International Journal of Soft Computing 3 (5): 390-396, 2008
ISSN: 1816-9503

A New Technique on Neural Cryptography with Securing of Electronic Medical Records in Telemedicine System

[1]N. Prabakaran, [2]P. Saravanan and [1]P. Vivekanandan
[1]Department of Mathematics, Anna University, Chennai-600 025, India
[2]Department of Information Technology, Higher College of Technology, Sultanate of Oman

Abstract: There is a necessity to secure the Electronic Medical Records (EMR) when the exchange of medical information is taken place among the patients and doctors. We can generate a common secret key using neural networks and cryptography. Two neural networks which are trained on their mutual output bits are analyzed using methods of statistical physics. In the proposed Tree Parity Machines (TPMs), hidden layer of each output vectors are compared. That is, the output vectors of hidden unit using Hebbian learning rule, left-dynamic hidden unit using Random walk learning rule and right-dynamic hidden unit using Anti-Hebbian learning rule are compared. Among the compared values, one of the best values is received by the output layer. Similarly, the other hidden units, left-dynamic hidden units and right-dynamic hidden units perform the same operations and values are received by the output layer. The output layer receives the inputs from the hidden layer and then calculates the weights using different transfer functions, which reduce the feedback mechanism because each output is compared. Then the best compared weight is updated in the output unit. The EMR is compressed using Huffman compression, the CEMR (Compressed EMR) which is based on password-protection from the combination of lower layer's spy unit vector and upper layer's spy unit vector. A network with feedback generates a secret key, which can be used to encrypt and decrypt the CEMR using Rijndael Algorithm. Also, the timing to break a secret key using brute force attack is also explained in this study.

Key words: Neural cryptography, electronic medical records, Huffman compression, Rijndael algorithm

INTRODUCTION

Some studies have shown how to generate a secret key over a public channel for exchange of secret message. Recently, it has been shown how to use synchronization of neural network by mutual learning to generate secret keys over public channel and this algorithm is called neural cryptography (Kinzel and Kanter, 2002).

Telemedicine represents a valuable resource for delivering health-related services to remote, suburban areas, providing greater access to health care for consumer and health professionals. The real-time telemedicine is called 'store and forward', in which EMR information is sent to a provider at a distant site for their evaluation. This does not allow for a dialogue between the patient and doctor (http://www.utahtelehealth.net/faqs.html).

Huffman compression reduces the average code length of the EMR which is used to represent the symbols of an alphabet. Symbols of the source alphabet which occur frequently are assigned with short length codes. The general strategy is to allow the code length to vary from character to character and to ensure that the frequently occurring character have shorter codes.

Rijndael algorithm (Advanced Encryption Standard) is symmetric since the same secret key (256 bits) is used for encryption and decryption of the CEMR. The input data block is 512 bits. The secret key effectively doubles the strength of an algorithm, when defined as the time necessary for an attacker to stage a brute force attack (Sharma and Sudarshan, 2005).

NEURAL SYNCHRONIZATION

The weight vectors of the two neural networks begin with random numbers. The partners A and B receive a common input vector at each time, their outputs are calculated and then communicated. If they agree on the mapping between the current input and the output, their weights are updated according to the learning rule.

A structure of tree parity machine: The TPMs consist of K-hidden units, K-left dynamic hidden units

Corresponding Author: N. Prabakaran, Department of Mathematics, Anna University, Chennai-600 025, India

(Prabakaran *et al*., 2008) and K-right dynamic hidden units each of them being a perceptron with an N-dimensional weight vector w. The lower layer spy unit is connected to the input units. The upper layer spy unit is connected to the hidden units (σ), left-dynamic hidden units (δ), right dynamic hidden units (ϒ), left-dynamic output unit (α_1), output unit (α_2) and right-dynamic output unit (α_3).

The lower layer and upper layer spy units receive the input values from the N-input units, K-left dynamic hidden units, K-right dynamic hidden units, K-hidden units and output unit with feedback mechanism.

The structure of this TPM is shown in Fig. 1. The components of the input vectors x are binary,

$$x_{ij} \in \{-1, +1\}, x_{im} \in \{-1, +1\}, x_{ik} \in \{-1, +1\} \quad (1)$$

and the weights are discrete numbers between -L and +L (Ruttor *et al*., 2006):

$$\begin{aligned} w_{ij} &\in \{-L, -L+1, \ldots\ldots, L-1, L\}, \\ w_{im} &\in \{-L, -L+1, \ldots\ldots, L-1, L\}, \\ w_{ik} &\in \{-L, -L+1, \ldots\ldots, L-1, L\} \end{aligned} \quad (2)$$

where, L is the depths of the weights of the networks.

The index i = 1,….., K denotes the ith hidden unit (σ) of TPM (Ruttor *et al*., 2004), m = 1, ….., K left-dynamic hidden unit (δ) of the TPM (Prabakaran *et al*., 2008), k = 1,…..., K right-dynamic hidden unit (ϒ) of the TPM and j = 1,…., N denotes the N components. The hidden layer transfer functions are given below:

$$\sigma_i = \sin\left(\sum_{j=1}^{N} w_{ij} \bullet x_{ij}\right) \quad (3)$$

$$\delta_i = \tanh\left(\sum_{m=1}^{N} w_{im} \bullet x_{im}\right) \quad (4)$$

$$\Upsilon_i = \arctan\left(\sum_{k=1}^{N} w_{ik} \bullet x_{ik}\right) \quad (5)$$

where, Eq. 3 is the transfer function of the hidden unit, the Eq. 4 is the transfer function of the left-dynamic hidden unit, the Eq. 5 is the transfer function of the right-dynamic hidden unit.

The transfer functions for lower layer spy unit and upper layer spy unit are given:

$$\vartheta = -\sin\left(\sum_{i=1}^{N}\left[\sum_{j=1}^{N} x_{ij}\right]\right) \quad (6)$$

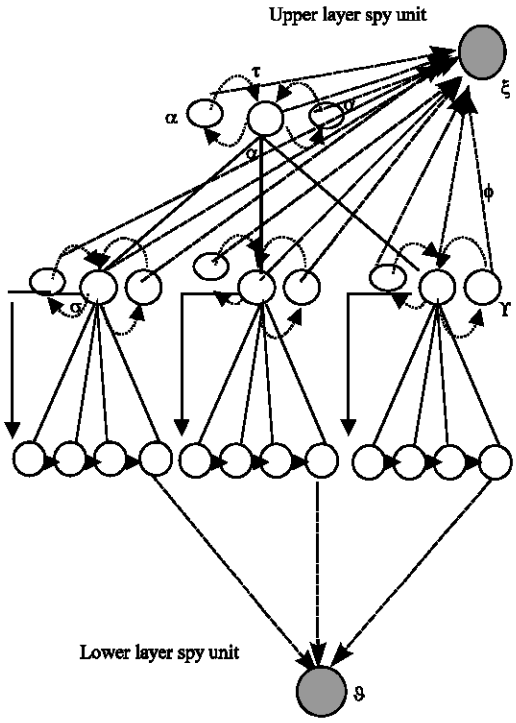

Fig. 1: A structure of tree parity machine with K=3, δ=3, ϒ=3, α=3, ξ=1, ϑ=1 and N=4

$$\xi = -\sin\left(\sum_{i=1}^{K} \delta_i \ \sigma_i \Upsilon_i \ \alpha_i\right) \quad (7)$$

where the Eq. 6 is the transfer function of the lower layer spy unit (Prabakaran *et al*., 2008) and Eq. 7 the transfer function of the upper layer spy unit.

The K-hidden units of σ_i, left-dynamic hidden units of δ_i and right-dynamic hidden units of Υ_i define common output bits of hidden layer of the network and are given by:

$$\beta_a = \prod_{i=1}^{K} \sigma_i \quad (8)$$

$$\beta_b = \prod_{i=1}^{K} \delta_i \quad (9)$$

$$\beta_c = \prod_{i=1}^{K} \Upsilon_i \quad (10)$$

where, Eq. 8 is the output for the hidden units, Eq. 9 the output for the left-dynamic hidden units and Eq. 10 the output for the right-dynamic hidden units.

The 2 TPMs compare the hidden layer's output bits (Prabakaran *et al*., 2008) and then update the weights from

hidden units, left-dynamic hidden units and right-dynamic hidden units as well as partners A and B that are trying to synchronize their weights.

$$\psi_i^{A,B} = \text{comp}\ (\beta_a\ ,\ \beta_b,\ \beta_c) \tag{11}$$

$$\phi_i^A = w_{ij}^A\ x_{ij}^A\ \tau^B \psi_i^A \tag{12}$$

$$\phi_i^B = w_{ij}^B\ x_{ij}^B\ \tau^A \psi_i^B \tag{13}$$

where, Eq. 11 represents comparison of the output of hidden, left dynamic and right dynamic hidden units of A and B. The Eq. 12 and 13 represent output of hidden, left and right dynamic hidden units of A and B, respectively.

The transfer functions of the output layer are given below:

$$\alpha_1 = \sin\left(\sum_{i=1}^{K} \phi_i\right) \tag{14}$$

$$\alpha_2 = \tanh\left(\sum_{i=1}^{K} \phi_i\right) \tag{15}$$

$$\alpha_3 = \arctan\left(\sum_{i=1}^{K} \phi_i\right) \tag{16}$$

The Eq. 14 is the transfer function of left-dynamic output unit, the Eq. 15 is the transfer function of output unit and the Eq. 16 is the transfer function of right-dynamic output unit.

$$\tau^{A,B} = \text{comp}\ (\alpha_1, \alpha_2, \alpha_3) \tag{17}$$

where, Eq. 17 represent output layer vector from the comparison of output unit, left-dynamic output unit and right-dynamic output unit.

Learning rules: The partners have their own networks with a same TPM architecture. Each party selects a random initial weight vectors w_i (A) and w_i (B) at t = 0.

The 2 TPMs are trained by their mutual output bits τ^A and τ^B as well as receive common input vectors x_i and corresponding output bit τ of its partner at each training steps.

The following are the learning rules:

- If the output bits are different, $\tau^A \neq \tau^B$, nothing is changed.
- If $\tau^A = \tau^B \equiv \tau$, the hidden, left and right dynamic hidden units are trained which have an output bit identical to the common output $\phi_i^{A/B} = \tau^{A/B}$.
- To adjust the weights, we consider three different learning rules. They are:

- Hebbian learning rule for hidden units:

$$\begin{aligned} w_i^A(t+1) &= w_i^A(t) + x_i \tau^A \Theta(\tau^A \phi_i^A) \Theta(\tau^A \tau^B) \\ w_i^B(t+1) &= w_i^B(t) + x_i \tau^B \Theta(\tau^B \phi_i^B) \Theta(\tau^A \tau^B) \end{aligned} \tag{18}$$

where, Θ is the Heaviside step function (Engel and Broeck, 2001), if the input is positive then the output is 1 and if input is negative then the function evaluates to 0.

- Random walk learning for left-dynamic hidden units:

$$\begin{aligned} w_i^A(t+1) &= w_i^A(t) + x_i \Theta(\tau^A \phi_i^A) \Theta(\tau^A \tau^B) \\ w_i^B(t+1) &= w_i^B(t) + x_i \Theta(\tau^B \phi_i^B) \Theta(\tau^A \tau^B) \end{aligned} \tag{19}$$

- Anti-Hebbian learning for right-dynamic hidden units:

$$\begin{aligned} w_i^A(t+1) &= w_i^A(t) - \phi_i x_i \Theta(\tau^A \phi_i^A) \Theta(\tau^A \tau^B) \\ w_i^B(t+1) &= w_i^B(t) - \phi_i x_i \Theta(\tau^B \phi_i^B) \Theta(\tau^A \tau^B) \end{aligned} \tag{20}$$

Order parameters: The size of a matrix F is (2L+1)×(2L+1) in TPMs (Rosen-Zvi *et al*., 2002). Their elements are $F^i(\mu)$, $F^j(\mu)$ and $F^k(\mu)$ where 'μ' is the state of the machines in the time step, 'i' is hidden units, where 'j' is the left-dynamic hidden units and 'k' is the right-dynamic hidden units. The element f_{qr}^i of matrix stands for the matching components in the ith weight-vector in which the A's components are equal to 'q' and the matching components of B are equal to 'r'. The element f_{st}^j, matching components of jth weight-vector in which the A's components are equal to 's' and the matching components of B are equal to 't'. The element f_{uv}^k, matching components of kth weight-vectors in which the A's components are equal to 'u' and the matching components of B are equal to 'v'. The values of q, r, s, t, u and v belong to any one of the values of -L, …. -1, 0, 1,…..., L. The overlap of the weights belonging to the ith hidden unit (Ruttor *et al*., 2006), jth left-dynamic hidden unit and kth right-dynamic hidden unit in the 2 parties are given below:

$$R_i^{A,B} = \frac{W_i^A \cdot W_i^B}{N},\ R_j^{A,B} = \frac{W_j^A \cdot W_j^B}{N},\ R_k^{A,B} = \frac{W_k^A \cdot W_k^B}{N} \tag{21}$$

Also their norms

$$Q_i = \frac{W_i^A \cdot W_i^A}{N},\ Q_j = \frac{W_j^A \cdot W_j^A}{N} \text{ and } Q_k = \frac{W_k^A \cdot W_k^A}{N}$$

are hidden, left and right-dynamic hidden units of A's TPM, respectively.

$$Q_i = \frac{W_i^B \cdot W_i^B}{N},\ Q_j = \frac{W_j^B \cdot W_j^B}{N} \text{ and } Q_k = \frac{W_k^B \cdot W_k^B}{N}$$

are hidden, left and right-dynamic hidden units of B's TPM, respectively. They are given by the matrix elements

$$R_i^{A,B} = \sum_{q,r} qr\ f_{qr}^i \tag{22}$$

$$R_j^{A,B} = \sum_{s,t} st\ f_{st}^j \tag{23}$$

$$R_k^{A,B} = \sum_{u,v} uv\ f_{uv}^k \tag{24}$$

The Eq. 22-24 represents overlap between 2 hidden units, 2 left-dynamic hidden units and 2 right-dynamic hidden units of A and B, respectively.

$$Q_i^A = \sum_{q,r} q^2 f_{qr}^i, \quad Q_i^B = \sum_{q,r} r^2 f_{qr}^i \tag{25}$$

$$Q_j^A = \sum_{s,t} s^2 f_{st}^j, \quad Q_j^B = \sum_{s,t} s^2 f_{st}^j \tag{26}$$

$$Q_k^A = \sum_{u,v} u^2 f_{uv}^k, \quad Q_k^B = \sum_{u,v} v^2 f_{uv}^k \tag{27}$$

The Eq. 25-27 represent weight distribution of hidden units, left-dynamic hidden units and right-dynamic hidden units of A and B, respectively.

These overlaps (from Eq. 21-24) and norms (from Eq. 25-27) fixed the probabilities of deriving the same internal representation via the normalized overlap,

$$\rho_i^{A,B} = \frac{R_i^{A,B}}{\sqrt{Q_i^A Q_i^B}},\ \rho_j^{A,B} = \frac{R_j^{A,B}}{\sqrt{Q_j^A Q_j^B}} \text{ and } \rho_k^{A,B} = \frac{R_k^{A,B}}{\sqrt{Q_k^A Q_k^B}}$$

then

$$\rho_{ijk}^{A,B} = \frac{R_i^{A,B}}{\sqrt{Q_i^A Q_i^B}} + \frac{R_j^{A,B}}{\sqrt{Q_j^A Q_j^B}} + \frac{R_k^{A,B}}{\sqrt{Q_k^A Q_k^B}} \tag{28}$$

More precisely, the probability of having different results in the ith hidden unit, jth left-dynamic hidden unit and kth right-dynamic hidden unit of the partners A and B is given by the well-known generalization error for the perceptron $\varepsilon_p^i = 1/\pi \arccos(\rho_{ijk})$ (Engel and Broeck, 2001).

The quantity ε_p^i is a measure of the distance between the weight vectors of the corresponding hidden units, left-dynamic hidden units and right-dynamic hidden units. Since, the hidden unit, left-dynamic hidden unit, right-dynamic hidden units are independent, also the values ε_p^i determine the conditional probability P_r for a repulsive step and P_a for a attractive step between 2 hidden units (Ruttor *et al*., 2004), left-dynamic hidden units and right-dynamic hidden units given identical output bits of the 2 TPMs. In the case of identical distances $\varepsilon_p^i = \varepsilon$, the values of K, δ and Υ are found as K = 3, δ = 3 and Υ = 3.

$$P_a = \frac{1}{2}\frac{(1-\varepsilon)^9 + 3(1-\varepsilon)^3\varepsilon^2}{(1-\varepsilon)^9 + 9(1-\varepsilon)^3\varepsilon^2} \tag{29}$$

$$P_r = \frac{6(1-\varepsilon)^3\ \varepsilon^2}{3(1-\varepsilon)^9 + 9(1-\varepsilon)^3\varepsilon^2} \tag{30}$$

The Eq. 29 and 30 represents probability of attractive and repulsive steps between two hidden units, two left-dynamic hidden units and two right-dynamic hidden units of A and B, respectively.

Synchronization with feedback: The 2 TPMs A and B which start with different randomly chosen weight vectors 'w' and common randomly chosen input vectors 'x'. The feedback mechanism is defined as follows (Ruttor, 2006):

- The input is shifted at each step 't'. That is $x_{i,j}(t+1) = x_{i,j-1}(t)$ for $j > 1$.
- If $\tau^A(t) = \tau^B(t)$ then $x_{i,1}(t+1) = \phi_i(t)$, else $x_{i,1}(t+1)$ are reset to common values.
- If $\tau^A(t) \neq \tau^B(t)$ for R steps, then all input values are reset to common values.

These evaluations give some privacy to inputs and additionally system becomes sensitive about the learning rule. As described in Ruttor *et al*. (2006), learning rule of anti-hebbian will reveal less information than hebbian learning rule and random walk learning rule. Therefore, the anti-hebbian learning will be more appropriate for the feedback scheme.

OVERVIEW OF TELEMEDICINE AND EMR

The use of electronic information and tele-communication technologies to support long-distance

clinical health care, patient and professional health-related education, public health and health administration are explained (http://www.utahtelehealth.net/faqs.html). Telemedicine represents a valuable resource for delivering health-related services to remote, underserved areas, providing greater access to health care for consumers and health professionals.

Telemedicine mainly uses video conferencing equipment. This is an interactive technology and enables patients and health care providers at distant sites to interact 'face-to-face'. Technological advances now allow for these interactions to occur using a desktop computer.

An alternative to real-time telemedicine is called 'store and forward', in which clinical and patients care (medical records) information is sent (like email) to a provider at a distant site for their evaluation. This does not allow for a dialogue between the patient and provider.

The basic five principles of transition process of EMR are given (CPSA, 2004).

Patient information must be secured: The security of paper-based medical records is mainly based on physical security while EMR introduce many new issues and threats that must be considered. Effective security is a combination of administrative practices, physical security and technical security and data integrity, confidentiality and availability of the medical records.

Privacy of patient information must be maintained: Electronic records enable a dramatically enhanced capacity for the management of patient information. This increased potential needs to be evaluated in terms of the professional responsibility to maintain patient records and also the legal responsibilities as a custodian of health information.

The integrity of the medical record content must be maintained: Managing health information in a transition environment carries the risk that the quality of care may be adversely affected if the transition is not effectively managed. There will be a period of time within most practices where both paper and electronic records will be in use until all relevant patient data has been established in the EMR and all physicians in the practice use the electronic record.

The integrity of the clinical workflow supported by the medical record must be maintained: There are many clinical processes directly or indirectly supported by the medical record. The transition to electronic records may alter these processes which may include important safety precautions or other critical workflow items.

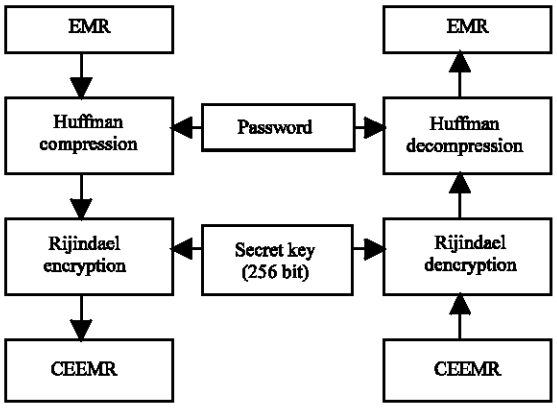

Fig. 2: The block diagram of a security process of an EMR

Continuity and quality of care must be maintained through the transition period: The overriding issue during this period of transition is that the level of patient care cannot suffer in a manner that risks patient safety or the quality of care.

An Electronic Medical Records (EMR) are secured electronic file of patient history, medical transcription notes, billing information and all other information necessary to have a complete patient profile.

From the Fig. 2, Electronic Medical Records are compressed using Huffman compression technique and then encrypted with Rijndeal algorithm. The Compressed Electronic Medical Records (CEMR) is protected by a password and then encrypted using a secret key (256 bit secret key).

HUFFMAN COMPRESSION

The algorithm narrows the alphabets for the file based on the pattern of that EMR and assigns the code for each character of an alphabet depending on the frequency of occurrence of that character. It assigns shorter code to the frequently used characters and longer code to the less-frequently used ones. Therefore, it reduces the number of bits used for each high-frequency character which may increase this number for low-frequency character (http://www.cra.org/Activities/craw/damp/awards). These assignments results in the CEMR to about 20-40%. The frequency table in the form of a text files is considered separately during the encoder. The CEMR is protected by a password, which is the combination of lower layer's spy unit vector and upper layer's spy unit vector.

RIJNDAEL ALGORITHM

The Rijndael takes 512-bits of CEMR data block as input and performs several transformations for encryption and decryption. These are represented as two-dimensional array of bytes. Rijndael encryption and decryption are based on four transformations that are performed repeatedly in a certain sequence. The total number of rounds is 24 for encrypting and decrypting the CEMR data block.

Add round key transformation: A Round key is added to the state by a simple bitwise XOR operation with portion of the expanded secret key.

SubBytes transformation: The SubBytes transformation is a non-linear byte substitution that operates independently on each byte of the state using a substitution table (S-Box), which is invertible, is constructed by composing 2 transformations (Sharma and Sudarshan, 2005):

- The multiplicative inverse in the Galois Field (2^8) with the irreducible polynomial is $m(x) = x^8 + x^4 + x^3 + x + 1$. The element {00} is mapped to itself.
- Apply the affine transformation (over GF(2^8)).

The inverse of SubBytes transformation, which is needed for decryption, is the inverse of the affine transformation followed by the same inversion as the SubBytes transformation.

Shiftrows transformation: The ShiftRows transformation rotates each row of the input state to the left shift and then offset of the rotation corresponds to the row number. The inverse of this transformation is computed by performing the corresponding rotations to the right shift.

Mixcolumns transformation: The Mixcolumns transformation operates on the state column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomial over GF (2^8) and multiplied modulo $x^4 + 1$ with a fixed polynomial a(x), given by:

$$a(x) = \{03\}\ x^3 + \{01\}\ x^2 + \{01\}x + \{02\}$$

The coefficient of a(x) are also elements of GF(2^8) and are represented by the hexadecimal values in this equation. The inverse mixcolumn transformation is the multiplication of each column with:

$$a^{-1}(x) = \{0B\}\ x^3 + \{0D\}\ x^2 + \{09\}\ x + \{0E\}$$

modulo $x^4 + 1$ for decryption process.

THE SECURITY OF AN EMR

The CEEMR has more secured during the transition due to Huffman Compression and Rijndael Encryption. Rijndael uses S-boxes as non-linear components. Rijndael appears to have an adequate security margin. On the other hand, the simple structure may have facilitated its security analysis during the timeframe of the AES development process.

There is currently no evidence that Rijndael has any weakness making any attack other than brute force attack. A secret key is 256-bit offers a sufficiently 1.16×10^{77} of possible keys. An exhaustive search is made using brute force attack to find the right key of the attacker about 10^{56} years.

CONCLUSION

In the proposed TPMs, the synchronize time of the attacker is increased by the three transfer functions in the hidden unit using Hebbian learning rule, left-dynamic hidden unit using Random walk rule and right-dynamic hidden unit using Anti-Hebbian learning rule. The output layer uses different transfer functions, which reduce the feedback mechanism. Also, the CEMR, which reduce the time to transmit large files and reducing the space required to store them on disk. The CEMR is password-protection from combination of lower layer's and upper layer's spy units vector, which increases the security of CEMR. The CEMR is fed as the input of Rijndael algorithm for encryption and decryption using 256 bit secret key. For the attacker to find all possible of secret key, it will take trillion years against the brute force attack.

REFERENCES

CPSA, 2004. Guideline. Transition to Electronic Medical Records (EMR), College of Physicians and Surgeons of Alberta.

Engel, A. and C.V.D. Broeck, 2001. Statistical Mechanics of Learning. Cambridge University Press, Cambridge.

Kinzel, W. and I. Kanter, 2002. Interacting neural networks and cryptography. Advances in Solid State Physics, by Kramer B. (Springer, Berlin) [cond-mat/0203011], 42: 383-391.

Prabakaran, N., P. Karuppuchamy and P. Vivekanandan, 2008. A New approach on Neural Cryptography with Dynamic and Spy units using multiple transfer functions and learning rules. Asian Journal of Information Technology, Vol. 4, No. 7.

Prabakaran, N., P. Loganathan and P. Vivekanandan, 2008. Neural Cryptography with Multiple Transfer function and Multiple Learning Rule. Int. J. Soft Comput., 3 (3): 177-181.

Rosen-Zvi, M., E. Kleign, I. Kanter and W. Kinzel, 2002. Cryptography based on neural networks-analytical results. Phys. Rev. E, 35 (47): L703-L713.

Ruttor, A., 2006. Neural Synchronization and Cryptography. Ph. D. Thesis.

Ruttor, A., I. Kanter and W. Kinzel, 2006. Dynamics of neural cryptography. [cont-mat/061257/ 21].

Ruttor, A., I. Kanter, R. Naeh and W. Kinzel, 2006. Genetic attack on neural cryptography. [cond-mat/0512022 v2/1].

Ruttor, A., W. Kinzel, L. Shacham and I. Kanter, 2004. Neural cryptography with feedback. Phys. Rev. E., 69: 1- 8.

Selective Encryption with text files, 2004. <http://www.cra.org/Activities/craw/damp/>/awards/2004/Phu/Documents/Technical%20Report.pdf.

Sharma, S. and T.S.B. Sudarshan, 2005. Design of an efficient architecture for advanced encryption standard algorithm using syslotic structures. International Conference of High Performance Computing (HiPC).

Utah Telehealth Network, Telemedicine FAQ's, 2008. University of Utah Health Sciences Center. <http://www.utahtelehealth.net/faqs.html>.