A Novel Text Encryption Algorithm

Elena Acevedo, Ángel Martínez, Marco Acevedo, Fabiola Martínez

Escuela Superior de Ingeniería Mecánica y Eléctrica,
Instituto Politécnico Nacional, Mexico City, Mexico

{eacevedo, macevedo, fmartinezzu}@ipn.mx

Abstract. The encryption/decryption processes is applied to text. Both algorithms involved the use of the associative models, in particular we propose the Alpha-Beta associative memories. The original text is divide in sets of 9 elements, and together with a secret key they build an associative memory which represents the encrypted text. The advantage of our proposal is the cyphertext does not have the same dimension that the plaintext. Additionally, the cyphertext is represented as an image, therefore, since the beginning the encrypted text has a different meaning.

Keywords: Artificial Intelligence, associative models, alpha-beta associative memory, text encryption.

1 Introduction

Cryptrography [1] is the science of protecting data and communications. A cryptosystem has two parts: encryption, which is done at the sender's end of the message and means to put the actual plaintext (original messages) into cyphertext (secret code), and decryption, which is done at the recipient's end and means to translate the cyphertext back into the original plaintext message. Generally an encryption or decryption algorithm will relay on a secret key [2], which may be a number with particular properties, or a sequence of bits; the algorithm itself may be well known, but to apply the decryption to a given cyphertext requires knowledge of the particular key used.

Traditional encryption algorithms are private key encryption standards (DES and AES), public key standards such as Rivest Shamir Adleman (RSA), and the family of elliptic-curve-based encryption (ECC), as well as the international data encryption algorithm (IDEA).

There are other algorithms for encrypting text. Some of them use the corresponding decimal ASCII code, convert it to binary numbers and apply a process [3], [4], [5] for changing the order of the bits. Another proposal is a technique on matrix scrambling which is based on random function [6], shifting and reversing techniques of circular queue. A symmetrical encryption algorithm [7] is proposed in this paper to prevent the outside attacks to obtain any information from any data-exchange in Wireless Local Area Network. Other algorithms [8] encipher message into nonlinear equations using public key and decipher by the intended party using private key. Some works applied modifications to traditional encryption algorithms:

AES [9], blowfish [10], DES [11] and RSA [12]. In this work we propose a text-image encryption as [13] but with a different approach.

2 Basic Concepts

Sometimes, we recall a person, a place or a feeling when we smell certain perfume, we see a blue sky or we watch a movie, in other words, we can associate a person with a perfume, a place with a blue sky or a feeling with a movie. We also can resolve a quadratic equation because we at once associate the form of this equation with its solution as follows,

$$ax^2 + bx + c = 0 \rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Therefore, our brain learns by association. Neural Networks try to simulate the structure of the brain by interconnecting a set of neurons, on the other side, Associative Memories try to simulate the behavior of the brain, i.e., they associate concepts.

Associative Memories (AM) associate patterns x with y, which can represent any concept: faces, fingerprints, DNA sequences, animals, books, preferences, diseases, etc. We can extract particular features of these concepts to form patterns x and y.
There are two phases for designing an associative memory: Training and Recalling.
In the Training Phase (see Figure 1), the process of associate patterns x with patterns y is performed. Now, we say that the memory is built.

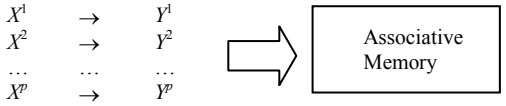

Fig. 1. Training Phase of an Associative Memory

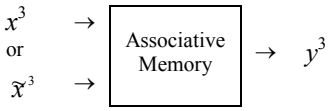

Fig. 2. Recalling Phase of an Associative Memory

Figure 2 shows the Recalling phase where the pattern x is presented to the AM and the corresponding pattern y is recalled. Also, a noisy version of a pattern x could be presented and the memory should recalls is corresponding y pattern.

The input and output patterns are represented by vectors. The task of association of these vectors is called Training Phase and the Recognizing Phase allows recovering patterns. The stimuli are the input patterns represented by the set $\mathbf{x} = \{x^1, x^2, x^3, \ldots, x^p\}$ where p is the number of associated patterns. The responses are the output patterns and are represented by $\mathbf{y} = \{y^1, y^2, y^3, \ldots, y^p\}$. Representation of vectors x^μ is

$x^{\mu} = \{x_1^{\mu}, x_2^{\mu}, \ldots, x_n^{\mu}\}$ where n is the cardinality of x^{μ}. The cardinality of vectors y^{μ} is m, then $y^{\mu} = \{y_1^{\mu}, y_2^{\mu}, \ldots, y_m^{\mu}\}$. The set of associations of input and output patterns is called the fundamental set or training set and is represented as follows: $\{(\mathbf{x}^{\mu}, \mathbf{y}^{\mu}) \mid \mu = 1, 2, \ldots, p\}$

There are two types of associative memories concerning to the nature of the input and output patterns.

A memory is ***Autoassociative*** if it holds that $x^{\mu} = y^{\mu} \ \forall \mu \in \{1, 2, \ldots, p\}$, then one of the requisites is that $n = m$.

A memory is ***Heteroassociative*** when $\exists \mu \in \{1, 2, \ldots, p\}$ for which $x^{\mu} \neq y^{\mu}$. Notice that there can be heteroassociative memories with $n = m$.

Now, we will describe the Morphological associative model.

The fundamental difference between classic associative memories (Lernmatrix [14], Correlograph [15], Linear Associator [16] and Hopfield [17]) and Morphological associative memories [18] lies in the operational bases of the latter, which are the morphological operations: dilation and erosion. This model broke out of the traditional mold of classic memories which use conventional operations for vectors and matrices in learning phase and sum of multiplications for recovering patterns. Morphological associative memories change products to sums and sums to maximum or minimum in both phases.

The basic computations occurring in the proposed morphological network are based on the algebraic lattice structure $(R, \vee, \wedge, +)$, where the symbols $\vee$ and $\wedge$ denote the binary operations of maximum and minimum, respectively. Using the lattice structure $(R, \vee, \wedge, +)$, for an m x n matrix A and a p x n matrix B with entries from R, the matrix product $C = A \nabla B$, also called the max product of A and B, is defined by equation (1).

$$c_{ij} = \bigvee_{k=1}^{p} a_{ik} + b_{kj} = \left(a_{i1} + b_{1j}\right) \vee \cdots \vee \left(a_{ip} + b_{pj}\right) \tag{1}$$

The *min product* of A and B induced by the lattice structure is defined in a similar fashion. Specifically, the *i,j*th entry of $C = A \Delta B$ is given by equation (2).

$$c_{ij} = \bigwedge_{k=1}^{p} a_{ik} + b_{kj} = \left(a_{i1} + b_{1j}\right) \wedge \cdots \wedge \left(a_{ip} + b_{pj}\right) \tag{2}$$

Suppose we are given a vector pair $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t \in \mathbf{R}^m$. An associative morphological memory that will recall the vector when presented the vector is showed in equation (3)

$$W = y \nabla (-x)^t = \begin{bmatrix} y_1 - x_1 & \cdots & y_1 - x_n \\ \vdots & \ddots & \vdots \\ y_m - x_1 & \cdots & y_m - x_n \end{bmatrix} \tag{3}$$

Since W satisfies the equation $W \Delta \mathbf{x} = \mathbf{y}$ as can be verified by the simple computation in equation (4)

$$W \nabla x = \begin{bmatrix} \bigvee_{i=1}^{n} (y_1 - x_i + x_i) \\ \vdots \\ \bigvee_{i=1}^{n} (y_m - x_i + x_i) \end{bmatrix} = y \quad (4)$$

Henceforth, let $(\mathbf{x}^1, \mathbf{y}^1)$, $(\mathbf{x}^2, \mathbf{y}^2)$, …, $(\mathbf{x}^p, \mathbf{y}^p)$ be p vector pairs with $x^k = \left(x_1^k, x_2^k, \ldots, x_n^k\right)^t \in \mathbf{R}^n$ and $y^k = \left(y_1^k, y_2^k, \ldots, y_m^k\right)^t \in \mathbf{R}^m$ for k = 1, 2, …, p. For a given set of pattern associations $\{(\mathbf{x}^k, \mathbf{y}^k) \mid k = 1, 2, \ldots, p\}$ we define a pair of associated pattern matrices (X, Y), where $X=(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p)$ and $Y=(\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^p)$. Thus, X is of dimension n x p with i,jth entry x_i^j and Y is of dimension m x p with i,jth entry y_i^j. Since $\mathbf{y}^k \nabla (-\mathbf{x}^k)^t = \mathbf{y}^k \Delta (-\mathbf{x}^k)^t$, the notational burden is reduced by denoting these identical morphological outer vector products by $\mathbf{y}^k \times (-\mathbf{x}^k)^t$. With each pair of matrices (X, Y) we associate two natural morphological m x n memories M and W defined by

$$M = \bigvee_{k=1}^{p} \left(y^k \otimes \left(-x^k\right)^t\right) \quad (5)$$

$$W = \bigwedge_{k=1}^{p} \left(y^k \otimes \left(-x^k\right)^t\right) \quad (6)$$

With these definitions, we present the algorithms for the training and recalling phase.

Training Phase

1. For each p association $(\mathbf{x}^\mu, \mathbf{y}^\mu)$, the minimum product is used to build the matrix $\mathbf{y}^\mu \Delta (-\mathbf{x}^\mu)^t$ of dimensions m x n, where the input transposed negative pattern $\mathbf{x}^\mu$ is defined as $(-\mathbf{x}^\mu)^t = \left(-x_1^\mu, -x_2^\mu, \ldots, -x_n^\mu\right)$.
2. The maximum and minimum operators ($\vee$ and $\wedge$) are applied to the p matrices to obtain M and W memories as equations (5) and (6) show.

Recognizing phase

In this phase, the minimum and maximum product, Δ and ∇, are applied between memories M or W and input pattern $\mathbf{x}^\omega$, where $\omega \in \{1, 2, ..., p\}$, to obtain the column vector $\mathbf{y}$ of dimension m as equations (7) and (8) shows:

$$\mathbf{y} = M \Delta \mathbf{x}^\omega \quad (7)$$

$$\mathbf{y} = W \nabla \mathbf{x}^\omega \quad (8)$$

Now, we present an illustrative example for learning and recognizing phases of a Morphological associative memory.

Suppose we want to associate a set of three pairs of patterns, then $p = 3$. The cardinality of **x** and **y** will be $n = 3$ and $m = 4$, respectively. The three pairs of patterns are:

$$x^1 = \begin{pmatrix} -255 \\ 0 \\ 0 \end{pmatrix} \rightarrow y^1 = \begin{pmatrix} 255 \\ 255 \\ 255 \\ 255 \end{pmatrix}, x^2 = \begin{pmatrix} 0 \\ -255 \\ 0 \end{pmatrix} \rightarrow y^2 = \begin{pmatrix} 155 \\ 128 \\ 0 \\ 0 \end{pmatrix}$$

$$x^3 = \begin{pmatrix} 0 \\ 0 \\ -255 \end{pmatrix} \rightarrow y^3 = \begin{pmatrix} 255 \\ 255 \\ 255 \\ 0 \end{pmatrix}$$

Now, we apply the first step of training phase for associating the pair number one

$$y^1 \times (-x^1)^t = \begin{bmatrix} 255 \\ 255 \\ 255 \\ 255 \end{bmatrix} \times -[-255\ 0\ 0] = \begin{bmatrix} 255+255 & 255-0 & 255-0 \\ 255+255 & 255-0 & 255-0 \\ 255+255 & 255-0 & 255-0 \\ 255+255 & 255-0 & 255-0 \end{bmatrix} =$$

$$= \begin{bmatrix} 510 & 255 & 255 \\ 510 & 255 & 255 \\ 510 & 255 & 255 \\ 510 & 255 & 255 \end{bmatrix}$$

The same process is performed at remain pairs of patterns, and then the maximum of each element of every matrix is obtained as follows:

$$M = \begin{bmatrix} 510 & 255 & 255 \\ 510 & 255 & 255 \\ 510 & 255 & 255 \\ 510 & 255 & 255 \end{bmatrix} \bigvee \begin{bmatrix} 155 & 410 & 155 \\ 128 & 383 & 128 \\ 0 & 255 & 0 \\ 0 & 255 & 0 \end{bmatrix} \bigvee$$

$$\bigvee \begin{bmatrix} 255 & 255 & 510 \\ 255 & 255 & 510 \\ 255 & 255 & 510 \\ 0 & 0 & 255 \end{bmatrix} = \begin{bmatrix} 510 & 410 & 510 \\ 510 & 383 & 510 \\ 510 & 255 & 510 \\ 510 & 255 & 255 \end{bmatrix}$$

Now, we present the first input vector to the *max*-type morphological associative memory

$$M \Delta x^1 = \begin{bmatrix} 510 & 410 & 510 \\ 510 & 383 & 510 \\ 510 & 255 & 510 \\ 510 & 255 & 255 \end{bmatrix} \Delta \begin{bmatrix} -255 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 510+(-255) \wedge 410+0 \wedge 510+0 \\ 510+(-255) \wedge 383+0 \wedge 510+0 \\ 510+(-255) \wedge 255+0 \wedge 510+0 \\ 510+(-255) \wedge 255+0 \wedge 255+0 \end{bmatrix}$$

$$M \Delta x^1 = \begin{bmatrix} 255 \\ 255 \\ 255 \\ 255 \end{bmatrix} = y^1$$

When we present the other two input patterns (x^2 and x^3) to the memory, we recall their corresponding output patterns (y^2 and y^3).

The process of this illustrative example will be used in the following section for explaining our proposal.

3 Proposed Model

3.1 Encryption Algorithm

In order to describe both algorithms: encryption and decryption, we will present an illustrative example.

Suppose the message:

Help me! I am in danger, please

In the first step, groups of 9 characters are formed as follows,

Help me!
I am in d
anger, pl
ease____

If the number of characters is not multiple of 9, then we have to add spaces.

Then, each character of the message is converted to the ASCII code, and we have:

$$y^1 = [72\ 101\ 108\ 112\ 32\ 109\ 101\ 33\ 32]$$

$$y^2 = [73\ 32\ 97\ 109\ 32\ 105\ 110\ 32\ 100]$$

$$y^3 = [97\ 110\ 103\ 101\ 114\ 44\ 32\ 112\ 108]$$

$$y^4 = [101\ 97\ 115\ 101\ 32\ 32\ 32\ 32\ 32]$$

These vectors of dimension 9 represent the output patterns. Now, we have to build the input patters, which are the private key. As we have four vectors, the four input vectors have a dimension of 4, and they are built as follows,

$$x^1 = [-300\ 0\ 0\ 0]$$

$$x^2 = [0\ -300\ 0\ 0]$$

$$x^3 = [0\ 0\ -300\ 0]$$

$$x^4 = [0\ 0\ 0\ -300]$$

With these pairs of patterns we apply the training phase for building *max* morphological associative memory. The first pair is associated,

$$y^1 \times (-x^1)^t = \begin{bmatrix} 72 \\ 101 \\ 108 \\ 112 \\ 32 \\ 109 \\ 101 \\ 33 \\ 32 \end{bmatrix} \times -[-300\ 0\ 0\ 0] = \begin{bmatrix} 72+300 & 72-0 & 72-0 & 72-0 \\ 101+300 & 101-0 & 101-0 & 101-0 \\ 108+300 & 108-0 & 108-0 & 108-0 \\ 112+300 & 112-0 & 112-0 & 112-0 \\ 32+300 & 32-0 & 32-0 & 32-0 \\ 109+300 & 109-0 & 109-0 & 109-0 \\ 101+300 & 101-0 & 101-0 & 101-0 \\ 33+300 & 33-0 & 33-0 & 33-0 \\ 32+300 & 32-0 & 32-0 & 32-0 \end{bmatrix} =$$

$$= \begin{bmatrix} 372 & 72 & 72 & 72 \\ 401 & 101 & 101 & 101 \\ 408 & 108 & 108 & 108 \\ 412 & 112 & 112 & 112 \\ 332 & 32 & 32 & 32 \\ 409 & 109 & 109 & 109 \\ 401 & 101 & 101 & 101 \\ 333 & 33 & 33 & 33 \\ 332 & 32 & 32 & 32 \end{bmatrix}$$

Remain pairs are associated as in the previous process. The built *max* associative memory is showed in Figure 3.

$$M = \begin{bmatrix} 372 & 373 & 397 & 401 \\ 401 & 332 & 410 & 397 \\ 408 & 397 & 403 & 415 \\ 412 & 409 & 401 & 401 \\ 332 & 332 & 414 & 332 \\ 409 & 405 & 344 & 332 \\ 401 & 410 & 332 & 332 \\ 333 & 332 & 412 & 332 \\ 332 & 400 & 408 & 332 \end{bmatrix}$$

Fig. 3. Associative Memory representing the encrypted text

This memory represents the encryption of the message. We obtain the greatest number from the matrix, in this case 414, and we calculate 415 – 255 = 160. We subtract 160 from all the elements in the matrix then we have the results showed by the Figure 4.

$$M = \begin{bmatrix} 212 & 213 & 237 & 241 \\ 241 & 172 & 250 & 237 \\ 248 & 237 & 243 & 255 \\ 252 & 249 & 241 & 241 \\ 172 & 172 & 254 & 172 \\ 249 & 245 & 184 & 172 \\ 249 & 245 & 184 & 172 \\ 173 & 172 & 252 & 172 \\ 172 & 240 & 248 & 172 \end{bmatrix}$$

Fig. 4. Resulting matrix when we subtract 160 from all the elements of the matrix in Figure 3.

The idea of the subtraction is to show the memory as an image, as Figure 3 shows.

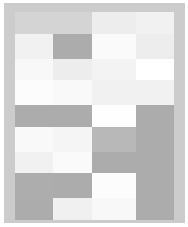

Fig. 5. Image representing the memory

At the first sight, we can imagine that the image is an encryption from another image.

If we perform a cryptanalysis, we will do it to obtain the original image without thinking that is a text.

This is the main advantage of our proposal.

Now, we will describe the decryption algorithm.

3.2 Decryption Algorithm

We recover the original text from the last matrix.

First, we add 160 to all the elements to the matrix in Figure 4. The result will be the matrix in Figure 3.

Now, we generate the input vectors or the private key.

These patterns are presented to the morphological associative memory, as follows,

$$M\Delta x^1 = \begin{bmatrix} 372 & 373 & 397 & 401 \\ 401 & 332 & 410 & 397 \\ 408 & 397 & 403 & 415 \\ 412 & 409 & 401 & 401 \\ 332 & 332 & 414 & 332 \\ 409 & 405 & 344 & 332 \\ 401 & 410 & 332 & 332 \\ 333 & 332 & 412 & 332 \\ 332 & 400 & 408 & 332 \end{bmatrix} \Delta \begin{bmatrix} -300 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 372+(-300) \wedge 373+0 \wedge 397+0 & 401+0 \\ 401+(-300) \wedge 332+0 \wedge 410+0 & 397+0 \\ 408+(-300) \wedge 397+0 \wedge 403+0 & 415+0 \\ 412+(-300) \wedge 409+0 \wedge 401+0 & 401+0 \\ 332+(-300) \wedge 332+0 \wedge 414+0 & 332+0 \\ 409+(-300) \wedge 405+0 \wedge 344+0 & 332+0 \\ 401+(-300) \wedge 410+0 \wedge 332+0 & 332+0 \\ 333+(-300) \wedge 332+0 \wedge 412+0 & 332+0 \\ 332+(-300) \wedge 400+0 \wedge 408+0 & 332+0 \end{bmatrix}$$

$$M\Delta x^1 = \begin{bmatrix} 72 \\ 101 \\ 108 \\ 112 \\ 32 \\ 109 \\ 101 \\ 33 \\ 32 \end{bmatrix} = y^1$$

We perform the same process with the rest of the patterns, and we obtain the original output vectors.

$$y^1 = [72\ 101\ 108\ 112\ 32\ 109\ 101\ 33\ 32]$$
$$y^2 = [73\ 32\ 97\ 109\ 32\ 105\ 110\ 32\ 100]$$
$$y^3 = [97\ 110\ 103\ 101\ 114\ 44\ 32\ 112\ 108]$$
$$y^4 = [101\ 97\ 115\ 101\ 32\ 32\ 32\ 32\ 32]$$

Each element of the patterns are converted to its corresponding ASCII code, then all the patterns are concatenated and, finally we have the original text.

4 Experiments and Results

We perform 1000 tests with different messages, length of the messages and values in the private key.

The Table 1 shows three examples of messages with different values of private key and lengths.

Table 1. Examples of messages with different lengths

Original message	Value of private key	Recovered message
Help me! I'm in danger, please (30 characters)	-10	felp[mf[cI_m?in?ddngerb[pleaseWi[Wi[
	-20	\elpQmeQYIUm5in5dangerXQpleaseM_QM_Q
	-50	Help3me3;I7m in danger:3please/A3/A3
	-80	Help me! I'm in danger, please # #
	-82	Help me! I'm in danger, please ! !
	-83	Help me! I'm in danger, please
	-84	Help me! I'm in danger, please
	-90	Help me! I'm in danger, please
	-100	Help me! I'm in danger, please
	-300	Help me! I'm in danger, please
	-500	Help me! I'm in danger, please
	-1000	Help me! I'm in danger, please
Then there was the bad weather. It would come in one day when the fall was over. We would have to shut the windows in the night against the rain and the cold wind would strip the leaves from the trees in the Place Contrescarpe. (227 characters)	-10	*Not recovered*
	-20	*Not recovered*
	-50	*Not recovered*
	-80	*Not recovered*
	-87	*Not recovered*
	-88	*Recovered*
	-89	*Recovered*
	-90	*Recovered*
	-100	*Recovered*
	-300	*Recovered*
	-500	*Recovered*
	-1000	*Recovered*
4090 characters, a sheet approximately	-255	*Recovered*

From Table 1, we can observe that the value of the private key must be higher when the number of characters is increased to assure the original message be recovered. We can see that for some values of private key the recovering fails.

5 Conclusions

Associative models have been applied in classification, prediction, pattern recognition and feature selection. In this paper we demonstrated that this approach can also be applied in text encryption.

The algorithm is symmetric and it has a private key.

The original massage is recovered always: the number of characters is not a limitation; however, the value of the private key must be increased if the number of characters in the message is increased too.

Acknowledgments. The authors would like to thank the Instituto Politécnico Nacional (COFAA, SIP and EDI), and SNI for their economic support to develop this work. Ángel Martínez is grantee PIFI.

References

1. Stanoyevitch, A.: Introduction to Cryptography with mathematical foundations and computer implementations. CRC Press, pp. 1−2 (2011)
2. McAndrew, A.: Introduction to Cryptography with open-source software. CRC Press, pp. 4−5 (2011)
3. Ayushi: A Symmetric Key Cryptographic Algorithm. International Journal of Computer Applications (0975−8887) vol. 1(15) (2010)
4. Solanki,K. H., Patel, C. R.: New Symmetric Key Cryptographic algorithm for Enhancing Security of Data. International Journal Of Research In Computer Engineering And Electronics. vol. 1(3) (Dec 2012)
5. Mathur, A.: A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms. International Journal on Computer Science and Engineering (IJCSE), vol. 4(9), Sep 2012, pp.1650−1657 (2012)
6. Kiran Kumar M., Mukthyar Azam, Shaik Rasool, S.: Efficient Digital Encryption Algorithm Based On Matrix Crambling Technique. International Journal of Network Security & Its Applications (IJNSA), vol. 2(4), October 2010, pp. 30−41 (2010)
7. Ramesh G., Umarani, R.: A Novel Symmetrical Encryption Algorithm with High Security Based On Key Updating. International Journal of Communication Engineering Applications-IJCEA, vol. 2(5), November-December 2011, pp. 329−341 (2011)
8. Buba, Z. P., Wajiga, G. M.: Cryptographic Algorithms for Secure Data Communication. International Journal of Computer Science and Security (IJCSS), vol. 5(2), pp. 227−243 (2011)
9. Hameed, S., Riaz, F., Moghal, R., Akhtar, G., Ahmed, A., Dar, A. G.: Modified Advanced Encryption Standard For Text And Images. Computer Science Journal vol. 1(3), December 2011, pp. 120−129 (2011)

10. Agrawal, M. and Mishra, P.: A Modified Approach for Symmetric Key Cryptography Based on Blowfish Algorithm. International Journal of Engineering and Advanced Technology (IJEAT), vol. 1(6), August 2012, pp. 79−83 (2012)
11. Kruti, S., Gambhava, B.: New Approach of Data Encryption Standard Algorithm. International Journal of Soft Computing and Engineering (IJSCE), vol. 2(1), March 2012, 322−325 (2012)
12. Jamgekar, R. S., Joshi, G. S.: File Encryption and Decryption Using Secure RSA, International Journal of Emerging Science and Engineering (IJESE), 1(4), February 2013, pp. 11-14 (2013)
13. Abusukhon, A., Talib and Ottoum, M. I.: Secure Network Communication Based on Text-to-Image Encryption. International Journal of Cyber-Security and Digital Forensics (IJCSDF), vol. 1(4), pp. 263−271 (2012)
14. Steinbuch, K.: Die Lernmatrix, Kybernetik. vol. 1(1), pp. 36-45 (1961)
15. Willshaw, D., Buneman, O. and Longuet-Higgins, H.: Non-holographic associative memory. Nature, vol. 222, pp. 960−962 (1969)
16. Anderson, J. A.: A simple neural network generating an interactive memory. Mathematical Biosciences, vol. 14, pp. 197−220 (1972)
17. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, vol. 79, pp. 2554−2558 (1982)
18. Ritter, G., Sussner, X. P. and Diaz de León, J. L.: Morphological Associative Memories. IEEE Transactions on Neural Networks vol. 9, pp. 281−293 (1998)