SOM-PAD: Novel Data Security Algorithm on Self Organizing Map

Marghny Mohamed[1], Abeer Al-Mehdhar[2] and Mohamed Bamatraf[3]

[1]Faculty of Information and Computers, Assiut University, EGYPT
marghny@aun.edu.eg
[2]Faculty of Science Hadhramout University of Science and Technology, Yemen
apora_18@yahoo.com
[3]Faculty of Science Hadhramout University of Science and Technology, Yemen
mbamatraf1@yahoo.com

ABSTRACT

Data security is one of major challenges in the recent literature. Cryptography is the most common phenomena used to secure data. One main aspect in cryptography is creating a hard to guess cipher. Artificial Neural Networks (ANN) is one of the machine learning techniques widely employed in several fields based on its characters, depending on the application area. One of these fields is data security. The state of art in this paper is the use of self organizing map (SOM) algorithm concept as a core idea to construct a pad; this pad is used to generate the cipher at one end. At the other end of communication the same process is synchronized to generate the same pad as the deciphering key. The security of the proposed model depends on the complex nature of ANN's. The algorithm could be categorized under symmetric cryptography, merging both stream and block cipher. A modified version of the same algorithm also presented employs permutation and variable SOM neighborhoods. The proposal can be applied over several file formats like videos, images, text files, data benchmarks, etc as show in experimental results.

KEYWORDS

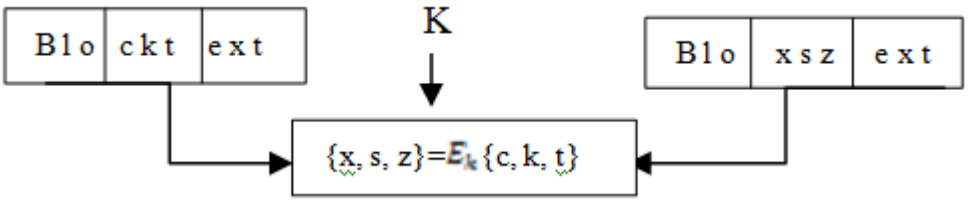
Self organizing map, Cryptography, Steganography, permutation, block cipher, stream cipher.

1. INTRODUCTION

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. A block cipher takes as input a block of plaintext and a key to generate a block of cipher, usually, of the same size. Since messages are almost always longer than a single block, several blocks can also be merged. We should clearly consider that security of a system can't depend on how plaintext is grouped/arranged rather than the operation(s) used to generate the cipher; but it is somehow considered during the cryptanalysis process. The most commonly used and secured algorithms uses block based ciphers. The Data

Sundarapandian et al. (Eds): CoNeCo,WiMo, NLP, CRYPSIS, ICAIT, ICDIP, ITCSE, CS & IT 07,
pp. 325–333, 2012. © CS & IT-CSCP 2012 DOI : 10.5121/csit.2012.2431

Encryption Standard (DES) [1] and the Advanced Encryption Standard (AES) [2] are blocking based ciphers, which have been designated cryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted). Despite its deprecation as an official standard, DES (especially its still-approved and much more secure triple-DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access. Many other block ciphers have been designed and released, with considerable variation in quality example of block cipher:



Stream based ciphers, in contrast to the 'block' type, create an arbitrarily stream of data units, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad [3]. The output of stream cipher is generated iteratively as ciphering process operates depending on the internal state of that process. The internal changes based on the key and the operation(s). Stream based ciphers also showed high acceptance. RC4 [4] is a widely used stream cipher.

Finally, we can use block cipher as stream if and vice versa, if the stream is formed from set of blocks, or the block is generated and encrypted as a stream of data units.

Speaking about the vulnerability of any algorithm must consider the set of operations used in the ciphering and deciphering process. The most classical techniques are substitution and transposition. Surprisingly a wide range of computer security systems employ these techniques. Permutation is one of the substitution techniques. It is the first step in the DES is to permute the input. In fact, permutations, along with substitutions, form the backbone of all modern cryptographic systems. In this article, simple permutation is applied to improve the security of proposed algorithm.

Hiding a secret message with in a larger object in such a way that others cannot discern the presence of the hidden message is another approach called steganography. As an example, it is possible to embed a text inside an image or an audio file. It is common enough to not attract any attention. One of the mostly used steganographic techniques is the Least Significant Bit (LSB) [5], modifies the rightmost bit(s) in each byte by replacing it with a bit(s) from the secret message.

Even though the proposed system hides data in a medium, employing a change in the first LSB, but it is not concerned with what changes in the pad (cover medium), aiming high confusion and diffusion.

Steganography can be used in a large amount of data formats. The most popular data formats used are .bmp, .doc, .gif, .jpeg, .mp3, .txt and .wav.

To make it quick and easy we experimented over gray image and most common datasets benchmarks.

1.1 Cryptography versus steganography :

The science of data encryption in order that the enemy does not understand the content of communication is called cryptography. To increase data security, data can be concealed or hidden within another object before sending; the technique used to hide the information is called Steganography as mentioned previously. It enables you to send sensitive information across insecure networks (such as the Internet).

However, steganography and cryptography differ in the way they are evaluated: steganography fails when the ”enemy” is able to access the content of the cipher message, while cryptography fails when the ”enemy” detects that there is a secret message present in the steganographic medium[6].

Here are some points discuss cryptography versus steganography:

- Common technology: Most algorithms known to government departments Strong algorithm are currently resistant to brute force attack Large expensive computing power required for cracking technology increase reduces strength in cryptography, Conversely the contrary in steganography little known technology still being developed for certain formats once detected message is known Many Carrier Formats.
- Steganography can be used in a large amount of data formats. The most popular data formats used are .bmp, .doc, .gif, .jpeg, .mp3, .txt and .wav, although the cryptography using mathematics to encrypt and decrypt data. It enables you to send sensitive information across insecure networks (such as the Internet).
- Steganography can be said to protect both messages and communicating parties, In contrast cryptography protects the contents of a message.

And hence comes our contribution to integrate both techniques, as cryptography, we could achieve the security in terms of confusion and diffusion, at the other side data is concealed within a map changing the major of its features, at the same time the process of creating the cipher comes from the steganography sense (i.e. embedding).

2. BACKGROUND

2.1 Self Organizing Map:

It is one of the widely applied neural networks and has some interesting features over other neural networks. It is unsupervised learning method and computationally simple. It produces a topology preserving mapping between high dimensional input space and low dimensional map space [7].

When you try to associate SOM with security you will find that it has been used so far as a classifier or clustering tool mostly in intrusion detection/prevention systems. Other categories of ANN's have also been employed in data and network security disciplines. In very few works they have been used in hashing [8], symmetric encryption [2], authentication [9], and semi-public key systems [10]. But they are commonly in network security as intrusion behavior classification and

alert correlation see [11, 12, and 13]. To our knowledge SOM has never been used as neither hashing, encryption, nor authentication.

2.1.1 SOM Algorithm:

Step 1: Initialization.
Set initial weights to small random values, say in an interval [0, 1], and assign a small positive value to the learning rate parametera.

Step 2: Activation and Similarity Matching.
Activate the SOM network by applying the input vector X, and find the winner-takes-all or best matching unit (BMU) neuron j_x at iteration p , using the minimum-distance Euclidean criterion

$$j_x(p) = \min_j \left\| \mathrm{X} - \mathrm{W_j}(\mathrm{p}) \right\| = \left\{ \sum_{i=1}^{n} [\mathrm{X_i} - \mathrm{W_i}(\mathrm{p})]^2 \right\}^{1/2}, \quad \mathrm{j} = 1,2,\cdots,\mathrm{m} \qquad (1)$$

Where n is the number of neurons in the input layer, and m is the number of neurons in the SOM layer.

Step 3: Learning.
Update the weights

$$W_{ij}(p+1) = W_{ij}(p) + \Delta W_{ij}(p) \qquad (2)$$

Where $\Delta W_{ij}(p)$ is the weight correction at iteration p.The weight correction is determined by the competitive learning rule:

$$\Delta \mathrm{W_{ij}}(\mathrm{p}) = \begin{cases} \alpha[\mathrm{X_i} - \mathrm{W_{ij}}(\mathrm{p})], & \mathrm{j} \in \Lambda_\mathrm{j}(\mathrm{p}) \\ 0, & \mathrm{j} \in \Lambda_\mathrm{j}(\mathrm{p}) \end{cases} \qquad (3)$$

Where α is the learning rate parameter, and $\Lambda_j(p)$ is the neighborhood function centered around the BMU neuron j_x at iteration p.

Step 4: *Iteration*.
Increase iteration p by one, go back to Step 2 and continue until the minimum-distance Euclidean criterion is satisfied, or no noticeable changes occur in the feature map.

After all of the input is processed (usually after hundreds or thousands of repeated presentations), the result should be a spatial organization of the input data organized into clusters of similar (neighboring) regions.

2.2 Permutations [14]

A permutation is an ordered arrangement of events. There are two types of it with replacement and without replacement.

$$\frac{\mathrm{n!}}{(\mathrm{n}-\mathrm{r})!} = (\mathrm{n}-1)(\mathrm{n}-2)(\mathrm{n}-3)(\mathrm{n}-\mathrm{r}+1)$$

With replacement = once an event occurs, it can occur again (after you roll a 6, you can roll a 6 again on the same die).

Without replacement = an event cannot repeat (after you draw an ace of spades out of a deck, there is 0 probability of getting it again). When the order of items matters, that's called a permutation.

Since we are *not* allowed to repeat items, we use the following formula: Number of possible permutations

$$\frac{n!}{(n-r)!}$$

If you have a collection of n distinguishable objects or locations, then the number of ways you can pick/select a number r of them ($r < n$) is given by the permutation relationship:

$$nPr = \frac{n!}{(n-r)!}$$

A binary word has its bits reordered (permuted), the re-ordering forms the key if use n bit words, the key is n! bits, which grows more slowly.

Permutations have been widely used in both steganography and cryptography as well like in [15 and 16].

3. PROPOSED MODEL (SOM-Pad)

Every cryptographic system involves two ends, sender and receiver. In symmetric crypto-systems both ends share the same secret called the private key. In ideal cases, using the same key both ends should communicate securely in any public unsecure communication channel.

3.1 Key generation

Both ends share the same key which is a triplet (DS, It, In) denoted as K (DS, It, In), where:

Ds is the data set used to in SOM training. It can be image video, audio, a benchmark dataset like

IRIS [17], MONK [17], etc. It can also be any structured randomly generated file.

Ds = {{x0}, {x1}…{xn}}, n = number of input samples (instances), in case of images it is the height of the image in terms of pixels, whereas it is the number of records in datasets and structured files.

Len {xi} is the number of attributes / fields in each instance, in case of images it is the width of the image in terms of pixels.

It is the number of iterations used in SOM training to generate the trained map.

In is the initial value for the SOM network, it is a two dimension vector space of length h x w, h and w are the height and the width of the map respectively. Every node in the map Ij is of the same structure as xi.

3.2 Encryption

Step 1: Generate SOM map $S_{hw,}$ h is the height and w is the map width, and set the initial value for all the nodes to In.
Step 2: Let M plaintext stream message of length L,

$$0 < L < (h \times W)$$
$$M = \{m_0, m_{1,\dots,} m_l\};\ m_i \epsilon \{ {}^{0}/_{1} \}$$

Randomly select any node n_i ;$n_i \in S$ Note that n_i is a set of bits of length k, k>0, Convert the LSB from 0 to 1 or vice versa.
Step 3: select the neighbor nodes at level j for every node in the neighborhood of $n_i$freeze the lsb; embed the for the rest k-1 bits set their value to the plaintext stream sequentially until the set M is empty if no more nodes repeat this step for the rest of M.
Step 4: the newly modified map S' represents the cipher.

3.3 Decryption

Step 1: repeat step 1 as encryption.
Step 2: compare the lsb in both S and S'; for all nodes where unmatched is found repeat steps 3 to 4.
Step 3: select the *k* neighbor node of the select node in S' .
Step 4: drop the lsb from each neighbor nodes.
Step 5: construct the stream from the left k-1 bits sequentially the generated stream M' will identical to M representing the original message.

4. MODIFIED (SOM-Pad)

Note that every central node n_i is used to decide the location to hide the data in a single order we here suggest two more options for data placement to improve cipher complexity without any time overhead to the algorithm. First the variable value of k and second is the permutation of distributing data over the neighborhood nodes. To store the meta-data for these two variables the central node bits are used. As the first lsb is reserved as a flag to indicate whether the node is central or a simple data node; the next two bits used to indicate the value of k. finally the rest k-4 bits used to store the order of embedding plain text data blocks.

5. EXPERIMENTAL RESULT

For system evaluation we used some of standard grey scale images (baboon, lenna, white house, and camera man) as an application Figure 1. Cover images.

Figure 1. Cover images.

Two plain text streams were used:
"Hello hello"
"Attack is postponed until tomorrow night"
These selected streams are used to check the repeated occurrences of characters and words as a metric for confusion and diffusion level.
The correlation between the occurrence of the same character of block of characters over the same cipher, then the same is drawn between the plain text and the cipher, finally the correlation between different ciphers for the same plaintext message over different pads. From the results we can clearly notice the insignificant correlation between all the above mentioned factors as shown in Figures (2, 3). This practically/experimentally indicates the high level of confusion and diffusion presented in the generated cipher.

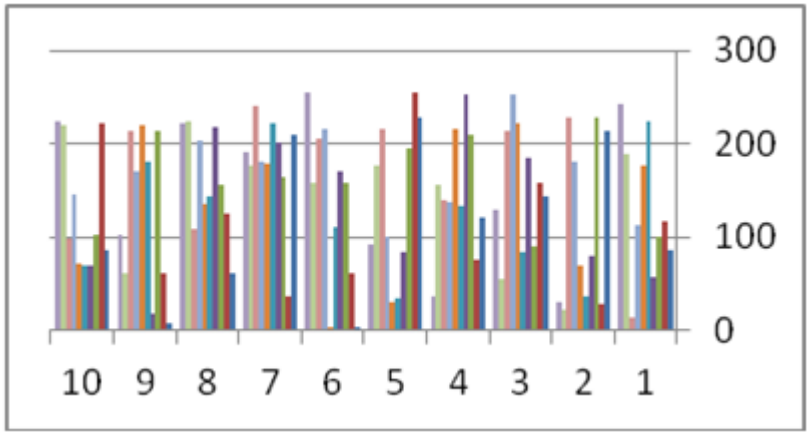


Figure 2. Map frequency diagram before embedding.

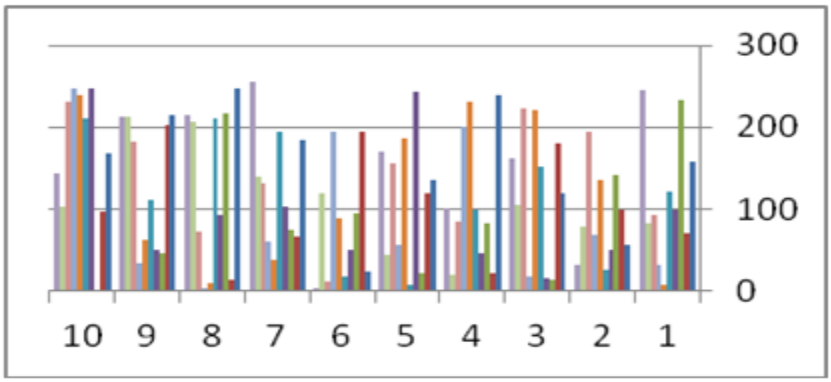


Figure 3. Map frequency diagram after embedding

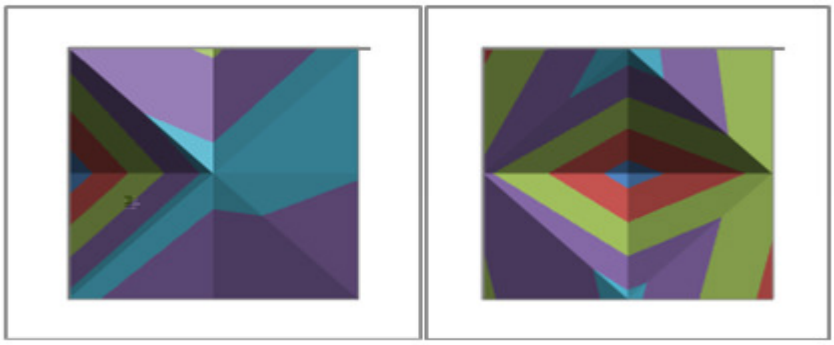


Figure 4. Contour diagram for the pad before and after embedding

6. SYSTEM COMPLEXITY

Cryptanalysis is the process of decrypting messages or breaking codes and ciphers.
The complexity of guessing the plaintext message in case of unknown plaintext can be measured as multiple of 3 equations
SOM:
Minimum error:

$$E_{min} = \frac{1}{2}\{\sum_{i=1}^{n}[X_i - W_{ij}(p)]^2\} \quad (1)$$

Neighboring hood function:

$$\Delta W_j = \eta y_{jx} \quad (2)$$

Where W_j is the weight vector, ηthe learning rate, y_j the output response, and x the input vector.

Permutation place equations:

$$\frac{n!}{(n-r)!} = (n-1)(n-2)(n-3)(n-r+1) \quad (3)$$

7. CONCLUSION

This paper introduces a data security system employing the self organizing map neural network. The SOM is used to create the pad for embedding the data and regenerated later at deciphering end. To increase system security permutation and neighboring level are also used. The system is tested using image datasets and the IRIS benchmark data as well. Results show high level of confusion and diffusion. At a glance the analysis of the system shows an acceptable time complexity. The proposed system can be applied over several file formats as a pad like audio, video, images, any structured data sets and file, etc. Further research can target the improvement of system applicability. The system keeps the ability of applying machine learning techniques to create complex ciphers open for further researches.

REFERENCES

[1] Seung-Jo Han, Heang-SooOh and Jongan Park .1996. The improved Data Encryption Standard (DES) Algorithm. IEEE. 3(1996), 1310-1314.

[2] Behrouz A. Forouzan. 2008. Cryptography and Network Security. McGraw-Hill, ISBN: 978-007-126361-0.

[3] [http://users.telenet.be/d.rijmenants/en/onetimepad.

[4] Allam Mousa and Ahmad Hamad.2006. Evaluation of the RC4 Algorithm for Data Encryption. International journal of computer science and application. 3(2).

[5] Nameer, N. EL-Emam. 2008. Embedding a Large Amount of Information Using High Secure Neural Based Steganography Algorithm. International Journal of Information and Communication Engineering. 4(2).

[6] Kumar, M., Siddique, S. and Noor, H., 2009. Feature-based alert correlation in security systems using self organizing maps, in Proceedings of SPIE, 734404-734404.

[7] Lau, K.W., Yin, H. and Hubbar, S., 2006. Kernel self-organising maps for classification. Neurocomputing. 69, 2033–2040.

[8] Mihir Bellare, Ran Canetti and Hugo Krawczyk .1996. Keying Hash Functions for Message Authentication. Springer-Verlag,1109.

[9] Shengbao Wang, Zhenfu Cao and Haiyon Bao. 2008. Efficient Certificateless Authentication and Key Agreement (CL-AK) for Grid Computing, International Journal of Network Security, 7(3), 342–347.

[10] Michael Malkin and Ton Kalker. 2006. A Cryptographic Method for Secure Watermark Detection. Springer, 4437, 26-41

[11] Venkatachalam,V., and Selvan,S., 2007. Intrusion Detection using an Improved Competitive Learning Lamstar Neural Network, IJCSNS International Journal of Computer Science and Network Security, 7(2).

[12] Vokorokos,L., Balaz, A. and Chovanec,M., 2006. INTRUSION DETECTION SYSTEM USING SELF ORGANIZING MAP", Acta Electrotechnica et Informatica, 6(1).

[13] Nouha Oualha, Melek Önen and Yves Roudier,2008. A Security Protocol for Self-Organizing Data Storage, International Information Security Conference, doi: 10.1007/978-0-387-09699-5_44, 675-679.

[14] http://www.intmath.com/counting-probability/3 permutations.php.

[15] Shihchun Tu, Hungwei Hsu and Wenkai Tai 2010, Permutation Steganography for Polygonal Meshes Based on Coding Tree, The International Journal of Virtual Reality, 9(4), 55-60.

[16] Zhijie Shi and Ruby B. Lee, 2000. Bit Permutation Instructions for Accelerating Software Cryptography, IEEE International Conference on Application-specific Systems, 138-148, 2000.

[17] Odajima, K., 2008, Greedy rule generation from discrete data and its use in neural network ruleextraction, Neural Networks, vol. 6,1833-1839