Attack and Construction of Simulator for Some of Cipher Systems Using Neuro-Identifier

Khaled Alallayah[1], Mohamed Amin[1], Waiel Abd El-Wahed[2], and Alaa Alhamami[3]
[1]Department of Mathematical & Computer Science, Al Menoufia University, Egypt
[2]Department of Computer Science, Al Menoufia University, Egypt
[3]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 classical and stream cryptosystems are discussed. The constructing of Neuro-Identifier mode is to achieve two objectives: the first one 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: *Neuro-Identefier, classical cipher, stream cipher, black box, emulation, and cryptanalysis.*

Received May 27, 2008; accepted September 1, 2008

1. Introduction

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 [19]. 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 [7]. 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]. Artificial Neural Networks (ANNs) 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 [5, 16, 18]. 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 [11, 18, 23].

2. System Identification

There are two approaches for system identification [9, 11], depending on the available information, which describes 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. Figure 1 illustrates an unknown system with x_m input signals and y_n output signals. The central concept in identification problems is identifiability [9, 11]. 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 (θ^*) if:

$$M(\theta) = M(\theta^*), \quad \theta \in D_M \Rightarrow \theta = \theta^* \tag{1}$$

where M is a model structure, q is a parameter vector, ranging over a set of values D_M [17].

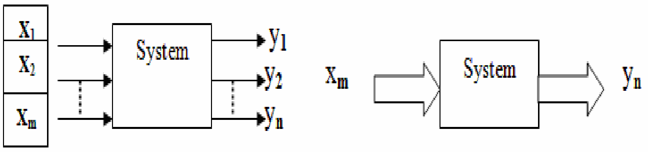

Figure 1. System with m inputs and n outputs.

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 [21]. Under this assumption, a system may be considered a Black-Box as shown in Figure 2. 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 [21, 23].

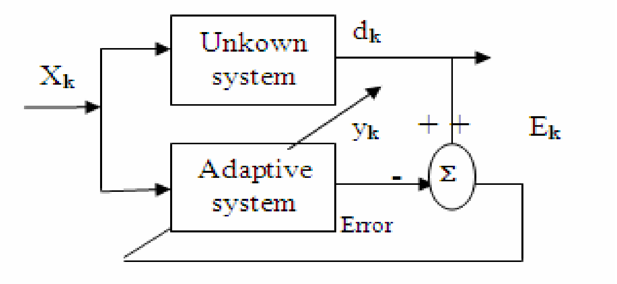

Figure 2. Adaptive system identification architecture.

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* [21]. The system can be described as in equation 2.

$$Y(k) = Hx \tag{2}$$

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 [20, 21].

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]*, as in equation 3.

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

where *[x(k), y(k)]* representing the input-output pairs of the system at time *k*, and *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 in equation 4 [5, 7, 8, 11]:

$$\begin{aligned} y(k)=a_1y(k-1)+a_2y(k-2),\ldots+a_ny(k-n) \\ +b_1x(k-1)+b_2x(k2),\ldots.+b_mx(k-m) \end{aligned} \tag{4}$$

where $a_i(i=1,2,...,n)$ and $bi(i=1,2,...,m)$ are real constants. Equation 4 can be rewritten in matrix notation as in equation 5.

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

For Multi-Input Multi-Output (MIMO), *y(k)* and *x(k)* are of dimensions m and p respectively, equation 5 can be rewritten as in equation 6 [8]:

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

where *Ai* and *Bj* an (m x m) and (m x p) matrices respectively.

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 keyspace (*K*). See a reversible encryption algorithm [20, 21] in equation 2.

$$E_k: M \rightarrow C, \text{ such that:}$$

$$E_k(m) = c,\ k \in K,\ m \in M,\ c \in C \tag{7}$$

An inverse decryption algorithm as in equation 8. $D_k = E^{-1}{}_k$: D_k: C→ M, such that:

$$D_k(c) = D\,k\,[E_k(m)] = m \tag{8}$$

The keys should uniquely define the enciphered message as in equation 9;

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

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 [12].

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 [4, 13, 14]. 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). The optimal structure then, is the one having the fewest units (neurons) for which the criterion is met. Neuro-Identifiers (NIDs) 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 [8, 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 [6, 22]. Figure 3 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 (number of neurons in the hidden layer) 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 [11, 21]. The essential constraint on increasing the size of hidden layers is the limitation of the hardware architecture used in the experimental work.

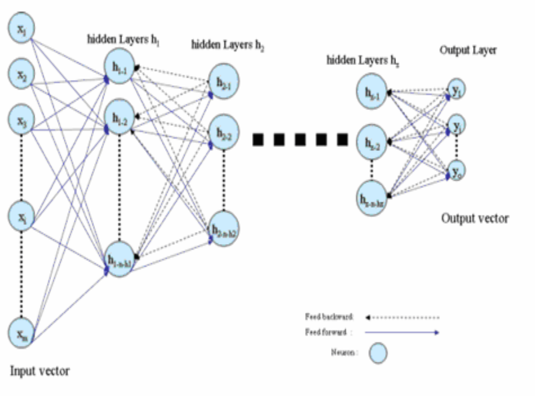

Figure 3. Multi-layer feed forward Neuro-identifier architecture.

Training Algorithm

```
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. $h0i = \sum i=1 Xi Wij + bi$ .  $hi = f(h0j)$..
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.
    $Wb = [w1b1 w2b2 ... wpbp].$
    $\Delta Wb = (JT.J + \eta I)-1. (-JT.\delta).$
7- Update biases and weights.
    $Wij (new) = wij (old)+ \Delta Wb.$
8- End.
```

5.1. 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: construction of an neuron-model for the target unknown cipher system.

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: determination of the key of a given plaintext and ciphertext pair, which belongs to unknown cipher system.
   -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 behavior 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; obtained is clearly a pure cryptanalysis target (total break). One way this is done is 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. Classical Ciphers

NID, as described above, has been used in this research in classical 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 simple and polyalphabetic substitution systems (Caesar**,** Affine, Beaufot and Vigenere ciphers). Lower case alphabets *(a, b, c,..., z)* are chosen as a subset from ASCII character set, and used for training and testing the NID. That is because of the limitation of the available hardware, i.e., a large network is needed for the large space of training parameters (plaintext, key, and cipher text).Training data are built as a combination of pairs of plaintext *P*, and key *K* and passed to the target cipher algorithm to produce the ciphertext *C*. Hence for 26 possible plaintext characters, and 26 possible key characters, we use (26 * 26 = 676) possible pairs of (*P*, *K*) and 676 possible cipher characters (*C*). This means that each possible combination of plaintext character and key character is taken as a training example. Testing data is taken from a text file that includes only lower case alphabets *(a, b, c,..., z)*. This file is encrypted by the target some cipher system algorithms for simple substitution and polyalphabetic substitution ciphers [12]. In simple substitution (or monoalphabtic) ciphers each character of the plaintext is replaced with a corresponding character of ciphertext. A single one-to-one mapping function (*f*) from plaintext to ciphertext character is used to encrypt the entire message using the same key (*k*), such that [1].

$$Ek\ (M) = f\,(m_1)\,f\,(m_2)\ldots.f\,(m_n) = C \qquad (10)$$

where n is the length of the message, M is plaintext message given by $M= (m_1, m_2,\ldots, m_n)$*, and* C is ciphertext message given by $C= (c_1, c_2,\ldots.,c_n)$.

Simple substitution ciphers are often called monoalphabetic ciphers. Several forms of *f* can be used in simple substitution, such as:

- Shifted alphabet (Caesar cipher): the most straightforward substitution cipher is the Caesar substitution, named after the Roman Emperor Julius Caesar (100-44 BC). The ciphertext is obtained by simply shifting the original alphabet, and can be represented mathematically as in equation 11:

$$f(a)= (a+k) \bmod n \qquad (11)$$

where k is the number of position to be shifted, a is a single character of the alphabet, and n is the size of the alphabet.

- Addition and multiplication (affine transformation): a mixture of addition and multiplication obtains the displacement as in equation 12.

$$f(a)= (ak_1+k_0) \bmod n \qquad (12)$$

where k_1 and n relatively prime. Simple substitution ciphers dose not hides the underlying frequencies different letters of the plaintext, and hence it can be easily broken. Apolyalphabetic cipher means a sequence of monoalphabetic ciphers, which are often referred to as substitution alphabets or just alphabet. In another meaning; it is made of multiple simple substitutions. The sequence of the substituting alphabet may have fixed length (*p*) and is denoted as its period [1, 20].

- Vigenere cipher: A popular form of periodic substitution ciphers is the Vigenere cipher. The key is specified by a sequence of letters, $K= k_1, k2,\ldots,kP,$ then Vigenere cipher system is defined as as in equation 13:

$$Fi(a)= (a+ki) \bmod n \qquad (13)$$

For plaintext letter *a*, and key letter *k*, the ciphertext $c=Fi$.

- Beaufort cipher: Another periodic cipher is Beaufort cipher which is similar Vigenere but using subtraction instead of addition, and defined as as in equation 14:

$$Fi(a)= (ki-a) \bmod n \qquad (14)$$

For plaintext letter *a*, and key letter *k*, the ciphertext $c= Fi$.

6.1. Training of Classical 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 NID has converged to the defined error goal, the weights (*W*) and biases (*B*) matrices are saved to be used later in the simulation phase. Figure 4 illustrates the error curve during the training process for the Beaufot cipher system.

Result shows in Table 1 that the creation of emulation models in classical cipher, and Result shows in Table 2 that the creation of attack (cryptanalysis) models in classical cipher.

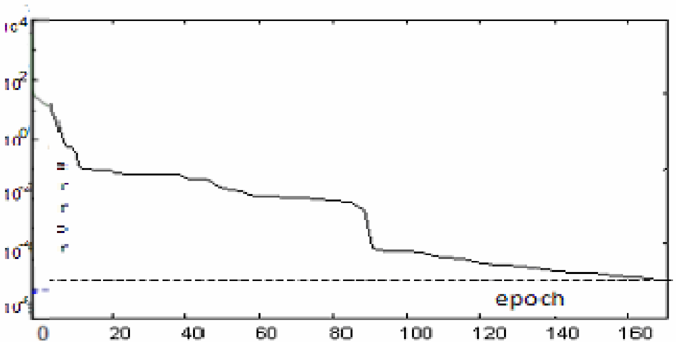

Figure 4. Error curve for Beaufort cipher.

6.2. Simulation

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 vigenere cipher in both approaches (cryptanalysis and emulation) gives 100% accuracy for any length of key. The possible key of vigenere cipher is any combination of lowercase alphabetic characters with maximum length of (676=26*26) which is the size of the training set. Figure 5 illustrates actual and simulated key of length (300 characters) for Beaufort cipher.

Table 1. Creation of emulation models in classical cipher.

Cipher System	Mode	Train Set	NN Size	No. of Epochs	No. of Flops	Execution Time Sec
Caesar	Encryption	26	26	8	7455570	.38
	Decryption	26	26	11	11644550	.40
Affine	Encryption	26	26	51	57230444	1.49
	Decryption	26	26	71	80171458	2.03
Vigenere	Encryption	676	26*26	78	1.2012 e9	1.0162 e3
	Decryption	676	26*26	101	1.5116e9	1.3609e3
Beaufort	Encryption	676	26*26	99	1.5094e9	1.3321e3
	Decryption	676	26*26	127	1.6934e9	1.8345e3

Table 2. Creation of (cryptanalysis) models in classical cipher.

Cipher System	Train Set	NN Size	No. of Epochs	No. of Flops	Execution Time Sec
Caesar	26	26	31	31548120	1.34
Affine	26	26	86	97171458	5.42
Vigenere	676	26*26	136	2.694e9	2.944e3
Beaufort	676	26*26	183	2.907e9	3.687e3

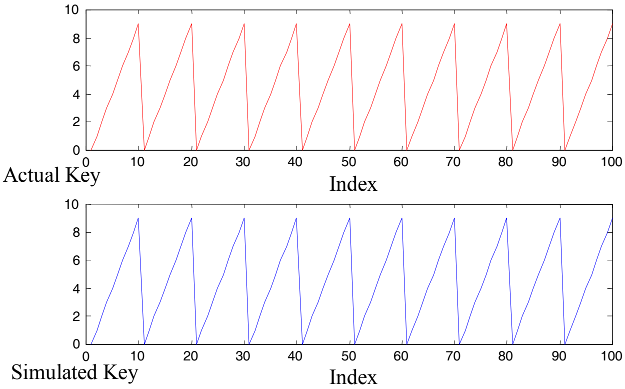

Figure 5. Actual and behaviours of simulated NID response for Beaufort cipher.

Actual and simulated key of vigenere cipher.

- *Plaintext:*
  securityofacryptographicsystemisdirectlyrelatedtothediffi
  cultyassociatedwithinvertingencryptiontransformationoft
  hatsystemtheprotectionaffordedbytheencryptionprocedure
  canbeevaluatedbytheuncertaintyfacinganopponentindeter
  miningthepermissiblekeysusedtherearetwofundamentally
  differentwaysinwhichcryptographicsystemsmaybesecurei
  nsomesystemstheamountofinformationavailabletothecryp
  tanalystisactuallyinsufficienttodeterminestheencipheringa
  nddecipheringtransformationsnomatterhowmuchcomputi
  ngpowerthecryptanalysthaveavailableasystemofthiskindis
  calledunconditionallysecureshannoncalledsuchsecrecyasp
  erfectsecr
- *Ciphertext:*
  ixcjnfhohfpyomfmovnxdxbchuphufihzffuvtauosbtttzqcja
  esectyvuapvoilorexhuwwxpewdoegpfbwxnrnvdjbocpood
  lfdnjojbockchxtthuphuftwamfemerpfcdtfukoruwbnpesug
  cgumhyhnenlquwugazoduetrxzktttzymjaejjzshmaxjqmvtc
  xjdodhpekksdmiczbhukmxjfbwmhtlbfcbsheyzudenorguw
  twaosqkeisltkgdpibbjtlauawvyegakhmtyhekkxbcwyomfm
  ovnxdxbchuphufsbwvpulerqosygsdibgolttiphxxabkrbjhfx
  jcchfaielbqoaxhxpbxtdpesskyepxbqeyhpfgqvtjwizobnhqc
  tyvitjqheweiaoaygehpesugcxleshbnvwkrtxcxleshbnvpood
  lfdnjojbocokccttiaovepmjyeqefpjpfbwiolaohxxcgumhqga
  auphxtvtwsoyeaqhboirsiajcvmhxohwdwihyxzbxdjjzcdwii
  elbqelnobqkkehdxbdhnrwizuwsjyeguvrtyvoiiegbbqjler
- *Actual key:*
  Khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhale
- *Simulated key:*
  Khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhaledmkhale
  dmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledmkh
  aledmkhaledmkhaledmkhaledmkhaledmkhaledmkhaledm
  khaledmkhaledmkhaledmkhaledmkhaledmkhale

ACCURACY = 100%

7. Stream Ciphers

NID has been used as a general Emulation model for all stream cipher systems (linear and nonlinear keystream generators). It is used as a black-box attack to extract the key sequence. Cryptanalysis approach arid emulation approaches have two phases: training phase and simulation phase. Both approaches will be discussed in the next articles. The lowercase alphabet is taken to represent the set of plaintext characters. Plaintext characters are converted into binary digits using the International Telegraph Alphabet No 2 (ITA2) which consist of five bits for each character. The reason for choosing such a coding system is the requirement to reduce the space of the generated key, and the space of the ciphertext consequently. In this case the range of ciphertext character will be in the range of (0 0-25 i.e., 0- 31).

The training data set is designed to exhibit the feature of each possible plaintext character with each possible keystream. It consists of combination pairs of plaintext and key stream with a size of (832=32*26). This is passed to the XORing function to produce the ciphertext.

The testing data set is built by choosing a text file as plaintext and selected keystreams from different known generators (as mentioned below) to produce the ciphertext [3,15]. The following nonlinear generators have been chosen as an example:

- Exclusive or "XOR": a nonlinear generator system consists of two LFSRs. Their lengths (*k1* and *k2*) are relatively prime and all feedback polynomials are primitive. The period of this generator is (*2k1-1*) (*2k2-1*) [2--]. For example LFSR of (5,7) bit stages feedback function (*S1 ⊕ S2*) and maximal length of ((25-1)*(27-1) = 3937) bits [1, 2].
- Hadmard Generator: a nonlinear generator system consists of two LFSRs. Their lengths (*k1 and k2*) are relatively prime and all feedback polynomials are primitive, *k1≠k2* and gcd (*k1,k2*)=1. The period of this generator is (*2k1-1*) (2k2-1) [2]. For example LFSR of (5,7) bit stages feedback function (*S1^S2*) and maximal length of ((25-1)*(27-1) = 3937) bits[1, 2].

7.1. Training of Stream Cipher

During the training by Levenberg Marquardt (LM) algorithms of both approaches, the error goal (sum square error) is defined as 0.00001 = 10-5, which gives 100% of accuracy. After the training process has finished and the neural 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. Figure 6 illustrates the error curve during the training process for stream cipher systems in cryptanalysis approach. Result shows in in Table 3 that the creation of attack (cryptanalysis) models in stream cipher.

7.2. Simulation

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 stream cipher in both approaches gives 100% accuracy for a key length less than or equal to the size of training set. The possible output keystreams are any output of the keystream generators combined with any plaintext (lowercase alphabet only) with maximum length of 832 characters (832*5=4160 bits) which is the size of the training set. Simulation of Stream in all modes gives 100% accuracy.

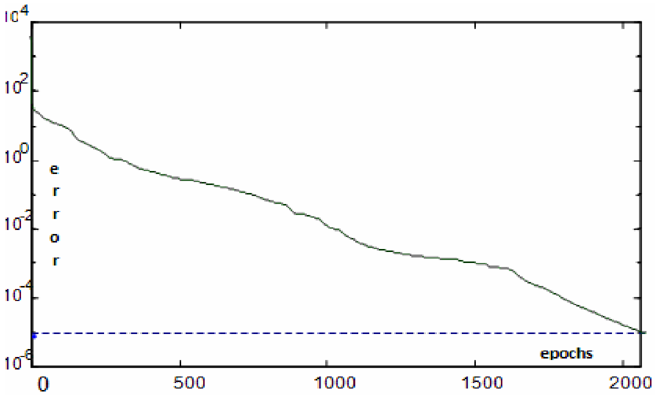

Figure 6. Error curve during the training process for stream cipher (cryptanalysis).

Table 3. That the creation of (cryptanalysis) models in stream cipher.

Cipher System	Train. Set	NN Size	No. Epoch	No. Flops	Ex. Time Sec
Crypt-Analysis	832	26*32	2155	8.712e12	3.285e5

8. Conclusions

- The proposed model attack method has been presented with two approaches; the first approach is to determine the enciphering key K, which satisfies the cryptanalysis goal. In cryptanalysis terminology, it is classified as a total break. The second approach presents a new method in cryptology; Black-Box NID offers the construction of a Neuro-Model for the target cipher system. The constructed Neuro-Model can be considered as an equivalent system to the target system.
- The LM algorithm form neural network is used to train the NID; it gives good approximation capabilities, faster convergence, more stable performance surface, and the ability to reach any degree of accuracy with enough degrees of freedom in the hidden layers. It is also demonstrated that LM algorithm is a proper choice in off-line training for complex nonlinear systems. Furthermore, it could be amended to accommodate on-line training of such systems.
- Most of identification techniques can identify certain cipher systems, but not all of them, the presented method are a generalized method that could identify many cipher system and build the equivalent system from the input-output observations.

References

[1] Al-Hamami H., "Attacking Classes of Stream Cipher Using GA," *in Proceedings of 3rd Scientific Conference on Data Security*, Iraq, pp. 241-244, 2001.

[2] Al-Hamami H., "Designing Stream Cipher Algorithm Using a New Approach," *Computer Researches Magazine*, vol. 3, no. 2, pp. 165-169, 2002.

[3] Alex V., "The Design of a Stream Cipher," *in Proceedings of Selected Areas in Cryptography*, pp. 67-75, 2007.

[4] Blankenship L. and Ghanadan R., "Daptive Control of Nonlinear Systems via Approximate Linearization," *Computer Journal of IEEE Transactions Control*, vol. 41, no. 3, pp. 618-625, 1996.

[5] Haykin S., *Neural Networks: A Omprehensive Foundation*, Prentice Hall, 1994

[6] Huang B. and Haroon A., "Upper Bounds on the Number of Hidden Neurons in Feed Forward Networks with Arbitrary Bounded Nonlinear Activation Functions," *Computer Journal of IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 224-229, 1998.

[7] Josef P. and Seberry J., *Cryptography an Introduction to Computer Security*, Prentice Hall, 1989.

[8] Karayiannis B. and Venetsanopoulos A., *Artificial Neural Networks, Learning Algorithms, Performance Evaluation, and Applications*, Kluwer Academic Publishers, 1995.

[9] Ljung L., *System Identification, Theory for the User*, Prentice-Hall, 1987.

[10] Lennart L., System *Identification, Theory for the User*, Prentice-Hall, 1987.

[11] Mahmood K. and Wassim A., "NonLinear System Identification Using Neural Networks," *Computer Journal of NCC*, vol. 8, no. 37, pp. 39-41, 2000.

[12] Matthew R. and John A., *Using Ants to Attack a Classical Cipher*, Springer, 2003.

[13] Ngia S. and Jonas S., Efficient Training of Neural Nets for Non Linear Adaptive Filtering Using a Recursive Levenberg Marquardt Algorithm. Internet Explorer, www.Chalmers.se /download/publication, Last Visited 1999.

[14] Ngia H. and Jonas S., Some Aspects of Neural Nets and Related Model Structure for Nonlinear System Identification, *Report CTH-TE-70*, 1998.

[15] Orr D. and Nathan K., Attacks on Stream Ciphers, *in Proceedings of State of the Stream Ciphers Workshop*, pp. 249-258, 2008.

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

[17] Prem K. and Rajiv S., "Identification International Symposium on Speech," *in proceedings of ICSSC AIAA 2001*, Hong Kong, pp. 487-490, 2001.

[18] Romariz A., "Neural Network Applied to Nonlinear Modeling," www.ene.unb.br/romariz/, Last Visited 1996.

[19] Sarle S., "Artificial Neural Networks and Their Biological Motivation," www.csa.ru, 1999.

[20] Schneier B., *Applied Cryptography, Protocols, Algorithms, and Source Codes in C*, John Wiley and Sons, 1996.

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

[22] Tanomaru J., *Comparative Study of Two Neural Network Approaches for Nonlinear*, Elsevier Science Ltd, 1994.

[23] Xing L. and Phams D., *Neural Networks for Identifications, Prediction and Control*, New Springer-Verlag Ltd, 1998.

[24] Zbikowski R. and Dzielinski A., *Neural Approximation: A Control Perspective Neural Network Engineering and Dynamic Control Systems Advances in Industrial Control*, Springer-Verlag, 1995.

Khalid Alalayah is presently at the Computer Science Department, Faculty of Science, Al Menoufia University, Shiben EL-Kom, Egypt. His research interested is in information security and neural networks.

Mohamed Amin is the head of Department of Computer Science, Faculty of Science. Al Menoufia University, Shiben EL-Kom, Egypt. His research interested includes compiler systems, artificial intelligence, information systems, information security, data base and data mining, and distributed systems.

Waiel Abd El-Wahed is the vice dean and head of Operations Research and Decision Support Department, Faculty of Computers and Information, El- Menoufia University, Shiben EL-Kom, Egypt.

Alaa Al-Hamami is presently a professor and dean of Graduate College for Computing Studies, Amman Arab University for Graduate Studies, Jordan. His research is focused on distributed database, data warehouse and data mining, and security in general. He is also author of many books in different fields in computer science.

.