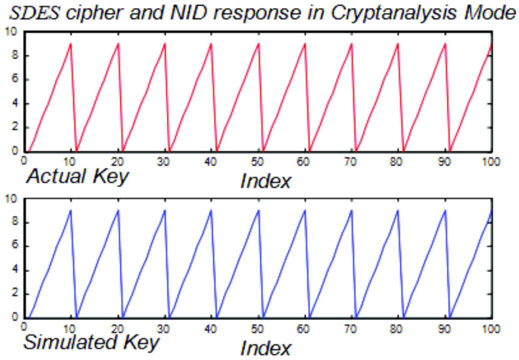

Figure 5. Actual and behaviours of simulated NID response for SDES cipher.

7. Conclusions

1. The Levenberg-Marquardt (LM) algorithm from neural network is used to train the Neuro-Identifier which gives good approximation capabilities, faster convergence, and more stable performance surface. This work present the idea of the equivalent cipher system, which is identical 100% to the unknown system, and that means an unknown hardware, or software cipher system could be reconstructed without knowing the internal circuitry or the algorithm.
2. Most of identification techniques can identify certain cipher systems, but not all of them, the presented method is a generalized one that could identify many cipher system and build the equivalent system from the input-output observations.
3. Emulation cryptography is a generalized method that could be used to all cryptographic systems. The only changeable parameter is the size of the hidden layers which should be made large enough to accommodate the key space of the target cipher system. The total number of neurons in the hidden layers is at most equal to the number of training samples, giving that the training samples are sufficient to describe the target system behavior. The feature of generalization is due to the characteristic of modelling.

4. Future works: It is possible to use this model with the most sophisticated cryptosystems such as public key, as well as to use search for any efficient algorithms in neural network. It is possible to use the genetics algorithm to reduce the space search for the keys and use the results as inputs for the neural networks to get the correct keys.

Actual and Simulated Key of SEDS Cipher
Plaintext:
amessageiscalledplaintexttheprocessofdisguisingamessageinsuchawayastohideitssubstanciscalledencryptionanencryptedmessageiscalledciphertexttheprocessofturningciphertextintoplaintextagainiscalleddecryptiontheartandscienceofkeepingmessagessecureiscryptographyanditispracticedbycryptographerscryptanalyst.
Ciphertext:

```
58  12 197 134 69 210  44109 34 120 175 234 12 172 142  5 255 247 16 32
107 169 197 230 248 64 101 109 199  81 14 78 197 134 69 178 176 219 34
120 34 155 130 134 6 254 44 0 214 29 173 220 153 126 142 230 234 126
206 242 215 234 204 192 226 210 207 255 156 107 22163 197 251 248 10
207 249 187 120 208 234  6 194 6 10 135 0 66 146 232 163 197  56  241
50 65 56 208 32 14 18 153 56 142 254 135 178 8 126 208
145 128 158 142 10 207 0 22 29 22 220 141 192 51 129 205 219 139 32
123 230 197 209 248 236 243 255 208 107 232 2 129 82 241 236  207 126
156 165 208 155 129 56 6 254 44 84 34 126 215 145 129 35 142 191 196
180 215 243 14 2 12 192 6 254 196 109 225 243 58 11153 251 164 230
207 84 16 146 238 145 128 160 142 142 106 62 199 243 22 119 6 35 15
236 145 178 208 234 107 163 65 194 6 236 234 84 92 127 199 170 197 27
35 230 234 15 214 29 173 220 153 126 142 10 207 109 139 191 157 145
130   134 241 50 65 56 208 127 34 213 153 247 15 55 145 94 90 32 208
222 65 247 192 210 135 255 34  242 232 163 28 241 241 50 65 56 208 127
34 213 153 247 15 236 106 126 139 81 132 2 35 192 164 210 245 62 33
243.
```

Simulated Key:

```
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6
7   8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3
4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0
1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7
8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3   4
5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1
2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8
9   0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1   2
3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6
7   8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3
4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0
1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7
8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3   4
5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1
2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8
9   0   1   2   3   4   5   6   7   8   9.
```

Actual Key:

```
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6
7   8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3
4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0
1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7
8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3   4
5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1
2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8
9   0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1   2
3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6
7   8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3
4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0
1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7
8   9   0   1   2   3   4   5   6   7   8   9   0   1   2   3   4
5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1
2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8
9   0   1   2   3   4   5   6   7   8   9.
```

Accuracy = 100%

References

[1] Ball G., Mian S., Holding F., Allibone R., and Lowe J., “An Integrated Approach Utilizing Artificial Neural Networks and SELDI Mass Spectrometry for The Classification of Human Tumours and Rapid Identification of Potential Biomarkers,” *Journal of Bioinformatics*, vol. 18, no. 3, pp. 395-404, 2002.

[2] Blankenship L. and Ghanadan G., “Adaptive Control of Nonlinear Systems via Approximate Linearization,” *IEEE Translation Control*, vol. 41, no. 4, pp. 618-625, 1996.

[3] Bruce S., *Applied Cryptography 2ed*, John Wiley and Sons, 1996.

[4] Chen C., *Linear System Theory and Design 3rd*, Oxford University Press, 1999.

[5] Cinar A., “Nonlinear Time Series Models for Multivariable Dynamic Processes,” *Journal of Chemometrics and Intelligent Laboratory Systems*, vol. 6, no. 1, pp. 147-158, 1996.

[6] Huang G. and Babri H., “Upper Bounds on the Number of Hidden Neurons in Feed Forward Networks with Arbitrary Bounded Nonlinear Activation Functions,” *IEEE Transaction on Neural Networks*, vol. 9, no. 1. pp. 224-229, 1998.

[7] Josef P. and Jennifer S., *Cryptography, an Introduction to Computer Security*, Upper Saddle River NJ, Prentice Hall, 1989.

[8] Khaled A., Waiel F., Mohamed A., and Alaa A., “Attack and Construction of Simulator for Some of Cipher Systems Using Neuro-Identifier,” *International Arab Journal of Information Technology*, vol. 7, no. 4, pp. 365-372, 2010.

[9] Lennart L., *System Identification: Theory for the User*, 2nd Edition Englewood Cliffs NJ Prentice-Hall, 1987.

[10] Lester H. and Jonas N., “Some Aspects of Neural Nets and Related Model Structures for Nonlinear System Identification,” *Journal of Computer Science*, vol. 6, no. 1, pp. 29-35, 2010.

[11] Liu X. and Pham T., *Neural Networks for Identifications, Prediction and Control*. New York: Springer-Verlag Ltd, 1998.

[12] Mahmood I. and Wassim A., “NonLinear System Identification Using Neural Networks,” *Journal of Automation Today, NCC*, vol. 89, no. 37, pp. 23-25, 2001.

[13] Nagi L. and Sjoberg J., “Efficient Training of Neural Nets for Non-Linear Adaptive Filtering Using a Recursive Levenberg-Marquardt Algorithm,” *IEEE Transactions on Signal Processing*, vol. 48, no. 7, pp. 1915-1927, 2000.

[14] Patterson W., *Artificial Neural Networks, Theory and Application*, Singapore, Prentice Hall. 1996.

[15] Romariz, A., Neural Network Applied to Nonlinear Modelling, available at:

http://www.ene.unb.br/romariz/, last visited 1996.

[16] Saggar M., Mericli S., and Iikkulainen J., "System Identification for the Hodgkin-Huxley Model Using Artificial Neural Networks," *in Proceeding of the International Joint Conference on Neural Networks, IEEE Explore Press*, Florida, pp. 2239-2244, 2007.

[17] Sarle S., Artificial Neural Networks and Their Biological Motivation, Internet Explorer, Artificial Intelligent World Wide Navigator, available at: http://www.csa.ru., last visited 1999.

[18] Schaefer F., "A Simplified Data Encryption Standard Algorithm," *Journal of Cryptology*, vol. 20, no. 1, pp. 77-84, 1996.

[19] Simon H., *Neural Networks*: *A Comprehensive Foundation*, 2nd Edition Prentice Hall PTR Upper Saddle River, USA, 1998.

[20] Tanomaru J., "Comparative Study of Two Neural Network Approaches for Nonlinear Identification," *in Proceeding of International Symposium on Speech, Image Processing and Neural Networks*, Hong Kong, pp. 487-490, 2002.

[21] Thomas B., "Quantitative Security of Block Ciphers: Designs and Cryptanalysis Tools," *PhD École Polytechnique Fédérale De Lausanne*, Suisse, 2008.

[22] Whitfield D. and Martin H., "Privacy and Authentication an Introduction to Cryptography," *Computer Journal of IEEE*, vol. 67, no. 3, pp. 397-427, 1979.

[23] Zbikowski R. and Dzielinski A., "Neural Approximation: A Control Perspective," *in Proceeding of Neural Network Engineering and Dynamic Control Systems, Advances in Industrial Contro*l, Berlin, pp. 1-25, 1995.

[24] Zhenran J. and Yanhong Z., "Using Gene Neural Networks to Drug Target Identification," *Journal of Integrative Bioinformatics,* vol. 1, no. 2, pp. 26-28, 2006.

Khaled Alallayah received his PhD in computer science from the Faculty of Science, El-Menoufia University, Egypt. His research interests are in the areas of information security and neural network.

Mohamed Amin is an assistant professor. Currently, he is the head of the Department Computer Science, Faculty of Science, El-Menoufia University, Egypt. His research interests are in the areas of compiler systems, artificial intelligence, information systems, information security, database and data mining and cooperating distributed systems.

Waiel AbdElwahed is the vice dean and the head of Operations Research and Decision Support Department, Faculty of Computers and Information, El-Menoufia University, Egypt. He is a reviewer of many international journals: European J. of Operational Research, International Journal of Fuzzy Sets and Systems, International J. of Mathematical Analysis and Computation, International J. of Information and Optimization, International J. of Mathematical Analysis and Simulation, International Journal of Artificial Intelligence Tools, International Journal of Transportation Research. His research interests include: Operations research and decision support and artificial intelligence.

Alaa Alhamami is currently a professor and the dean of College of Computer Sciences and Informatics, Amman Arab University, Jordan. His research interests include: Distributed database, data warehouse and data mining, and security in general. He is also the author of twelve books in different fields in computer science.