ANALYSIS OF NEURAL NETWORK BASED CIPHERS

By

Maryam Arvandi
B.Sc. in Computer Science, Ryerson University, June 2003

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of

Master of Applied Science
in the program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2005

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

Signature Maryam Arvandi

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature Maryam Arvandi

Analysis of Neural Network Based Ciphers
Master of Applied Science, 2005
Maryam Arvandi
Electrical and Computer Engineering
Ryerson University

Abstract

Cryptography can be considered one of the most important aspects of communication security with existence of many threats and attacks to the systems. Unbreakableness is the main feature of a cryptographic cipher. In this thesis, feasibility of using neural networks, due to their computational capabilities is investigated for designing new cryptography methods. A newly proposed block cipher based on recurrent neural networks has also been analyzed. It is shown that: the new scheme is not a block cipher, and it should be referred to as a symmetric cipher; the simple architecture of the network is compatible with the requirement for confusion, and diffusion properties of a cryptosystem; the back propagation with variable step size without momentum, has the best result among other back propagation algorithms; the output of the network, the ciphertext, is not random, proved by using three statistical tests; the cipher is resistant to some fundamental cryptanalysis attacks, and finally a possible chosen-plaintext attack is presented.

ACKNOWLEDEMENTS

I would like to express my sincere appreciation to Dr. Alireza Sadeghian for giving me such a wonderful opportunity to perform research in a friendly and encouraging environment. This effort was not possible to be completed without his valuable and insightful guidance. I am also very grateful to the members of committee Dr. A. Anpalagan, Dr. I. Woungang, and Dr. E. Harley for their instructive suggestions.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION TO DATA SECURITY

1.1 Introduction

In today's universal electronic world, security plays an important role due to existence of viruses, hackers, electronic eavesdroppings and electronic frauds. Computer systems and their interconnections via networks have been grown explosively. Therefore, organizations and individuals are being more dependent on using these systems for storing information and communicating. At the same time, as potential attackers can operate from all over the globe, this information is more vulnerable to abuse and also the risk for misuse has considerably increased. Moreover, if someone malicious, gains access to an electronic information system, compare to a paper-based system, the scale and impact of the abuse can be much larger [1].

Information Security is an old concept, but a new field of specialization. It involves protection of information, especially personal or sensitive information, from deliberate or accidental loss or misuse. The field is increasingly important in government and insurance companies, banks, hospitals, universities, etc. The dependence on computers to store and transmit information accurately and securely is growing rapidly, and at the same time the vulnerabilities in the systems and human habits are becoming increasingly obvious and open to exploitation.

In past, safeguarding or tracking the data was relatively easy in paper-based businesses. When we began to use computers for business functions, data management became more difficult, but it was still manageable and most vital functions were handled on paper. In those early days of computing, no one thought much about information security because networks were not common and few people had the knowledge of how to use them. Each computer had its own files and there was not an easy way to communicate with other computers. In order to share information, it should be copied on a disk and then the disk should be taken to the other computer to transfer the data. If the computer's user set a password on the files and locked the office door, the information stored on the computer could be relatively safe [2].

Then by introduction of networks, the rules of storing information changed fundamentally, but human behavior didn't change. Computers were connected together, building networks and subnets, sharing common network space, printers and other services. Still, no one paid much attention to information security. Most people thought the only way to get a computer virus was from a floppy disk, and almost everyone believed that their personal information was worthless to anyone else.

From that time, the information world has changed. On the good side, worlds of information not previously available are now freely downloadable, and the ability to conduct research no longer depends completely on physical access to good collections of books and papers. However, there has been a cost, which could not have been envisioned in computing's early days. Information has become valuable. The way information is used and the way information travels has changed.

In an organization all the assets are subject to loss, damage or destruction from various causes, and information systems tend to be very susceptible to these dangers for many reasons. First, components of computers in comparison to other devices are more fragile. Computer hardware can be damaged more easily than, for example the tools on an automobile assembly line, and data files are extremely fragile compared with most other organizational assets. Second, computer systems are likely to be the target of disgruntled workers, protestors, and even criminals. Finally, the use of networks, distributed processing and decentralized facilities have increased the vulnerability of information and computers [1].

Data security has a main goal that is restricting access to information and resources to just authorized principals. A malicious person can have access to information by threatening or attacking the distributed systems.

Security threats are categorized in three broad areas [3]:

- Leakage, which happens when unauthorized recipients have access to information.
- Tampering, which happens when information is altered by unauthorized principals.
- Vandalism, which happens when proper operation of a system is interfered without a gain to the perpetrator.

On distributed systems an attacker can obtain access to existing communication channels or establish new channels that disguise as authorized connections. Attacks can be classified based on the way in which a channel is misused [3]:

- Eavesdropping: Attacker obtains copies of messages without having authority.
- Masquerading: Attacker sends or receives messages by using the identity of another principal without having their authority.
- Message Tampering: Attacker intercepts messages and alters their contents and then passes them on to the intended recipient.
- Replaying: Attacker stores intercepted messages and sends them at a later date.
- Denial of service: Attacker floods a channel or other resource with messages so others cannot have access to it.

Both X.800[1] and RFC 2828[2] use another classification of security attacks, which is in terms of passive attacks (such as, eavesdropping, release of message contents and traffic analysis) and active attacks (such as, masquerade, replay, message tampering and denial of service).

Although the threats and attacks to the distributed systems are significant, the advantages of working in a networked world are substantial. The internet has unlocked a world of information; the intellectual riches of the ages and the best ideas of tomorrow are freely

[1] The International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) Recommendation, Security Architecture for OSI (Open Systems Interconnection)

[2] Internet Security Glossary

available to all. The ability to communicate, to read and to learn, in this environment, is an opportunity that has never before been available. In order to negotiate safely in this new world, a few skills are required and the rewards of following these skills are enormous [2].

In order to prevent and identify the above-mentioned attacks, and to protect electronic information systems, adequate security services are required. The security services are categorized into [4]:

- Authentication (the protection against active eavesdroppers): Assures that the communicating entity is the one that it claims to be.
- Access Control: Prevents unauthorized use of a resource.
- Data Confidentiality (the protection against passive eavesdroppers): Protects data from unauthorized disclosure.
- Data Integrity: Assures that data received are exactly as sent by an authorized entity.
- Nonrepudiation: Prevents either sender or receiver from denying a transmitted message.

One of the most important aspects of communications security is probably cryptography and it can be considered an important basic building block of computer security [4]. The use of cryptography for protection of the secrecy of information is as old as writing itself, which makes it an old art and a young science [1],[5]. It is both a very difficult area of mathematics and a key technology for the information society.

The basic objective of cryptography is to enable two persons to communicate over an insecure channel in such a way that an opponent cannot understand what is being said [5]. In addition to provide confidentiality, it is often required for cryptography to perform authentication, integrity and nonrepudiation. Cryptographic methodologies are categorized into two groups: conventional and non-conventional. They are discussed in the following two subsections.

1.2 Conventional Cryptography

In general there are two well-known types of conventional cryptosystems: private key (symmetric, secret key, or single-key) and public key (non-symmetric, double key).

In 1977, the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), adopted the Data Encryption Standard (DES) [4],[8] as the most widely used encryption scheme. The standard is referred to as Federal Information Processing Standard 46 (FIPS PUB 46) and the algorithm is referred to as the Data Encryption Algorithm (DEA). This algorithm is a symmetric block cipher, which encrypts data in 64-bit blocks using a 56-bit key. In a series of steps, algorithm transforms 64-bit input into a 64-bit output. Decryption uses the same steps, with the same key (reverse of the encryption).

In 1999, a new version of DES standard (FIPS PUB 46-3) was issued by NIST. In this standard was mentioned that triple DES (3DES) [4],[8] should be used instead of DES (DES can only be used for legacy systems). The main drawback of 3DES is that the algorithm is relatively slow in software. A secondary drawback is that both DES and 3DES use a 64-bit block size. To have both efficiency and security, a larger block size is required. As a replacement, NIST in 1997 issued a call for proposals for a new Advanced Encryption Standard (AES), which should have a security strength equal to or better than 3DES and its efficiency should be significantly better. In addition, the algorithm must be a symmetric block cipher with a block length of 128 bits and it should support for key lengths of 128, 192 and 256 bits. In November 2001, NIST selected Rijndael as the proposed AES algorithm [4],[8]. The two researchers who developed and submitted Rijndael for the AES are both cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen. The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits.

A new approach to cryptography was introduced by Diffie and Hellman in 1976 [4]. This new approach challenged cryptologists to come up with a cryptographic algorithm that could meet the requirements for public-key systems [4],[8],[20]. In 1977 Ron Rives, Adi Shamir, and Len Adleman at MIT responded to the challenge by developing Rivest-Shamir-Adelman (RSA) scheme, first published in 1978. Since then, the RSA scheme has been granted as the most widely accepted and implemented general-purpose approach to public-key encryption. It is a block cipher in which the plaintext and ciphertext are integers between zero and n-1 for some n.

1.3 Non-Conventional Cryptography

In this methodology computational intelligence methods such as Neural Networks (NNs) are being used. Application of NNs in cryptography does not have a long history. In recent years, there have been limited attempts to use NNs for cryptography purposes. Su et al. [6] proposed to use unpredictable outputs of a NN together with a dedicated hardware to encrypt digital signals. The randomness of the output of the system, built using a specific VLSI architecture, determines whether the encrypted data is predictable or not. Yee and De Silva [7] suggested the use of MultiLayer Perceptron (MLP) for key scheduling that employs a mutating algorithm comprising of modular arithmetic and Feistel cipher [8]. This method has a fixed key length. They also proposed the use of MLP Networks [9],[10] in public key cryptography and as a one-way hash function. Meletiou et al. [5] proposed the use of feedforward NNs for computing the Euler function in RSA cryptosystem. These works [5]-[7],[9],[10] are still at their early stages and they have only been published in the form of short conference papers.

Kinzel and Kanter [11] proposed and studied analytically a neural cryptography scheme that is based on a mutual learning process between two parity feed-forward NNs with discrete and continuous weights. The synchronization process is claimed to be non-self-averaging and the analytical solution is based on random auxiliary variables. The learning time of an attacker that is trying to imitate one of the networks has been examined analytically and is reported to be much longer than the synchronization time. Kilmov et al. [12] have shown that Kinzel's protocol can be broken by geometric, probabilistic, and genetic attacks, and as such is not entirely secure. Karras and Zorkadis [13] have proposed the construction of robust random number generators based on MLP networks to be used in security mechanisms.

1.4 Conclusion

Security plays a primary role in our daily lives and we automatically take basic precautions to secure our valuables. Information is a valuable asset for many companies, banks, hospitals, universities etc. Therefore, the ultimate goal is to design an unbreakable cryptographic method. Block cipher design and key management have critical roles in successfully designed cryptographic systems [15]. In addition to meeting standard specifications relating to encryption and decryption, such systems must meet increasingly stringent specifications concerning the information security. This is mostly due to the steady demand to protect data and resources from disclosure, to guarantee the authenticity of data, and to protect systems from web based attacks.

There has been explosive growth in research in all aspects of cryptography, and cryptanalysis is one of the most active areas. Many cryptosystems, which had been thought to be secure, have been broken, and at the same time a large set of mathematical tools useful in cryptanalysis have been developed [14]. On the other hand, developments of both hardware and software of computer have entered an accelerative period. All these lead to the fact that new data encryption technologies are necessary.

Availability of a variety of cryptanalytic attacks that can often be successfully used to recover the key or the plaintext, make the design of cryptography algorithms complicated. An additional confounding factor is that the security of the cryptographic systems is a difficult property to measure and can be often known only empirically. The security of most of the algorithms is directly related to the degree of difficulty associated with the recovery of the key. For these reasons, the development of new cryptographic algorithms is a challenging task.

An attractive solution to these problems is provided by NNs based approach that has a suitable framework within which data can be readily coded. NNs, or to be more precise Artificial Neural Networks (ANNs), represent an emerging technology rooted in many disciplines. They have some unique attributes such as, universal approximation (input-output mapping), and the ability to invoke weak assumptions about the underlying physical phenomena responsible for the generation of the input data. ANNs gain their computing power from two properties, first, their massively parallel distributed structures and, second, their ability to learn and therefore, generalize (generalization refers to the neural network producing reasonable outputs for inputs, which are not encountered during training process) [16].

The use of NNs as means of cryptography and block cipher design was proposed in a previous thesis [17]. The approach presented a new potential source for private/public key cryptographic schemes, which are not based on number theoretic functions, and have small time and memory complexities. When RSA was invented, it was predicted that the original RSA challenge number (129-digit) would not be factored within the foreseeable future. However, advances in algorithms and computer architecture have made such factorization fairly routine (although substantial computing resources are still required). Even the new RSA challenge number, RSA-754 (174-digit), was successfully factored.

It is proposed that the inherent knowledge representation problem and the lack of transparency [18] in NNs can be advantageous when used for cryptography purposes. Therefore, NNs were applied to design block ciphers based on symmetric key. The intention was to show that by selecting learning rate to be large, the ciphertext looks random even if the plaintext has a repeated pattern. The security of the proposed block cipher was based on the assumption that the weight distribution of the hidden layers is unpredictable without knowledge of the original key. The proposed block cipher should fulfill a number of requirements including: support for variable key length, support for variable block length, and support for improved security. This is in contrast to previous works, where the applications of hardware specific/dependent chaotic NNs, use of MLP

NNs for key scheduling with fixed key length, and use of synchronizing feedforward NNs for encryption are suggested.

The objectives of this thesis are to:

- Perform a thorough literature research on the previous works which have been done on NN-based cryptography.
- Check if the cipher is really a block cipher or not.
- Explain the reasons behind using such a simple architecture for the cipher.
- Test the cipher using different kinds of back propagation algorithms to see the differences in performance.
- Check for randomness in ciphertext - originally it was assumed that the ciphertext is random.
- Analyze the strength of the cipher against the known attacks such as differential, linear and brute-force attacks.
- Present a possible chosen-plaintext attack.

As with any new cryptography method, analyzing the cipher plays a crucial role in its development process. The main contribution of this thesis is to analyze the newly proposed NN-based block cipher. In order to accomplish this task, chapter two is allocated to describing cryptography methods and thorough literature review of the related works. In chapter three, the block cipher [17] is described. Chapter four is the main chapter, in which the cipher is analyzed and its strengths and weaknesses are highlighted. Chapter five is the conclusion and also in this chapter some future research directions are described.

CHAPTER 2: Background and Methods of Cryptography

2.1 Cryptography

Cryptography is the science and study of secret writing [19]. An original message is known as plaintext, while the coded message is called the ciphertext. The process of transforming from plaintext to ciphertext is known as enciphering or encryption. The process of turning ciphertext back into plaintext is deciphering or decryption. The area of study of cryptography contains many schemes used for encryption and each of these schemes is known as a cryptographic system or a cipher.

Cryptography has a long and fascinating history. The predominant practitioners of the art were those associated with the military, the diplomatic service and government in general. Cryptography was used as a tool to protect national secrets and strategies. Due to the proliferation of computers and communications systems in private systems, demand for information protection in digital form and providing security services has been increased [20].

A cryptographic system (cryptosystem) has five components [19]:

1. A plaintext message, M ,
2. A ciphertext message, C ,
3. A key, K ,
4. An encryption algorithm (function), $E_k : M \rightarrow C$, where $k \in K$.
5. A decryption algorithm (function), $D_k : C \rightarrow M$, where $k \in K$.

Cryptosystems can be characterized in three independent areas [4]:

1. The type of operations used for transforming plaintext to ciphertext. There are two general principles for encryption algorithms: substitution (replacing bits, characters, or blocks of characters of plaintext with substitutes) and transposition (rearranging the elements in the plaintext). In this regard, all operations must be reversible and no information should be lost.

2. The number of keys used. For this area, two systems exist: symmetric or secret-key encryption (both sender and receiver use the same key) and asymmetric or public-key encryption (the sender and receiver each uses a different key).
3. The way in which the plaintext is processed. There are two processes: block cipher (is a function that maps n-bit plaintext block to n-bit ciphertext block) and stream cipher (it takes the plaintext string and produces a ciphertext string, it can be considered a block cipher with block length equal to one).

The basic objective of cryptography is to enable two people to communicate over an insecure channel in such a way that an opponent cannot understand what is being said. Cryptographic methodologies are categorized into two groups: conventional and non-conventional, which are mentioned in what follows. The two people that communicate are usually referred as "Alice" and "Bob".

2.1.1 Conventional Cryptography

In general there are two well-known types of conventional cryptosystems: private key (symmetric, secret key, or single-key) and public key (non-symmetric, double key). These two schemes are described in the following subsections.

A. Private Key Encryption

A private key (symmetric) encryption scheme has five elements [4]:

- Plaintext, which is the original comprehensible message or data. It is the input to the algorithm.
- Encryption algorithm, which performs different operations (substitution and transformations) on the plaintext.
- Secret key that is also an input to the encryption algorithm. The key is a value independent of the plaintext. By using different keys, the algorithm will produce different outputs.

- Ciphertext, the incomprehensible message, is the output of encryption algorithm. It depends on the plaintext and the secret key.
- Decryption algorithm, which is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

For a secure symmetric encryption, there are two requirements:

1. Encryption algorithm must be strong (one cannot decrypt the ciphertext or find out the key, even with the knowledge of the algorithm and access to ciphertext).
2. The copies of the secret key have to be given to sender and receiver in a secure way and the key must be kept secret.

The above requirements imply that there is no need to keep the algorithm secret and only the key must be kept secret. This is a great feature and it makes widespread use of symmetric encryption feasible [4]. Symmetric encryption has a main security problem, which is maintaining the secrecy of the key. In the following subsections two of the most widely used symmetric ciphers, DES and AES, are described.

i. Data Encryption Standard (DES)

The Data Encryption Standard (DES), known as the Data Encryption Algorithm (DEA) by ANSI[1] and theDEA-1 by ISO[2], has been a worldwide standard for more than 20 years. Although it has been used for a long time, it has resisted well against years of cryptanalysis and is still secure against all but possibly the most powerful enemies [8].

DES is a block cipher that encrypts data in 64-bit blocks. A 64-bit block of plaintext goes in one end of the algorithm and a 64-bit block of ciphertext comes out the other end (see figure 2.1). The key length is 56 bits. The key is usually expressed as a 64-bit number, but every eighth bit is used for parity checking and is ignored. These parity bits are the

[1] The American National Standards Institute
[2] International Organization for Standardization

least-significant bits of the key bytes. The key can be any 56-bit number and can be changed at any time. A few of numbers are considered weak keys, and they should be avoided. All the security of the cipher rests within the key [8].

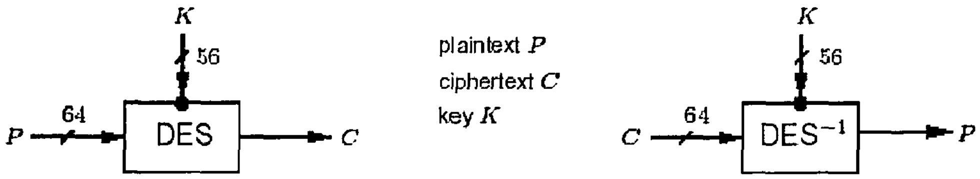

Figure 2.1. DES input-output [20]

At its simplest level, the algorithm is nothing more than a combination of the two basic techniques of encryption: confusion[3] and diffusion[4]. The fundamental building block of DES is a single combination of these techniques (a substitution followed by a permutation) on the text, based on the key. This is known as a round. DES has 16 rounds; in each round it applies the same combination of techniques on the plaintext blocks [8].

DES operates on a 64-bit block of plaintext. After an initial permutation, the block is broken into a right half and a left half, each 32 bits long. Then there are 16 rounds of identical operations, called Function f, in which the data are combined with the key. After the sixteenth round, the right and left halves are joined, and a final permutation (the inverse of the initial permutation) finishes off the algorithm.

In each round, the key bits are shifted, and then 48 bits are selected from the 56 bits of the key. The right half of the data is expanded to 48 bits via an expansion permutation, combined with 48 bits of a shifted and permuted key via an XOR, sent through eight S-boxes (or Substitution boxes, each S-box has a 6-bit input and a 4-bit output) producing 32 new bits, and permuted again. These four operations make up Function f. The output

[3] Confusion obscures the relationship between the plaintext and the ciphertext. This frustrates attempts to study the ciphertext looking for redundancies and statistical patterns. The easiest way to do this is through substitution [4],[8].

[4] Diffusion dissipates the redundancy of the plaintext by spreading it out over the ciphertext. The simplest way to cause diffusion is through transposition (also called permutation) [4],[8].

of Function f is then combined with the left half via another XOR. The result of these operations becomes the new right half; the old right half becomes the new left half. These operations are repeated 16 times, making 16 rounds of DES.

B_i is the result of the ith iteration, L_i and R_i are the left and right halves of B_i, K_i is the 48-bit key for round i, and f is the function that does all the substituting and permuting and XORing with the key, and a round looks like [8]:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \otimes f(R_{i-1}, K_i) \end{aligned} \tag{2.1}$$

DES decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed.

Since the time DES has been adopted as a federal standard, there have been some concerns about its level of security. These concerns, mostly, can be grouped into two areas: key size and the nature of the algorithm. The first concern is about the key size, if the key length is 56 bits, there are 2^{56} possible keys (approximately 7.2×10^{16} keys). Therefore a brute-force attack[5] does not appear to be practical. However, in July 1998, DES proved to be insecure, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption. They had used a special-purpose "DES cracker" machine. Another concern is the possibility of utilizing the characteristics of the DES algorithm for cryptanalysis purposes. The eight S-boxes, which are used in each iteration, have been the focus of concern. Because S-boxes' (and even the entire algorithm) design criteria were not made public (due to the possibility of weaknesses in S-boxes). But so far, the supposed weaknesses in the S-boxes have not been discovered by no one [4].

[5] The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

ii. Advanced Encryption Standard (AES)

NIST (National Institute of Standards and Technology) in 2001, published the Advanced Encryption Standard (AES). AES is a symmetric block cipher and it is intended to be the approved standard replacement of DES for a various applications. The two researchers who developed Rijndael for the AES are both cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen. The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits. Here a key length of 128 bits is assumed, which is possibly one of the most commonly implemented ones.

Rijndael was designed with these characteristics in mind [4]:

- To be resistance against all known attacks
- To be fast and have code compactness on a wide range of platforms
- To have simple design

A single 128-bit block, a square matrix of bytes, is the input to the encryption and decryption algorithm. This block is copied into the state array, which is modified at each stage of encryption or decryption. After the final stage, state is copied to an output matrix. Similarly, the 128-bit key is depicted as a square matrix of bytes. This key is then expanded into an array of key schedule words: each word is four bytes and the total key schedule is 44 words for the 128-bit key. The ordering of bytes within a matrix is by column. So, for example, the first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column, etc.

B. Public Key Encryptions

Public-key algorithms [4],[8] are designed in a way to use one key for encryption and a different but related key for decryption. All these algorithms have the following important characteristic:

- With only knowing the cryptographic algorithm and encryption key, it should be computationally impractical to determine the decryption key.

Some algorithms (such as RSA) also exhibit the following characteristic:

- One of the two related keys can be used for encryption, while the other one is used for decryption and vice versa.

A public-key encryption scheme has six elements [4]:

- Plaintext, which is the original comprehensible message or data. It is the input to the algorithm.
- Encryption algorithm, which performs different transformations on the plaintext.
- Public and private key, a pair of keys that have been chosen in a way that if one is used for encryption, the other one can be used for decryption.
- Ciphertext, the incomprehensible message, is the output of encryption algorithm. It depends on the plaintext and the secret key.
- Decryption algorithm, which takes the ciphertext and the matching key and produces the original plaintext.

Therefore, each user should generate a pair of keys to be used for the encryption and decryption of messages. Then one of the two keys is placed in a public register or other accessible file. This is the public key and the other key is kept private.

i. Rivest-Shamir-Adleman (RSA)

RSA scheme was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978 [4]. It makes use of an expression with exponentials. Plaintext is encrypted in blocks. Each block has a binary value less than some number n. n is product of two large odd primes and the block size must be less than or equal to $\log_2(n)$; in practice, the block size is k bits, where $2^k < n \leq 2^{k+1}$. For some plaintext block M and ciphertext block C, encryption and decryption have the following form:

$$\begin{aligned} C &= M^e \bmod n \\ M &= C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n \end{aligned} \tag{2.2}$$

The value of n must be known by both sender and receiver. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public key encryption algorithm with a public key of $KU = \{e,n\}$ and a private key of $KR = \{d,n\}$. To satisfy public key encryption, this algorithm should met the following requirements [4]:

1. Finding the values of e, d, n such that $M^{ed} = M \bmod n$ for all $M < n$, should be possible.
2. Calculating M^e and C^d for all values of $M < n$, should be easy.
3. Determining d given e and n, should be impractical.

The elements of the RSA scheme are [4]:

p, q, two prime numbers	(private, chosen)
$n = pq$	(public, calculated)
e, with $\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$	(public, chosen)
$d = e^{-1} \bmod \phi(n)$	(private, calculated)

where $\phi(n)$ is the Euler function, which is the number of positive integers less than n and relatively prime to n. Since RSA is a public key cryptosystem it cannot provide unconditional security. The security of RSA relies on the difficulty of factoring large integer numbers [5].

2.1.2 Non-Conventional Cryptography

Cryptography has hard and complex algebraic and number theoretical problems and recently, numerous techniques and methods have been proposed to address these issues. Some techniques are related to polynomial interpolation, discrete Fourier transforms, polynomial approximation and computational intelligence methods [21]. The focus of this thesis is on the latter methods, therefore a brief introduction to Neural Networks (NNs) is provided in the following section.

A. Neural Networks

In past, computing was based on the concept of programmed computing in which based on the currently dominant architecture, algorithms were designed and subsequently implemented. An alternative view, inspired from biological systems, was proposed with Artificial Neural Networks (ANNs). It is well known that biological system functionality is based on the interconnections of specialized physical cells called neurons. ANNs simulate this procedure, and they are a mathematical model with the ability to learn, adapt, generalize or to cluster and organize data. Compared to other techniques, all these operations are based on the parallel processing of data and can be quite advantageous and fast [5].

A neuron is an information-processing unit and it is a fundamental factor in the operation of a NN. Figure 2.2 shows the model for a neuron. Three basic elements of the neuron model can be identified as [16]:

1. A set of synapses or connecting links, each of them has its own weight or strength. Specifically, a signal x_j (at the input of synapse of neuron j, which is connected to neuron k) is multiplied by the synaptic weight w_{kj}.
2. An adder for summing the input signals. They are weighted by the respective synapses of the neuron. This operations form a linear combiner.
3. An activation function, which limits the amplitude of the output of a neuron.

The threshold θ_k in the model of a neuron shown in figure 2.2 lowers the net input of the activation function. The effect of the threshold is represented by performing two operations: 1) it adds a new input signal fixed at -1, and 2) it adds a new synaptic weight equal to the threshold θ_k. Using a bias term (the bias is the negative of the threshold) instead of a threshold, may increase the net input of the activation function [16].

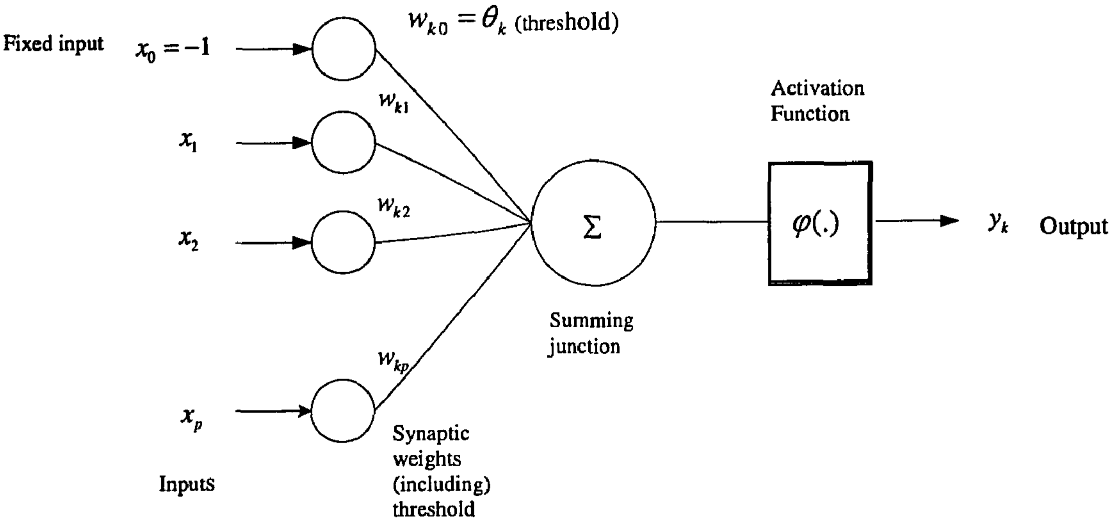

Figure 2.2. Nonlinear model of a neuron

In mathematical terms, a neuron k can be described by writing the following pair equations:

$$v_k = \sum_{j=0}^{p} w_{kj} x_j \tag{2.3}$$

and

$$y_k = \varphi(v_k) \tag{2.4}$$

where $x_1, x_2, \ldots, x_p$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{kp}$ are the synaptic weights of neuron k; v_k is the linear combiner output; θ_k is the threshold; $\varphi(.)$ is the activation function; and y_k is the output signal of the neuron.

A strict definition for an ANN is a structure composed of a number of interconnected units (artificial neurons). Each unit has an input/output characteristic and implements a local computation or function. The output of any unit is determined by its I/O characteristic, its interconnection to other units, and possibly external inputs. Although it is possible to hand craft the network, they usually develop an overall functionality through one or more forms of training. This definition covers a broad family of networks. The network topology, the training algorithm used and the neuron characteristics determine the overall functionality of the network. One very important type of ANNs are the feedforward ones. In this kind of networks, all paths lead to one direction. Furthermore the neurons can be split in layers. Especially in the Multilayer Feedforward Networks, the inputs form an input layer, while the output neurons form the output layer. All other neurons are assigned to a number of hidden layers. In a layer, each neuron is fully connected to all other neurons in the next layer. This structure makes it possible to describe this kind of networks with a series of integers that represent the number of neurons at each layer. For example a network with a topology 4-5-5-1 is a network with four neurons at the input layer, five neurons in each of the two hidden layers and one neuron in output layer [5].

The operation of such networks consists of iterative steps. At the beginning, the states of the input layer neurons are assigned to generally real inputs, and the remaining hidden and output layer neuron are passive. In the next step the neurons from the first hidden layer collect and sum their inputs and compute their output. This procedure is propagated to the following layers until the final outputs of the network are computed.

The computational power of NNs is based on the fact that they can adapt to a specific problem. It has also been proven [22, 23] that standard feedforward networks with only a single hidden layer can approximate any continuous function uniformly on any compact set and any measurable function to any desired degree of accuracy. Therefore, inadequate learning, insufficient number of hidden units or the lack of a deterministic relationship between input and target can be the reasons behind lack of success in applications [5].

The training process of feedforward networks is based on patterns for which the desired output is known a priori. A training set T of P patterns can be defined as:

$$T = \left\{ (x_k, d_k) \left| \begin{matrix} x_k = (x_{k1}, \ldots x_{kn}) \\ d_k = (d_{k1}, \ldots d_{km}) \end{matrix} \right., k = 1, \ldots, P \right\} \tag{2.5}$$

where x_k is the input vector of the kth training pattern and d_k is the vector of the desired output of the specific pattern. The aim of adaptation is to assign some values to the free parameters of the network W. These values are set such that the output of the network based on that set of weights will be the desired one (at the beginning the weights are assigned random values), i.e. it holds that:

$$y(W, x_k) = d_k, \qquad k = 1, \dots, P \tag{2.6}$$

The adaptation procedure starts by presenting all the patterns to the network and computing a total error function E defined as:

$$E = \sum_{k=1}^{P} (E_k) \tag{2.7}$$

where E_k is the partial network error with respect of the kth training pattern, and is computed by summing the squared discrepancies between the actual network outputs and the desired values of the kth training pattern, thus:

$$E_k = \frac{1}{2} \sum_{i=1}^{m} (y_i(W, x_k) - d_{ki})^2 \tag{2.8}$$

Each full pass of all the network patterns is called a training epoch. The aim of the adaptation method is to succeed in minimizing the total error function and in this case the problem is a non-trivial minimization problem. One very popular method for doing this task is the Back Propagation method, which is based on the well known steepest descent method. The Back Propagation learning process applies small iterative steps, which correspond to the training epochs. At each epoch t the method updates the weight values by the relation:

$$w_{ji}^{t} = w_{ji}^{t-1} - \Delta w_{ji} \tag{2.9}$$

where w_{ji}^{t} corresponds to the weight of the connection from neuron i to the non-input neuron j at epoch t while Δw_{ji} corresponds to the increment of the weights. The latter is proportional to the gradient of the error function $E(w)$ at the weight w^{t-1}:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} (w^{t-1}) \tag{2.10}$$

where η, $0 \le \eta \le 1$ is called learning rate and measures the influence of the gradient, computed at the specific epoch, on the general adaptation. To compute the gradient $\frac{\partial E}{\partial w_{ji}}(w^{t-1})$ firstly the sum rule is used to simplify the procedure to the computation of the sums of gradients of the partial error functions:

$$\frac{\partial E}{\partial w_{ji}}(w^{t-1}) = \sum_{k=1}^{P} \frac{\partial E_k}{\partial w_{ji}} \tag{2.11}$$

At the next step the rule for the composite function derivative can be used:

$$\frac{\partial E_k}{\partial w_{ji}} = \frac{\partial E_k}{\partial y_i} \frac{\partial y_i}{\partial \xi_j} \frac{\partial \xi_j}{\partial w_{ji}} \tag{2.12}$$

where:

$$\xi_j = \sum_{j \in j_m} w_{ji} y_j \tag{2.13}$$

and y_j is the output of neuron j. Obviously the $\frac{\partial y_j}{\partial \xi_j}$ and $\frac{\partial \xi_j}{\partial w_{ji}}$ can be easily computed and they depend on the activation function. The remaining partial derivative $\frac{\partial E_k}{\partial y_j}$ is computed at the output layer and its value is back-propagated through the network to the input neurons (the method derives its name from this procedure).

When the overall error value drops below some pre- determined threshold, the whole process will stop. At this point, the network has learned the problem well enough. It should be considered that, the network will asymptotically approach the ideal function, and cannot exactly learn it. The speed of the method is defined by the total number of epochs required [5].

Another important type of ANNs is Recurrent Neural Networks (RNNs). A RNN is different from a feedforward neural network, and it has at least one feedback loop. For example, a recurrent network may consist of a single layer neurons (or with hidden layer) with each neuron feeding its output signal back to the inputs of all the other neurons, there is no self –feedback loops. Self-feedback refers to a situation where the output of a neuron is fed back to its own input. The presence of feedback loops has a profound impact on the learning capability of the network, and on its performance. Moreover, the feedback loops involve the use of particular branches composed of unit-delay elements (denoted by z^{-1}); the nonlinear nature of the neurons results in a nonlinear dynamical behavior. A key role in the storage function of a recurrent network is played by nonlinear dynamics [16].

After this brief introduction to NNs, in the following sections, their applications in cryptography are discussed.

i. Neural Network Approach In The RSA Cryptosystem

This paper [5] is a first study of using NNs in RSA cryptosystem. The computational security of RSA is based on the difficulty of factoring large integers. In order to break RSA cryptosystem, it is enough to factorize N, where N is the product of two large prime numbers, $N = p.q$. This is equivalent to calculate $\varphi(N) = (p-1)(q-1)$ where φ is the Euler function. In this paper, feedforward NNs are trained to compute the Euler function. They have used various training methods such as the Standard Back Propagation (BP), the Back Propagation with Variable Stepsize (BPVS), the Resilient Back Propagation (RBP), and the On-Line Adaptive Back Propagation (OABP). All these methods have been extensively tested with a wide range of parameters. They have concluded that the training method does not play a significant role in tackling the particular problem. On the other hand, a crucial role is being played by the network architecture and the normalization portion of the training algorithm used. They have succeeded in reducing the network architecture as much as it was permitted by each training method, by applying two recent proposed techniques: the deflection and the function "stretching". They think that the NN approach in problems related to RSA cryptosystem is promising although many problems have to be solved and many future work needs to be done.

ii. MultiLayer Perceptron Networks In Symmetric Block Ciphers

In this paper [7], the applicability of using MultiLayer Perceptron (MLP) Networks in symmetric block ciphers is explored. A prototype symmetric block cipher is proposed. It employs a MLP Network that decides on the algorithm used for encryption. The MLP Network is in turn dependent on the secret key. By employing a mutating algorithm comprising of cryptographically proven modular arithmetic and feistel networks, it is hoped that such a symmetric block cipher will be resistant to modern cryptanalytic attacks such as differential and linear attacks. Even though the proposed cipher may seem to be resistant to linear and differential cryptanalysis, more in depth studies should be made. The effect of other forms of cryptanalytical attacks, including interpolation attack, differential-linear attack, related-key attack, timing attack, partition attack and power attack, should be studied. The feasibility of employing MLP based block ciphers as both software and hardware solution should also be explored, as well as improvements in the speed of the algorithm in those implementations.

iii. MultiLayer Perceptron Networks In Public Key Cryptography

In this paper [9], the applicability of using a MultiLayer Perceptron (MLP) Network in public key cryptography is investigated. A system using the properties of MLP Networks is proposed and its security is examined. A 64*64 MLP Network is used by both parties for public key cryptography. The parties should choose their own private keys. The theory and experimental results show that MLP Networks are useful in public key cryptography. Further research in this are will include applying the MLP Network as a possible authentication tool for verification uses.

iv. MultiLayer Perceptron Networks As A One-Way Hash Function

In this paper [10], the applicability of using a MultiLayer Perceptron (MLP) Network as a possible hash algorithm is investigated. The difficulty of recovering an input from as MLP Network hashed output is presented. Important features of good hash algorithms such as resistance to birthday attacks and collision free hashing are explored with regard to the MLP Network. Possible advantages of using such an arrangement over existing hash algorithms are mentioned. The MLP Network structure developed for one-way hashing consists of a hidden layer and an output layer. The hidden layer contains 64 neurons with 641 inputs including the bias. The weights of the hidden neurons are truncated to three decimal places, while α, the slope of sigmoidal function (activation function), is set to 1. The output layer contains of 128 neurons with 65 inputs including the bias, with α set to 1000. The MLP Network thus structured is shown to be pre-image resistant, 2nd pre-image resistant and collision resistant. The proposal of applying the MLP Network as a hashing algorithm has several advantages over existing algorithms, as it can easily be adjusted to produce variable number of output bits, and it can also be initialized uniquely for multiple purposes.

v. Theory of Interacting Neural Networks

In this paper [24] and some other similar papers [11],[25], a connection between the theory of NNs and cryptography is presented. A new phenomenon, namely synchronization of NNs, is leading to a new method of exchange of secret messages. Two artificial networks being trained by the Hebbian learning rule on their mutual outputs develop an anti-parallel state of their synaptic weights. The synchronized weights are used to construct an ephemeral key exchange protocol for the secure transmission of secret data. The complexity of the generation of the secure channel is linear with the size of the network. An attacker who knows the protocol and all details of any transmission of the data finds it difficult to decrypt the secret message.

In [12] the security of the new key exchange protocol proposed in the above mentioned paper, which is based on mutually learning NNs, is analyzed. This is a new potential source for public key cryptographic schemes, which are not based on number theoretic functions, and have small time and memory complexities. This paper, the scheme is analyzed and it is explained why the two parties converge to a common key and why an attacker using a similar NN is unlikely to converge to the same key. However, in the second part of the paper, it is showed that this key exchange protocol can be broken in three different ways, genetic attacks, geometric attacks and probabilistic attacks, and thus it is completely insecure.

vi. Neural Networks and Pseudorandom Stream Generators

This paper [13] presents novel techniques, which rely on ANNs architectures, to strengthen traditional generators such as IDEA[6] and ANSI[7] X.9 based on 3DES and IDEA. Additionally, this paper proposes a non-linear test method for the quality assessment of the required non-predictability property, which relies on feedforward NNs. This non-predictability test method along with commonly used empirical tests based on statistics is proposed as a methodology for quality assessing strong pseudorandom stream generators. By means of this methodology, traditional and NN based pseudorandom stream generators are evaluated. The results show that the proposed generators behave significantly better than the traditional ones, in terms of non-predictability.

vii. A New Chaotic Neural Encryption/Decryption Network

In this paper [26], a new chaotic Neural Network[8] and its VLSI architecture for digital signal encryption and decryption are proposed. According to a binary sequence generated from a chaotic system, the biases and weights of neurons are set. The chaotic NN can be used to encrypt digital signal. The network's features are as follows: 1) high security, 2) no distortion, 3) suitable for system integration. The MATLAB simulation results have indicted that the algorithm can make raw images chaotic by the sense of sight and the disordered images have high fractal dimensions by the quantitative measure.

[6] International Data Encryption Algorithm, a symmetric block cipher.

[7] American National Standards Institute

[8] A network is called Chaotic Neural Network if its weights and biases are determined by a chaotic sequence.

viii. Neural Networks and Their Cryptographic Applications

Identification is a useful cryptographic tool. Since the appearance of zero-knowledge theory, several interactive identification schemes have been proposed (in particular Fiat-Shamir and its variants, Schnorr). These identifications are based on number theoretical problems. More recently, new schemes appeared with the particularity based on $\mathcal{NP}$ complete problems: PKP (Permuted Kernels Problem), SD (Syndrome Decoding) and CLE (Constrained Linear Equations). This paper [27] presents a new linear $\mathcal{NP}$complete problem, which comes from NNs and learning machines: the Perceptron problem. There are some constraints, m vectors X^i of $\{-1,+1\}^n$, and we want to find a vector V of $\{-1,+1\}^n$ such that $X^i.V \geq 0$ for all i. Next, they provided some zero-knowledge interactive identification protocols based on this problem, with an evaluation of it security. Eventually, those protocols are well suited for smart card applications.

2.2 Conclusion

The expansion of worldwide communications and the increased digitalization of our society can make information more vulnerable to abuse. This misuse can take many forms: eavesdropping of sensitive data (for example, e-mails, company secrets), unauthorized modification of information (changing money transfer s between banks, or introducing viruses into computer software), stealing information (images or audiovisual recordings), use of electronic services without paying for them, repudiating electronic orders, bringing down computer systems or networks, etc. These risks require adequate security measures to protect electronic information systems. It is clear that in an electronic world physical security or personal security by itself will not be sufficient.

The use of cryptography for protection the secrecy of information is as old as writing itself. The basic idea consists of applying a complicated transformation to the information to be protected. As it was discussed in this chapter, many different cryptographic algorithms are available. When designing or selecting a cryptographic algorithm several criteria are taken into account: security, which relates to the value of the information to be protected; performance, both in hardware and software; cost of design or licensing, cost of implementation and key management; availability, which is often related to commercial constraints or national security constraints; standardization, which makes the equipment reusable to its maximum and provides some security guarantees; error propagation and synchronization aspects, which are very important for encryption where channels of communication are unreliable [1].

NNs are an apparently valid computational model of how human brain operates and they have attracted a lot of attention in the last 60 years. Today the area of research of NNs is extremely active, and attracts researchers from a wide variety of backgrounds such as, Biology, Medicine, Psychology, Mathematics, Computer Science, etc [1]. Not surprisingly, researchers have also tried to use NNs in cryptography. Application of NNs in cryptography does not have a long history and few researches have been done, which were mentioned in this chapter. In next chapter a novel NN based block cipher [17] is described in details.

CHAPTER 3: The Block Cipher Design Using RNNs

3.1 Introduction

There are several security-related requirements for which improved cryptography techniques need to be considered. The most important requirement is that cryptography must ensure the confidentiality of the data and hence protect the privacy of the information. Next to confidentiality, data authentication is another important requirement. Nowadays, we are facing an ever-increasing denial of service attacks, mainly attributed to the fact that many information systems have no authentication mechanism. The identity of the user at the other side of the network must be correctly verified to ensure the service request is made by a legal customer who possesses the proper authorization. Data integrity is yet another important requirement. It is often a matter of great importance to ensure that information contents have not been tempered with during a transfer request made between an authorized user and an information system. Any cryptography technique needs to provide the data integrity service to ensure that no message contents can be tempered without being detected. All these requirements make it essential to investigate means to design cryptography techniques that can have the capability to resist different types of cryptanalysis attacks, while providing data integrity, and authentication guarantees.

The computational costs for cryptanalysis attacks are decreasing dramatically and the majority of existing cryptography techniques find themselves less secure. The question of how to increase the security level without degrading the performance becomes imperative, and hence, an efficient and cost effective answer is always desirable.

In the previous research [17], it was proposed to use the powerful parallel computing capability of ANNs as a potential solution for the data encryption improvement. A novel block cipher design using neural network as a parallel computing technique for cryptography implementation, was introduced. The block cipher was implemented in two steps: (i) a NN back propagation learning procedure was used for key extension. The secret key was presented to the network as the training set and the network weights and biases were initialized according to the secret key through the supervised learning procedure. The resulting network with the secret initial state was the extended secret key to be later used for data encryption, and (ii) the learning procedure was controlled by some neural network parameters to generate the chaotic cipher text and synchronize the weights/biases states during bulk data encryption/decryption. In the next section a complete description of this method is provided, all the descriptions are taken from [17].

3.2 Block Cipher Design Using RNNs

The proposed block cipher design is based on real-time recurrent neural networks (RRNN) shown in figure 3.1. The RRNN has a multilayer structure with the following two constraints: (i) the dimension of the input vector n is twice that of the output vector m, and (ii) one of the hidden layers has only one neuron with an output denoted by ξ. The block cipher operates in two stages: key extension and data encryption/decryption.

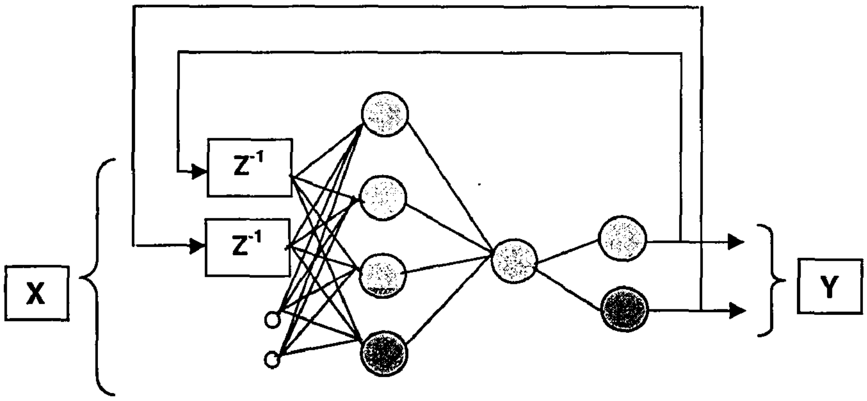

Figure 3.1 Proposed recurrent network for block cipher design

3.2.1 Key Extension

Suppose we have two users of a communication method with an identical block cipher based on a NN similar to that in figure 3.1. They will exchange a secret key S that contains the following three parts of information: (i) the input vector X, (ii) the training target Y, and (iii) the critical value of the self-adaptive procedure α. Vectors X and Y will then be presented to the NN for training. The training algorithm is based on error back-propagation learning. Here, the purpose of the training process is to make the NN detect, store or even "remember" the secret key information. The well-trained NN parameters will be kept unrevealed and become the extended secret key for the subsequent encryption and decryption procedures. The last actual output of the network during the key extension will be the initial vector, M_0, for the encryption. It is commonly assumed that the weight distribution of the hidden layers is chaotic and unpredictable without the knowledge of the training data (i.e. the original secret key). Therefore, it is not feasible for a cryptanalyst to analyze the extended key, i.e., network weight matrix. By changing the length of the secret key and the dimension or the hierarchy of the hidden layers, the user can adjust the security level accordingly. Hence, a major advantage of the proposed block cipher design is its capability to release constraints imposed on the length of the secret key.

3.2.2 Encryption

The structure of the block cipher design ensures that among the hidden layers of the neural network, there exists at least one that has only one neuron (denoted here as neuron ξ). This feature can be used to decompose the feedforward operation of the neural network into two functions F_1 and F_2. In such decomposition, F_1 is the feedforward operation over the weight and bias matrices performed from the input layer to neuron ξ, and F_2 is the same operation performed from neuron ξ to the output layer. The functions F_1 and F_2, are then used in the encryption process that consists of two steps: (1) cipher text generation, and (2) one-epoch training.

A. Cipher Text Generation

At first, a plain text is mapped to vectors $M_{i(i=1,\dots,n)} = \{M_1, \ M_2, \ M_3, \ \dots \ M_n\}$ according to the dimension of input vectors. The first message vector M_1 is combined with the initial vector M_0 from the key extension procedure to build the following initial input vector

$$X_1 = (M_0 \| M_1) \tag{3.1}$$

where || denotes a vector concatenation operator, this is, two $(n\times1)$ vectors M_0 and M_1 are concatenated to form a $(2n\times1)$ vector. Next, X_1 is presented to the neural network to produce both the intermediary neuron output V_1 in the hidden layer and the output Y_1. Then, the error signal is calculated as $E_1 = M_1 - Y_1$, where M_1 is the target of the identity mapping. Finally, E_1 and V_1 are considered as the first block of the cipher text referred to as $C_1\{V_1, E_1\}$.

B. One Epoch Training

After the first cipher text block C_1 is constructed, the neural network can be trained for one-epoch using (3.1) as the input vector and M_1 as the training target. From the second and all following plain text blocks, the preceding time instant output $Y_{i-1}(i = 2,\dots,n)$ of the neural network is combined with the current plain text block M_i to yield the current input vector. In other words, the input vectors can be built according to:

$$X_i = (Y_{i-1} \| M_i) \qquad i = 1,2,\dots,n \tag{3.2}$$

The above two steps of encryption are repeated to generate values for V_i and Y_i and hence train the neural network for one-epoch at a time. In fact, the above encryption procedure will result in a block cipher working in the CBC mode implicitly as shown in figure 3.2.

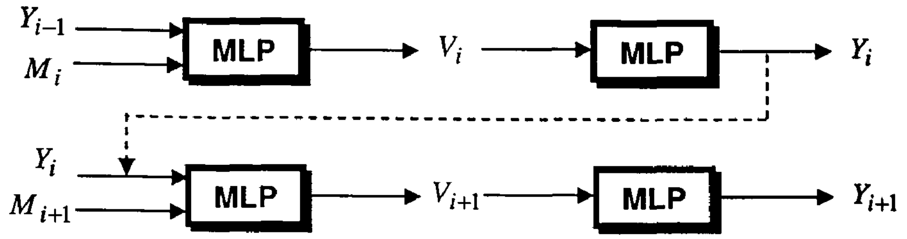

Figure 3.2 Block Cipher in CBC mode, MLP: Multi Layer Perceptron

To summarize the above procedure, the cipher text blocks C_i are constructed as follows:

$$V_i = F_1(X_i) \tag{3.3}$$

$$Y_i = F_2(V_i) \tag{3.4}$$

$$E_i = M_i - Y_i \tag{3.5}$$

$$ST = C\{V_i, E_i\} \tag{3.6}$$

where ST refers to the cipher text at instant i. The recurrent neural network structure in figure 3.2 is the schematic representation of equations (3.3)-(3.6). The first hidden layer defines F_1 in (3.3). The second hidden layer has one neuron ξ. The third hidden layer implements function F_2 that computes the output Y_i as in (3.4). Finally the output at time instant i is fed back through a zero order hold to construct the input to the network at the following time instant.

3.2.3 Decryption

The decryption procedure proposed in this work is illustrated in figure 3.3. The procedure works in a similar fashion as that of the encryption. When the block cipher receives the cipher text $C_i\{V_i, E_i\}$, the output Y_i is computed as

$$Y_i = F_2(V_i) \tag{3.7}$$

Next, the original plain text block can be restored using

$$M_i = Y_i + E_i \tag{3.8}$$

After the message block M_i is restored, the one-epoch training step is performed with $X_i = (Y_{i-1} \| M_i)$ as the input vector and M_i as the training target.

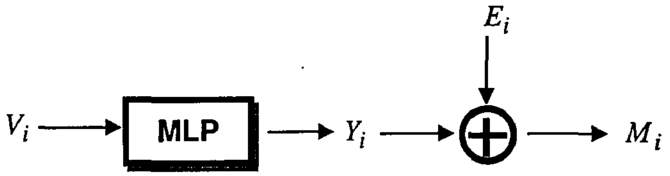

Figure 3.3 Decryption process

The output V_i of the final block can be used as the Message Authentication Code (MAC) for the whole cipher text. After calculating Y_i from V_i during decryption, we can produce M_i and hence reconstruct $X_i = (M_i \| Y_i)$ once again. Then, we compute

$$V_i' = F_1(X_i) \tag{3.9}$$

Next, we can compare V_i' with V_i to verify data integrity and authentication. In general, at the end of the data encryption/decryption stage, the Cipher Block Chaining-Message Authentication Code (CBC-MAC) [4] is prepared (or examined if already exists) to ensure data integrity. The CBC mode encryption and decryption is illustrated in figure 3.4, where P_i are the plain text blocks and C_i are the cipher text blocks. CBC-MAC is a simple method that uses the last encrypted block as the MAC for the cipher text chain.

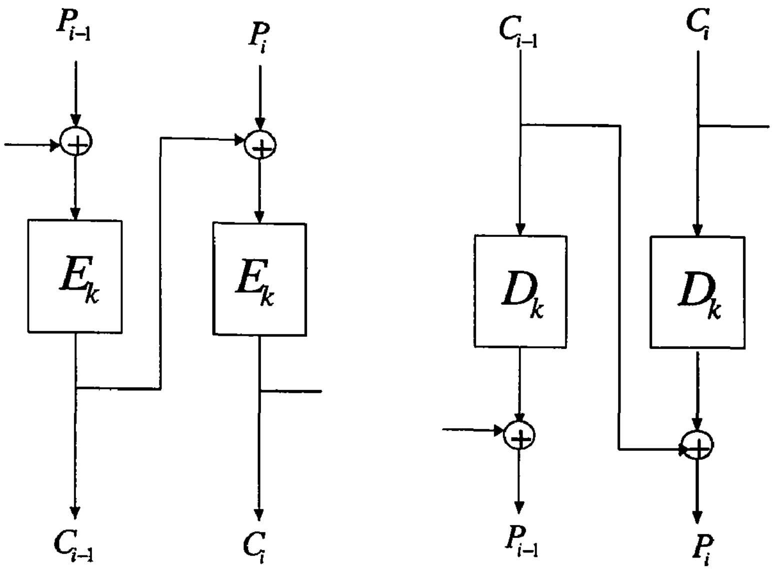

Figure 3.4 CBC mode encryption (left) and decryption (right)

3.2.4 RRNN and Block Cipher Design

By means of the RRNN (figure 3.1), the block cipher uses the forward dynamics (3.3), (3.4), (3.5) and (3.6) to generate the cipher text and Message Authentication Code (MAC). In specific terms, the output of the network forward dynamics is computed as

$$Y_j(n+1) = \varphi\left(\sum_{i=A\cup B} w_{ji}(n)U_i(n)\right) \qquad \text{for} \qquad j \in B \tag{3.10}$$

where φ is a nonlinear activation function, and the variable w_{ji} refers to the synaptic weight of neuron j. In (3.10), $U_i(n)$ is the input vector to the RRNN defined as [16]:

$$U_i(n)\begin{cases} X_i(n) & if \;\; i \in A \\ Y_i(n) & if \;\; i \in B \end{cases} \tag{3.11}$$

A denotes the set of indices i for which $X_i(n)$ is an external input, and B denotes the set of indices i for which $U_i(n)$ is the output of the neuron. Also, the term representing the argument of the linear activation function in (3.10) is the neuron internal activity function V_i in (3.3). Therefore, at every time step n, starting at $n=0$, we use the dynamic equations in (3.11) and (3.12) to compare the output values of N neurons; hence we use these output values to compute the external input values $U_i(n)$ in (11) for $i \in A \cup B$. We choose the initial values of the weight $w_{ji}(0)$ from a set of uniformly distributed random numbers.

Next, we define the dynamic process for updating the network weights in real time by means of the following triply index [16]:

$$\vartheta_{kl}^{j}(n+1) = \varphi'\left(V_j(n)\right)\left[\sum_{i \in B} w_{ji}(n)\vartheta_{kl}^{j}(n) + \delta_{kl}U_l(n)\right] \tag{3.12}$$

where $j \in B$, $k \in B$, $l \in A \cup B$, and $\varphi'(.)$ is the derivative of the nonlinear activation function. In (3.12), δ_{kl} is the Kronecker delta equals to one when $k=l$ and zero otherwise. The triply index is initialized such that $\vartheta_{kl}^{j}(0)=0$. We use the index in (3.12) to update the RRNN network weights as follows

$$\Delta w_{ki}(n) = \eta \sum_{j} E_j(n)\vartheta_{kl}^{j}(n) \tag{3.13}$$

where Δw_{kl} denotes the update to the weight w_{kl}, and the parameter η refers to the learning rate of the network. In (3.13), the error function E_j at time instant n is computed as

$$E_j(n) = M_j(n) - Y_j(n) \tag{3.14}$$

Finally, the weight w_{kl} is updated in accordance with

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \tag{3.15}$$

Both forward and backward dynamics are varying in time to ensure that the learning procedure of the RRNN has the capability to detect temporal patterns of the training data. Consequently, the block cipher can prepare MAC (section 3.2.3) to maintain both the data integrity and authentication.

3.3 Security Guarantee

The proposed block cipher design provides three types of cryptographic services: 1) data integration, 2) data authentication and, 3) data confidentiality. In addition, two possible types of attacks against the block cipher need to be investigated: i) attack against the Message Authentication Code, and ii) attack against the data encryption scheme.

3.3.1 Cryptographic Analysis against MAC

Several features of the Message Authentication Code can be viewed as potential targets for the cryptanalysis attacks. Among those features are: 1) Message Authentication Code needs to be a one-way function. For any given input x, it is easy to compute the authentication code by the secure hash function H. But it is computationally not feasible to arbitrary guess x from the message authentication code even if H is known, 2) Message Authentication Code needs to be Collision-resistant. It is not computationally feasible to find a pair (x, y) such that $H(x) = H(y)$, and 3) Message Authentication Code needs to be capable of data authentication. Only the secret key owner can prepare or verify the code because the hash value is encrypted by secret key.

Most of the cryptanalysis attacks against MAC focus on the collision resistance feature. The attacker tries to substitute the text x with the alternate text x' such that $H(x) = H(x')$. In doing so, the attacker can target either the key space of the MAC or its actual value. Also, without attempting to recover the secret key, the attacker may try to find a message that matches a given MAC value, and then use that message to replace the original one. When a cipher text message $C_i\{V_i, E_i\}$ in (3.6) is changed, one of the following two scenarios arises: i) either E_i or V_i are changed, or ii) both E_i and V_i are changed.

Now suppose that either E_i or V_i are changed during a cryptanalysis attack. The decryption process will produce ${}^*M_i$ from ${}^*C_i$ according to (3.7) and (3.8). Then the attacker will calculate $^{*}V_i$ according to (3.9). Furthermore, due to the fact that the value of $^{*}V_i$ and V_i will not match, data corruption may be detected. However, it is possible for the attacker to choose a cipher text $^{*}C_i\left\{^{*}V_i,{}^{*}E_i\right\}$ so as to pass the Message Authentication Code (MAC) check. Yet, this attack will be detected when the next MAC is checked because of the CBC mode. This is due to the fact that $^{*}Y_i$ is not only used for the MAC check of the current block, but also for one step ahead check. In other words, the input vector will be changed from $X_{i+1}=Y_i\|M_{i+1}$ to $^{*}X_{i+1}={}^{*}Y_i\|M_{i+1}$ and the data integrity corruption of cipher text C_i will be detected by the MAC check of the next cipher text C_{i+1}. Hence, the attacker will be forced to identify a chain of messages to replace the whole document of plain text for the attack against the CBC-MAC to be successful. Also, if the length of the plaintext is n bits, the effort will require approximately 2^n operations.

Now let us consider the case of an attack against the key space of Message Authentication Code. If the attacker successfully determines the secret key, he can generate a valid MAC value for any given message. If the attacker has the knowledge of some sets of both the plaintext and cipher text, he will try every possible secret key to generate MAC. By comparing the results, he may try to break the MAC and the block cipher. Suppose the total key size of the extended key is k bits and the length of the plaintext is n. Since the MAC is usually a many-to-one mapping, for the first round attack, it is expected the attacker will find about $2^{(k-n)}$ matching keys. It is necessary for the attacker to perform multiple rounds of attacks. For the second round, the attacker will search within the remaining $2^{(k-n)}$ keys and he will probably find $2^{(k-2n)}$ keys, and so on. Such effort will be reduced rapidly for the consecutive rounds. The overall effort of this type of attack will be roughly searching 2^k keys. In summary, the effort of the attacks against Message Authentication Code will be in finding $\beta = \min(2^k, 2^n)$ keys. According to modern cryptanalysis, the strength of the block cipher is required to be at least 128 bits.

3.3.2 Cryptanalysis against Data Confidentiality

The encryption procedure can be viewed as a nonlinear mapping and the cipher text is the nonlinear transformation of the plain text. If this transform function is static, the nonlinear equations can be possibly solved if the cryptanalyst has large volumes of plain text with the corresponding cipher text available. In comparison to other existing algorithms, the extended key length k of the proposed block cipher is much longer. Because the block cipher makes use of the learning procedure of NN to encrypt data, the plaintext blocks are assumed to be encrypted by a key stream. As a result, the extended key length k should be the total sum of all these keys within the same key stream period. The longer the key stream period is, the longer the extended key length k will be. This will result in a stronger block cipher. If we can guarantee that the learning procedure will not converge quickly, the block cipher can generate long period key stream. Consequently, the nonlinear transform function should be dynamic when it is applied for data encryption. The feedforward dynamics of RRNN must keep varying in time to provide security protection of the plain text. Furthermore, since the learning procedure usually tends to be convergent, cryptanalysis attack based on the stability of NN during learning may be an issue of importance. This will be studied in the following section.

A. Attack against Data Confidentiality

Let G denotes the set of plain texts, Z be the set of local and global minima, and L be the largest invariant set in Z. L will contain all of the possible points at which the solution might converge and the trajectory can be trapped. Assume L contains only one fixed-point y. A cryptanalyst will train the block cipher with the known plaintext repeatedly until the block cipher converges to L. One possible method for the cryptanalyst to achieve this goal is to insert as an input a large amount of known plaintexts before the secret plaintext is introduced. After the block cipher is stable, all the secret plaintexts input that belongs to G will be convergent to this fixed point. Although the cryptanalyst has no knowledge of the weight matrix and the initial state of the block cipher, he can obtain the convergent point y in L by means of the known plaintext. Then the cryptanalyst can restore the following secret plaintext M by the error signal e using $M = y + e$. It shows that the stability of the neural networks will eventually help the cryptanalyst to break the block cipher without the knowledge of the weight matrix. To resist such an attack, the learning procedure needs to guarantee that convergence will not drift towards an invariant set L after the training of large volume of plain texts. This consideration is directly related to the stability problem of neural networks.

B. Stability Problem of NNs during Learning

The RRNN can be modeled as nonlinear dynamic system. Also, the direct Lyapunov method [27] is applicable to the stability analysis of neural networks. The key is to apply the direct method to find the Lyapunov function that ensures the boundness of the network error during the learning process [16]. In general, it is not possible to identify a Lyapunov candidate for the recurrent back-propagation algorithm [27]. Hence, the direct method of Lyapunov will not be of any assistance in analyzing the security of the block cipher. Alternatively, through a "local" analysis of the learning procedure of neural networks, we can assume that the local stability of the forward propagation is a sufficient condition for the local stability of the backward propagation and vice versa [28]. Consequently, we need to guarantee the instability of the backward propagation, (3.11), so that the forward propagation (used to generate the cipher text) is ensured to be chaotic and unpredictable. According to (3.11), the instability of the backward propagation is related to both the error signal and the weight matrix. An instantaneous estimate of the gradient has been used to approximate the true gradient curve of the cost function in order to perform real-time learning. If the learning rate η is set to a large value, a small mismatch between the output and the learning target will have a dramatic effect on the weight update process and will hence cause the forward propagation to be unstable, i.e., chaotic. This chaotic oscillation of the learning behavior can be generated deliberately to provide the security protection for the data.

3.3.3 Self-adaptive Behavior of the Block Cipher

The self-adaptive function of the block cipher is a necessary component to resist possible cryptanalysis attacks. The algorithm detects the trend of the learning procedure via monitoring the mean squared error performance function (MSE) and then adjusting the learning rate by a Multiplicative-Increase Gradual-Decrease (MIGD) method, i.e., the TCP Vegas congestion control protocol [29]. At first, a low-pass filter for the *MSE* learns the trend detection as follows

$$T(k) = \delta T(k-1) + (1-\delta) * MSE(k) \tag{3.16}$$

where δ is often selected between 0 and 1, *T*(*k*) is the output of the low-pass filter of MSE at time *k* and the initial state *T*(0) is set to be zero. The learning stop condition (referred to as *the learning goal*), MSE^{stop} is defined as:

$$MSE^{stop} \leq \alpha \tag{3.17}$$

where α is the critical value of *T*(*k*). The learning rate will adapt itself according to the MIGD method and based on one of the following three cases:

1) Case 1: $T(k) \leq \alpha$

The condition shows that the learning procedure tends to be convergent to the learning goal. To avoid the stability of the learning and restore the chaotic behavior, the learning rate η is increased aggressively by a factor λ, i.e., $\lambda = 2$. In that case, we have: $\eta = \lambda\eta$.

2) Case 2: $T(k) > \alpha$ and $T(k) > T(k-1)$

The condition shows that the learning procedure tends to be oscillating. Hence, to maintain the learning rate close to the maximum allowable value, we start decreasing it gradually by a factor θ, for example $\theta = 0.9$. In that case, we have: $\eta = \theta\eta$.

3) Case 3: $T(k) > \alpha$ and $T(k) \leq T(k-1)$

In this case, the learning rate keeps the same value. The above self-adaptive procedure can be performed at the conclusion of each epoch of training in both the encryption and decryption procedures. The critical value α can guarantee that the learning procedure will not settle at a stable point. At the same time, it helps maintain the learning rate close to the maximum allowable learning rate so that the learning trajectory is closely related to the different training data. More precisely, it will make the learning trajectory behave more randomly, which in turns make the analysis of the learning procedure more difficult without the knowledge of the initial state of the network.

3.4 Conclusion

In this chapter, the novel block cipher design based on RNNs was explained. The block cipher design has several advantages resulting from the introduction of RNNs for symmetric-key block cipher design. The block cipher design releases the limitation on the length of secret key. The block cipher can flexibly adjust the secret key and message length to accommodate different security and performance requirements. Moreover, it is capable of providing both high secure data encryption and data integrity services. Different cryptographic services are provided by an integrity scheme with a relatively simple architecture. Furthermore, the inherent parallel computing capability of the block cipher can accommodate high performance data encryption requirements such as secure point-to-point file transfer between gateways. In next chapter, the cipher is analyzed from other perspectives.

CHAPTER 4: Analysis of the Proposed Cipher

4.1 Introduction

The proposal to use the principles of NN has the potential to create a completely secure encryption mechanism, but it has some shortcomings and limitations. In order to achieve the long term goal of the creation of an unbreakable cryptosystem, these shortcomings and limitations have to be identified and overcome. In addition there is a need to do more analysis (such as, cryptanalysis, randomness etc) in order to have a reliable cipher. The summary of these weaknesses and the analysis which have been done, are explained next:

a) The proposed cipher is not a block cipher, it is a symmetric NN-based cipher. This leads us to the fact that there is no need to have two decomposition functions F_1 and F_2.

b) The scheme is based on two decomposition functions F_1 and F_2 that operate on a structure with only one neuron. From the description, it is not obvious the reason that such a simple architecture was chosen, or under what suggestions this scheme has been adopted. It should be clearly stated and justified that, this architecture is used for simplicity reasons or some specific properties of the two functions.

c) The scheme has been tested using different kinds of back propagation algorithms to see the differences in performance. The differences in the results were relative to speed measures and size of the network output.

d) The output of the network, the ciphertext, has been tested for randomness using three different statistical tests. The results show that although there is no correlation between ciphertext stream values, the ciphertext is not random.

e) The scheme showed to be resistant to differential and linear cryptanalysis and brute-force attack.

f) There is a possible chosen-plaintext attack, which will be explained in the cryptanalysis section.

The complete descriptions and justifications can be found in the following sections.

4.2 The Symmetric NN-Based Cipher

In this section, the first three issues explained above are described. First it is explained why the cipher is not a block cipher. Then the reason for keeping the network architecture so simple is mentioned and at the end, the results of applying other back propagation algorithms are described.

The cipher introduced in chapter three is called a block cipher, which is not accurate. A block cipher is a function which maps n-bit plaintext blocks to n-bit ciphertext blocks; n is called the block length [20]. Typically, a block size of 64 or 128 bits is used. Virtually all block ciphers are product ciphers, i.e. they combine at least two or more transformations in a manner intending that the resulting cipher is more secure than the individual components. The underlying idea is to build a complex encryption function by combining several simple operations which offer complementary, but individually insufficient security properties.

In order for the proposed cipher to be a block cipher, it should follow the above definition. The following experiment has been used as a counterexample to prove that the cipher is not a block cipher. In this experiment, the plaintext presented to the network consists of six strings (each string consists of 39 characters and each character is one byte) of "aaaaaaaaaaaaaaaaaaaaaaaaaaazzaaaaaaaaaa", which gives us a block of 234 bytes. After running the network simulation in Matlab, the network output, the ciphertext, was a block of 252 bytes, as it can be seen in Figure 4.1.

```
zsh_bbYYYzz0"1ƒƒ;~yrh_bbYYYyy|3Œ88...@yrh
_□□%%%¢¢-'1,,;~yrh_bbYYYyy0"1ƒƒp0zsh_bb
YYY┘└└┘┘└└└┘┘yrh_bbYYYyy1"1ƒƒ;~yrx}CC¶¶¶
vv5•2└┘*syrh_bbYYYyy0"1ƒƒ;~zsh_bbYYYzz0
"1ƒƒ;~yW,Sqq%%%¤¤/"1,,;~yrh_bbYYYyy1³'9
9...@yrh_bb□□□□□□□□□□□□zsh_bbYYYzz0"1ƒƒ;¢
yrh_bbYYYyy□□□□□□□
```

Figure 4.1 The output of the network

As it has been shown, plaintext is a 239-byte block and ciphertext is a 252-byte block, this does not follow the definition of block cipher and therefore the cipher is not a block cipher. Therefore from now on, the cipher is referred to as a symmetric cipher, rather than a symmetric block cipher.

The structure of the cipher design ensured that among the hidden layers of the neural network, there exists at least one that has only one neuron (denoted here as neuron ξ). This feature was used to decompose the feedforward operation of the NN into two functions F_1 and F_2. In such decomposition, F_1 is the feedforward operation over the weight and bias matrices performed from the input layer to neuron ξ, and F_2 is the same operation performed from neuron ξ to the output layer. The functions F_1 and F_2, were then used in the encryption process. The reason to use such a structure was to be close to other block ciphers as much as possible. Now that it is proved the cipher is not a block cipher, there is no need to have two separate function notations and just one F satisfies the system requirements, as shown in figure 4.2.

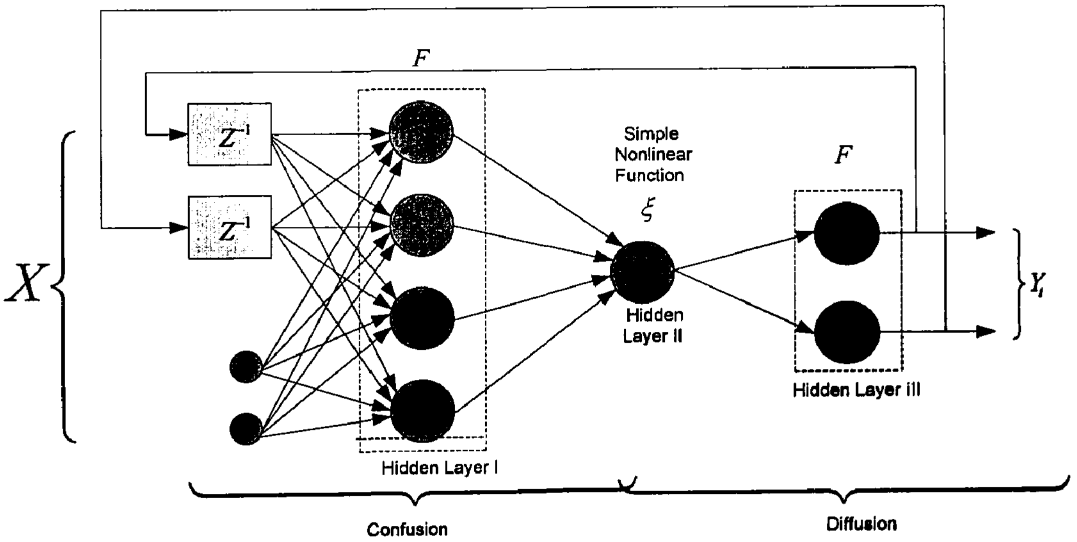

Figure 4.2 Network Architecture

The above fact does not have any impact on the architecture of the NN. Regarding the network architecture, as it is well-known, the problem of choosing the "optimal" network architecture for a given problem is very difficult and still remains an open problem [5]. But finding a small enough network is always desirable. The above simple architecture, with a hidden layer II having only one neuron, satisfies the confusion and diffusion properties of the cipher. The two basic techniques for obscuring the redundancies in a plaintext message are, according to Shannon [8], confusion and diffusion.

The relationship between the plaintext and the ciphertext is obscured by confusion [4],[8]. In order to apply confusion, redundancies and statistical patterns in ciphertext should be found. An easy way to achieve confusion is through substitution. The redundancy of the plaintext is dissipated by diffusion [4],[8], i.e. the redundancies are spread out over the ciphertext. In this way a cryptanalyst cannot easily find the redundancies. Transposition (permutation) is the simplest way to cause diffusion.

As it is shown in figure 4.2, from the input layer up to the hidden layer II, the layer with only one neuron ξ, we will have confusion, which is similar to the effect of substitution. Then by applying the simple non-linear function (sigmoid function in this case) to the inputs, we will achieve diffusion, which is similar to transposition. In addition, keeping the structure of the network simple can help with its analysis.

The original training method used for the Matlab simulation [17] was the Back Propagation with Variable Step size (BPVS) without momentum. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable. To make sure that it is the best algorithm, other algorithms such as [31]:

- Standard Back Propagation (BP) with and without momentum
- BPVS with momentum
- Resilient Back Propagation (RPROP)

were used to test the results. To get the result, Matlab simulation was run with the mentioned methods separately. One of the differences was relative to speed measures were RPROP managed to be the fastest, and the other difference was in term of the size of the network output. BPVS (without momentum) managed to have the shortest ciphertext. This result is consistent with the result of [5], i.e. the fact that the training method does not play a significant role in tackling the particular problem. On the other hand, a crucial role is being played by the network architecture.

4.3 Randomness of the Ciphertext

Originally, it has been assumed that the output of the network, the ciphertext, is random, but no test has been done to prove this assumption. In order to check randomness of the ciphertext three statistical tests, one correlation test and two run tests, were performed. The results show that although there is no correlation between ciphertext stream values (the fluctuations are random in nature), these values are not random. Using Matlab, first the output stream, shown in figure 4.1, was converted to ASCII, and the result is presented here:

```
x = [122 115 104 95 98 98 89 89 89  122 122 48 8220 49 402 402  59 126 121 114 104 95  98 98 89 89 89
121 121 124 51 338  56 56  8230 64 121 114 104 95 2 2 37 37 37 162 162 45 8216 49 8218 8218 59 126
121 114 104 95 98 98 89 89 89 121 121 48 8220 49 402 402 112 79 122 115 104 95 98 98 89 89 89 2 2 2 2
2 2 2 2 2 121 114 104 95 98 98 89 89 89 121 121 49 8220 49 402 402 59 126 121 114 120 125 67 67 182
182 182 118 118 53 8226 50 15 15 42 115 121 114 104 95 98 98 89 89 89 121 121 48 8220 49 402 402 59
126 122 115 104 95 98 98 89 89 89 122 122 48 8220 49 402 402 59 126 121 87 44 83 113 113 37 37 37
164 164 47 8220 49 8218 8218 59 126 121 114 104 95 98 98 89 89 89 121 121 49 179 8217 57 57 8230 64
121 114 104 95 98 98 2 2 2 2 2 2 2 2 2 2 2 2 122 115 104 95 98 98 89 89 89 122 122 48 8220 49 402 402
59 162 121 114 104 95 98 98 89 89 89 121 121 2 2 2 2 2 2 2];
```

Then the tests were done on the above sample data and they are explained in details in the following sections.

4.3.1 The Serial Correlation Test for Randomness of Fluctuations

The objective of this test is to examine the null hypothesis[1] that the fluctuations in a series are random in nature. It is assumed that the observations are obtained independently of each other and under similar conditions. The first serial correlation coefficient for a series of n terms, $x_i (i = 1,...,n)$, is defined [32] as:

$$r_1 = \frac{n}{n-1}\left\{\frac{\sum_{i=1}^{n-1}(x_i - \bar{x})(x_{i+1} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right\} \tag{4.1}$$

and this forms the test statistic. For $n \le 30$, critical values for r_1 can be found from the table available in [32]. For $n > 30$, the normal distribution provides a reasonable approximation. In both cases, if the test statistic exceeds the critical values, the null hypothesis is rejected.

In this case with $n = 252$, to find the critical values for the test statistics, normal distribution has been used. To plot the normal distribution, values for mean and standard deviation for x vector were calculated. Mean is equal to 615.68, variance is 3943232.51, which gives the standard deviation of 1985.75. The normal distribution for the critical values is shown in figure 4.3.

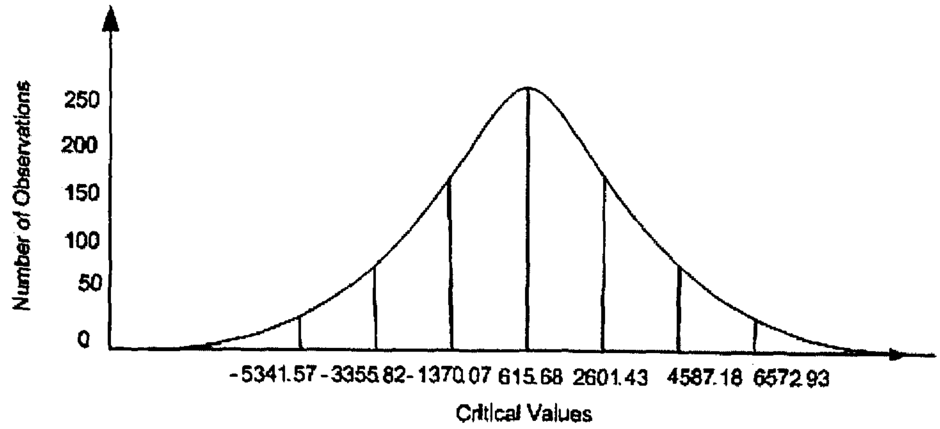

Figure 4.3 Normal Distribution for the critical values of r_1

[1] In statistics, a null hypothesis is a hypothesis that is presumed true until statistical evidence in the form of a hypothesis test indicates otherwise.

Using (4.1), the value of r_1 has been calculated and is equal to 0.0559. Based on the above distribution, level of significance $\alpha = 0.05$, is about 4587. Hence the null hypothesis is not rejected (because $r_1 = 0.0559 < 4587$); the correlation between successive observations is not significant.

4.3.2 The Run Test on Successive Difference

The objective of this test is to check the null hypothesis to see if the observations in a sample are independent of the order in the sequence. The observations in the sample must be obtained under similar conditions. From the sequence of observations, a sequence of successive differences is formed, i.e. each observation has the preceding one subtracted from it. The test statistics is provided by the number of runs of + and – signs, K, in the sequence of differences [32].

Let n be the initial sample size. For $5 \leq n \leq 40$, critical values of K can be obtained from the table available in [32]. For $n > 40$, K may be assumed to follow a normal distribution with mean $(2n-1)/3$ and variance $(16n-29)/90$. In both cases, when the test statistics lies in the critical region, the null hypothesis is rejected.

In this case, to find the critical values for the test statistics, normal distribution has been used. To plot the normal distribution, values for mean and standard deviation for x vector were calculated using the formula for mean and variance (mentioned above). Mean is equal 166.67, variance is 44.47, which gives the standard deviation of 6.66, and $K = 125$. The normal distribution for the critical values is shown in figure 4.4.

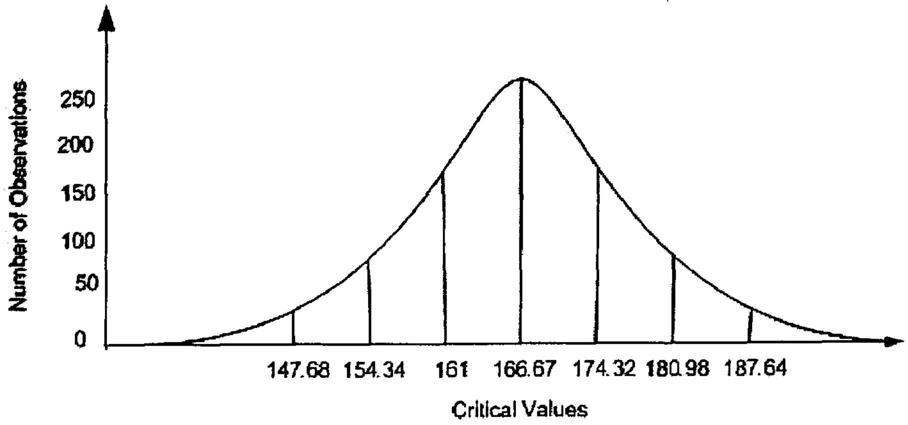

Figure 4.4 Normal Distribution for the critical value of K

Based on the above distribution, level of significance $\alpha = 0.05$, for $n = 252$, is 154 (the lower bound) and 181 (the upper bound). The test statistics lies in the critical region ($125 < 154$), therefore we have to reject the null hypothesis, i.e. observations in the sample are not random.

4.3.3 The Run Test

The objective of this test is to examine the significance of the order of the observations in a sample. It is necessary that the observations in the sample be obtained under similar conditions. All the observations in the sample larger than the median value are given a + sign and those below the median are given a – sign. If there is an odd number of observations then the median observation is ignored. This ensures that the number of + signs (n) is equal to the number of – signs. A sequence of values with the same sign is called a run and the number of runs, K, of the sample in the order of selection is found. K is the test statistic [32].

For $n > 30$, this test statistic can be compared with a normal distribution with mean $n+1$ and variance $\frac{1}{2}n(2n-2)/(2n-1)$. For $n < 30$, critical values for K are given in the table available in [32]. In both cases the null hypothesis that the observations in the sample occurred in a random order is rejected if the test statistics lies in the critical region.

In this case, to find the critical values for the test statistics, normal distribution has been used. To plot the normal distribution, values for mean and standard deviation for x vector were calculated using the formula for mean and variance (mentioned above). Mean is equal 253, variance is 125.74, which gives the standard deviation of 11.21, median is 98 and $K = 120$. The normal distribution for the critical values is shown in figure 4.5.

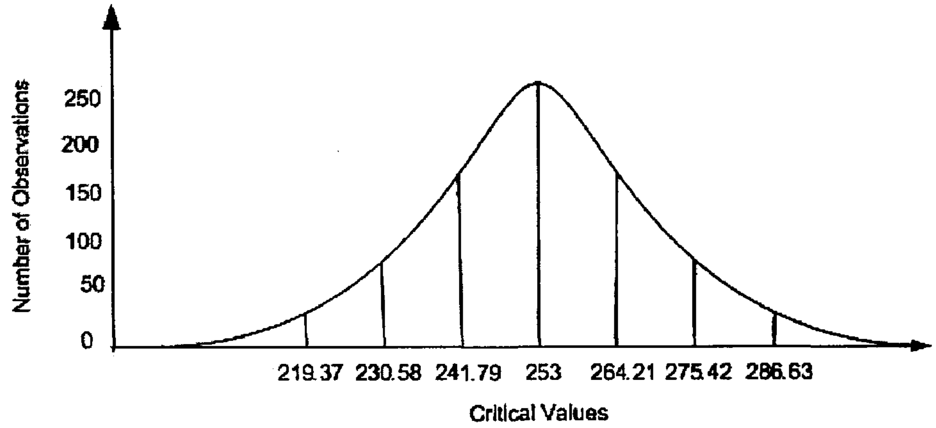

Figure 4.5 Normal Distribution for the critical value of K (run test)

Based on the above distribution, level of significance $\alpha = 0.05$, for $n = 252$, is 231 (the lower bound) and 275 (the upper bound). The test statistics lies in the critical region (120<231), therefore we have to reject the null hypothesis, i.e. observations in the sample are not random.

4.4 Cryptanalysis of the Cipher

Different cryptography algorithms offer different degrees of security. This security level depends on how hard they are to break. For an algorithm to be probably safe, it should follow these requirements [8]:

- The cost required to break it should be greater than the value of the encrypted data.
- The time required to break it should be longer than the time the encrypted data must remain secret.

- The amount of data encrypted with a single key should be less than the amount of data necessary to break the algorithm.

There is always a chance of new breakthrough in cryptanalysis that is why the term probably should be used because. On the other hand, the value of most data decreases over time. It is important that the value of the data always remain less than the cost to break the security protecting it. The objective of cryptography is to keep the plaintext (or the key, or both) secret from eavesdroppers. The communication channel is assumed to be insecure, i.e. eavesdroppers are assumed to have complete access to the communications between the sender and receiver.

A. Kerckhoffs in the nineteenth century introduced a fundamental assumption in cryptanalysis that the secrecy must reside entirely in the key. Kerckhoffs assumes that the cryptographic algorithm and its implementation are known to the cryptanalyst and he/she has the complete details. This is a good assumption to make, although real-world cryptanalysts don't always have such detailed information. Therefore, if cryptanalysis of an algorithm is not possible even with the knowledge of the algorithm, then without that knowledge, cryptanalysis is certainly impossible [8].

In chapter three, some of the possible cryptanalysis attacks were explained, and in this section three other fundamental attacks for symmetric ciphers (any newly proposed cipher must be analyzed against a regiment of known attack methods to be taken seriously), and a possible chosen-plaintext attack are described.

4.4.1 Resistance to Brute-force Attack

As with symmetric ciphers, the proposed algorithm is vulnerable to a brute-force attack. A brute-force attack is a method of defeating a cryptographic scheme by trying a large number of possibilities; for example, exhaustively working through all possible keys in order to decrypt a message. In most schemes, the theoretical possibility of a brute-force attack is recognized, but it is set up in such a way that it would be computationally infeasible to carry out. The counter measure is the same for all symmetric ciphers: using large keys. Because there is no constraint on the secret key length in this cipher, the key size can be increased to achieve required protection. As mentioned before by changing the length of secret key and the dimension of the hidden layers, user can flexibly adjust the security level.

4.4.2 Resistance to Differential Attacks

Differential cryptanalysis is a chosen-plaintext attack introduced by Eli Biham and Adi Shamir in 1990, and has been used to cryptanalyze Fiestel ciphers [33],[34]. It looks specifically at ciphertext pairs with known difference in, which lead to known difference out. The procedure begins with two plaintext messages m and $m^{'}$ with a known difference and they are traced through a probable pattern of differences after each round to find a probable difference for the ciphertext. For a 32-bit halves, there are two probable differences: $(\Delta m_{17} \parallel \Delta m_{16})$. Then m and $m^{'}$ are submitted for encryption to determine the actual difference and key is unknown. The result is compared to the probable difference. If there is a match,

$$E_K(m) \oplus E_K(m^{'}) = (\Delta m_{17} \parallel \Delta m_{16})$$

it is possible to suspect that all the probable patterns at all the intermediate rounds are correct. Considering this assumption, some deductions about the key bits can be made. To determine all the key bits, this procedure must be repeated many times [4]. In this cipher, plaintext is not divided in halves, so there is no probable difference. And there are no rounds therefore, there are no intermediate results, i.e. no assumptions can be made on the key bits.

4.4.3 Resistance to Linear Attacks

Linear cryptanalysis is a known-plaintext attack introduced by Matsui et al in 1993 [35]. It proved to be a very powerful attack and was able to cryptanalize DES faster than differential cryptanalysis. Here, a brief summery of the principle of linear cryptanalysis is described. For a cipher with n-bit plaintext and ciphertext blocks and m-bit key, let the plaintext block be labeled $P[1],...P[n]$, the ciphertext block $C[1],...C[n]$, and the key $K[1],...K[m]$. Then define

$$A[i, j,..., k] = A[i] \oplus A[j] \oplus ... \oplus A[k]$$

The objective of linear cryptanalysis is to find an effective linear equation of the form

$$P[\alpha_1, \alpha_2,..., \alpha_a] \oplus C[\beta_1, \beta_2,..., \beta_b] = K[\gamma_1, \gamma_2,..., \gamma_c] \tag{4.2}$$

where $x = 0 or 1$; $1 \leq a,b \leq n$, $1 \leq c \leq m$, and where the α, β, and γ terms represent fixed, unique bit locations. (4.2) holds with probability $p \neq 0.5$. If p is further from *0.5*, the equation is more effective. After determining a relation, the procedure is to compute the results of the left-hand side of (4.2) for a large number of plaintext-ciphertext pairs. If the result is 0 more than half the time, it should be assumed that $K[\gamma_1, \gamma_2,..., \gamma_c] = 0$. If it is 1 most of the time, assume $K[\gamma_1, \gamma_2,..., \gamma_c] = 1$. In this way a linear equation is constructed on the key bits. By getting more such relations, more key bits can be solved. The problem can be approached one round of the cipher at a time because of linear equations, and at the end, the results should be combined [4].

NNs are nonlinear dynamic machines that expand the expression of input data as a linear combiner of the input to the synapses and then perform a nonlinear transformation. This means having linear permutation followed by nonlinear transformation, which makes linear attack hard to achieve. In addition, as with differential attack, there is no rounds, therefore there is no intermediate results.

4.4.4 A Possible Chosen-plaintext Attack

As it was described in chapter three, plaintext messages have the form: $M_{i(i=1,...,n)} = \{M_1, M_2, M_3, ... M_n\}$, and ciphertext blocks have the form: $\{C_1, C_2,..., C_n\}$, where $C_1 = \{V_1, E_1\}$.

Now the following steps can be considered:

First Step

Enemy selects $\{M_1\},\{M_2\},...\{M_n\}$, that is a sequence of plaintext messages. **Each** message contains exactly one block.

Then for $j=1,...k$,

$$X_j=(M_0\|M_j)$$

$$V_j=F(X_j)$$

$$Y_j=F(V_j)$$

$$E_j=M_j-Y_j$$

$$C_j=\{V_j,E_j\}$$

Then enemy computes: $Y_j=M_j-E_j$. She has a number of equations

$$Y_j=F(V_j) \qquad j=1,...k$$

Also she knows Y_j,V_j, so she can organize a NN which approximates F (an equivalent NN).

Second Step

Consider the following sequence of plaintext messages: $\{M_1,M_{2,1}\},\{M_1,M_{2,2}\},...\{M_1,M_{2,k}\}$, each message contains exactly two blocks and the first block is fixed as M_1. The corresponding sequence of ciphertext message is given by $\{C_1,C_{2,j}\} \qquad j=1,...,k$.

After the computation of C_1, the NN will be trained for one epoch. After the encryption of the first block (M_1 to C_1), the function F will change due to one epoch training. The new function will be $\overline{F}$. Then

$$X_{2,j} = \left(Y_1 \| M_{2,j}\right)$$
$$V_{2,j} = \overline{F}\left(X_{2,j}\right)$$
$$Y_{2,j} = \overline{F}\left(V_{2,j}\right)$$
$$E_{2,j} = M_{2,j} - Y_{2,j}$$

Enemy computes $Y_{2,j}$ from $M_{2,j}$ and $C_{2,j}$. In a similar way like first step she can approximate $\overline{F}$ with NN's. If all the plaintext messages have M_1 as the first block (i.e. starting from M_1), she will be able to use the approximated function to encrypt the message.

4.5 Conclusion

In this chapter, the NN-based block cipher proposed in [17] has been analyzed. The result of this analysis is as following:

- It is proved that the cipher is a symmetric cipher, not a block cipher.
- The simple architecture of the network is compatible with the requirement for confusion, and diffusion properties of a cryptosystem. In addition, the simpler the network, the easier is to analyze it.
- It is shown that back propagation with variable stepsize algorithm is the best algorithm to use.
- Although there is no correlation between the values of the output stream, the ciphertext, it is proved that the ciphertext is not random.
- It is shown that the cipher is resistant to brute-force, differential and linear attacks.
- A possible chosen-plaintext attack is presented.

CHAPTER 5: Conclusion

Conclusion

Neural Networks (NNs) are a very powerful tool in many scientific disciplines, due to their function approximation and generalization capabilities. Also, since NNs are parallel and distributed processing devices they can be implemented in parallel hardware and, consequently, they can be used for real-time applications. In this thesis, a study of NN approach has been attempted to encounter new cryptographic cipher designs. Based on this work and work of others (described in chapter two) it is possible to train feedforward NNs to perform encryption/decryption. This work is among the first few attempts towards this direction, and it is strongly believed that there are numerous issues remaining in order to obtain a comprehensive view of the ability of NNs to simulate data encryption and decryption.

It has been shown that the NN-based symmetric cipher [17] has its own strengths and weaknesses. Its strengths can be described as:

- There is no limitation on key size and by increasing the key size one can increase the security and performance levels.
- It is capable to provide both security and data integrity.
- It has simple and easy to implement architecture, yet strong enough to provide confusion and diffusion.

The weaknesses are:

- Ciphertext is not random.
- Compared with other ciphers, it is not fast enough.
- It may be unreliable (due to the nature of back propagation algorithm).
- It involves many floating point calculations.

In order to achieve the main goal of any cryptography method, i.e. being unbreakable, the weaknesses of the symmetric cipher should be overcome. Therefore, further research work can be done by adopting other algorithms (such as, supervised Hebb Rule) instead of back-propagation learning algorithm. Because of the nature of this algorithm, the reliability of the cipher is not as strong as one would like it to be. This could be due to the fact that the speed of the change of the gradient of the learning trajectory is sometimes much faster than the change of the error. So, when the precision limit is met before the threshold, the learning procedure could be broken, and therefore could stop suddenly.

In general, it is not possible to control or predict the gradient for the algorithm, and the state of the system is difficult to determine when the cipher stops suddenly. By adopting supervised Hebb Rule instead of back-propagation learning algorithm, the abovementioned problem can be addressed. In addition, the application of Hebb Rule for data encryption can also solve the learning rate selection difficulty of the back-propagation thus can open a path to investigate in depth how to involve finite field operation in the framework, in order to avoid the floating point calculations in future. Thus the advantages of applying Hebb Rule are:

- It is faster.
- No gradient calculation is needed
- It is possible to operate based on binary operations

For any new cipher to be compatible with available ciphers, the ciphertext must be random. Thus, to have randomness in the ciphertext, a random number generator can be added to the network. In this case, the seed of random numbers should be added to the secret key *S*, described in chapter 3 (section 3.2.1).

The effect of other cryptanalytical attacks also should be studied and the cipher should be exposed to as many cryptanalysis as possible.

It is not encouraged that the cipher presented here, be used in any critical or sensitive data or application. The strength of any new algorithm can only become apparent after intense public scrutiny. In conclusion it is believed that the NN-based cryptography is promising although many problems have to be resolved and many further work needs to be done.

References

[1] J. Vandewalle, B. Preneel, and M. Csapodi, "Data Security Issues, Cryptographic Protection Methods, and the Use of Cellular Neural Networks and Cellular Automata", 1998 Fifth IEEE International Workshop on Cellular Neural Networks and their Applications, London, England, pp. 14-17 April 1998.

[2] http://security.uwo.ca/history.html. Information Security, Why Do We Need IT?

[3] G. Coulouris, J. Dollimore, and T. Kindberg. Distributed Systems: Concepts and Design. Addison Wesley, 3rd edition, pp. 251-300, 2001.

[4] W. Stallings. Cryptography and Network Security: Principles and Practices. Prentice Hall, 3rd edition, 2003.

[5] G.C. Meletiou, D.K. Tasoulis, and M.N. Vrahatis, "A First Study of the Neural Network Approach in the RSA Cryptosystem", 7th IASTED International Conference Artificial Intelligence and Soft Computing, 2002.

[6] S. Su, A. Lin, and J. Yen, "Design and realization of a new chaotic neural encryption/decryption network", Proc. IEEE Asia-Pacific Conf. on Circuits and Systems, pp. 335-338, 2000.

[7] L. Yee, and C. De Silva, "Application of multilayer perceptron networks in symmetric block ciphers", Proc. 2002 Int'l Joint Conf. on Neural Nets, vol. 2, pp. 1455-1458, 2000.

[8] B. Schneier. Applied Cryptography. John Wiley & Sons, Inc. 2nd edition, 1996.

[9] L. Yee, and C. De Silva, "Application of Multilayer Perceptron Networks in Public Key Cryptography", Proc. 2002 Int'l Joint Conf. on Neural Nets, vol. 2, pp.1439-1443, 2000.

[10] L. Yee, and C. De Silva, "Application of Multilayer Perceptron Networks as a One-Way Hash Function", Proc. 2002 Int'l Joint Conf. on Neural Nets, vol. 2, pp. 1459-1462, 2000.

[11] W. Kinzel, and I. Kanter, "Neural Cryptography", Proc. of the 9th Int'l Conf. on Neural Information Processing (ICONIP'02), vol. 3, pp. 1351-1354, 2002.

[12] A. Klimov, A. Mityaguine, and A. Shamir, "Analysis of Neural Cryptography", Proc. AsiaCrypt 2002, pp. 288-298. Springer Verlag, 2002.

[13] D. A. Karras, and V. Zorkadis, "On neural network techniques in the secure management of communication systems through improving and quality assessing pseudorandom stream generators", Neural Networks, vol.16, issues 5-6, pp. 899-905, June-July 2003.

[14] F.E. Brickell, and M.A. Odlyzko, "Cryptanalysis: A Survey of Recent Results", Proceedings of the IEEE, vol. 76, no. 5, May 1988.

[15] S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication", ACM Computing Surveys, vol. 35, issue 3, pp. 309-329, Sept. 2003.

[16] S. Haykin. Neural Networks, a comprehensive foundation. Macmillan College Publishing Company, 1994.

[17] S. Wu. A Block Cipher Design Using Recurrent Neural Networks. Master of Engineering Dissertation, Ryerson University, 2003.

[18] J. Benitez, and J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?", IEEE Trans. on Neural Networks, vol. 8, no. 5, pp.1156-1164, Sept. 1997.

[19] D. Denning. Cryptography and Data Security. Addison-Wesley, Inc., pp. 1-20, 1982.

[20] A. Menezes, P. Oorschot, and S. Vanstone. Handbook of Applied Cryptography. CRS Press, 1997.

[21] G.C. Meletiou, D.K. Tasoulis, and M.N. Vrahatis, "Cryptography through Interpolation and Computational Intelligence Methods", Bulletin of the Greek Mathematical Society, 2003.

[22] K. Hornik, "Multilayer Feedforward Networks are Universal Approximators", Neural Networks, 2, 1989, 359-366.

[23] H. White, "Connectionist Nonparametric Regression: Multilayer Feedforwrd Networks can learn arbitrary mappings", Neural Networks, vol. 2, pp. 359-366, 1989.

[24] I. Kanter, and W. Kinzel, "Theory of Interacting Neural Networks", Phys. Rev E 62, 2555 (2000).

[25] M. Rosen, I. Kanter, and W. Kinzel, "Cryptography based on neural networks: analytical results", cond-mat/0202350, 2002.

[26] S. Scott, L. Alvin, and Y. Jui-Cheng, "Design and Realization of a New Chaotic Neural Encryption/Decryption Network", Proc. 2000 IEEE Asia-Pacific Conference on Circuits and Systems, pp. 335-338, 2000.

[27] D. Pointcheval, "Neural Networks and their Cryptographic Applications", Livre des resumes Eurocode '94, Pascale Charpin Ed. INRIA, 1994.

[28] S. Townley, A. Iichmann, M. G. Weib, W. Mcclements, A. C. Ruiz, D. H. Owens, and D. Pratzel-Wolters, "Existence and Learning of Oscillations in Recurrent Neural Networks", IEEE Trans on Neural Networks, vol. 11, no. 1, pp. 205-214, Jan. 2000.

[29] L. Almeida, "A learning rule for asynchronous perceptrons with feedback in a combinatorial environment", Proc. 1st IEEE International Conference on Neural Networks, vol. 2, pp. 105-110, 1987.

[30] U. Hengartner, J. Bolliger, and T. Gross, "TCP Vegas revisited" *Proc. INFOCOM*, 19th Annual Joint Conference of the IEEE Computer and Communications Societies, vol.3, pp. 1546 –1555, March 2000.

[31] H. Demuth and M. Beale. Neural Network Toolbox. Math Works Inc. v. 4, 2000.

[32] G. K. Kanji. 100 Statistical Tests. SAGE Publications Inc. 1999.

[33] E. Biham and A. Shamir, "Differential Cryptanalysis of the Data Encryption Standard", Advances in Cryptology – Crypto '92 Proceedings, Springer-Verlag, 1993.

[34] E. Biham and A. Shamir, "Differential Cryptanalysis of the full 16-round DES", Advances in Cryptology – Crypto '92 Proceedings, Springer-Verlag, 1993.

[35] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", Proceedings of Eurocrypt '93, Springer-Verlag, Berlin, pp 386-397, 1993.

⑧ BC-74.211