Research Journal of Applied Sciences, Engineering and Technology 7(4): 740-744, 2014
DOI:10.19026/rjaset.7.311
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: March 11, 2013 Accepted: April 12, 2013 Published: January 27, 2014

Research Article

Appliance of Neuron Networks in Cryptographic Systems

Mohammed Al-Maitah
King Saud University, Saudi Arabia

Abstract: This study is dedicated to the examination of a problem of postquantum encryption algorithms which are connected with a potential crisis in modern cryptography that is caused by appearance of quantum computers. General problem formulation is given as well as an example of danger from the quantum algorithms against classical cryptosystems. Existing postquantum systems are analyzed and the complication of their realization and cryptosecurity are estimated. Among the others algorithms on the basis of neural networks are chosen as a starting point. The study demonstrates neuro cryptographic protocol based on a three-level neural network of the direct propagation. There was evaluated it's cryptosecurity and analyzed three types of this algorithm attack to show the reality of the hypothesis that neuro cryptography is currently one of the most promising post quantum cryptographic systems.

Keywords: Cryptography algorithms, neural network, quantum computer

INTRODUCTION

Modern algorithms of cryptography are based on the theory of numbers. Almost all methods for the creation of the keys of the sufficient length for secure encryption turn either to the generation of pseudo-random sequences or to the prime numbers expansion. In particular, prime numbers of big orders are used by the most popular method of the enciphering by the open key RSA. Nowadays this algorithm is cryptosecure, does not undergo cracking despite the increase in the speed of calculation equipment and the appearance of the new classes of the attacks which use the hardware peculiarities of the computers which generate the keys (Pellegrini *et al*., 2010).

However the main danger for the modern cryptography does not proceed from the sphere of new methods of cryptographic attacks, but from the other sphere which now is of experimental character which soon may become the reason for the new scientific and technical revolution. These are quantum calculations. Of course we are not talking about quantum computer but the messages on the creation of the first programmed computers of the such type are in Colorado (Colin, 2009) and in IBM Research, (2012).

Such developments as they are may question all the achievements in cryptography during the recent years. For example there is a quantum algorithm which sharply (in some cases exponentially) accelerates the determination of the prime factors for many-digit number, this is a so-called Shor's algorithm (Shor, 1994). The possibility of its realization on quantum computer even on the simplest one means the increase in the efficiency of the attacks onto the systems based on RSA.

Of course not all of the cryptographic systems are assailable for such attacks. There exist other cryptographic algorithms called post quantum because the theoretical calculations show there resistance to Shor's algorithm and to "frontal" attacks by the complete sorting. To this class belongs McEliece system (Bernstein *et al*., 2008), but it's "post quantum" characteristic imposes some restrictions that make it usless for most users. For example, the length of the key in the McEliece system is very high - 2^{19} bits. Also, because of the key length the encrypted message will always be larger than the original, which requires a very accurate protocol of sending (as the identity of the received message to send one is important for asymmetric codes). This makes the McEliece system practically unacceptable for the user's workstations.

Post quantum cryptographic algorithms are also the systems based on the theory of groups (the so-called lattice codes) (Micciancio and Regev, 2009). Lattice-based cryptography. They are based on the assumption that quantum algorithms for the lattice codes attack do not exist. At least the quantum algorithm, which would show better results than the classic one wasn't found yet. However, this complexity turns into the argument against masses.

One of the few post quantum approaches in cryptography with the prospect of becoming widespread is the use of neural networks. Earlier they were used to create new types of attacks on existing codes, based on the idea that any function can be represented as a neural network that can explore the solution space. This allows to solve many problems

related to cryptography-in particular, hashing and generating pseudo-random arrays.

Proposed in 1995 by Sebastien Dourlens neural cryptography is based on the applicability of neural networks to solve such problems. This French mathematician used neural networks for rearrangements in DES-algorithm, thus demonstrating the applicability of the fundamental neural network in the encryption area (Sébastien Dourlens Neuro-applied Cryptography, 1995-1996). Since then, more researches have been carried out, including the cryptography protocol with the public key, developed by Khalil (2012). This study attempted to cross the usual system of creating public and private keys with neural network algorithms. Generating of the private key by this approach was made by the usual methods and the public key presented a neural network that gains the ability to decrypt the message learning by back-propagation. However, this method has a disadvantage-for large amounts of data the time of neural network training is sharply increasing.

This are not the only examples of neural networks application in cryptography: there are a lot of works, including the original-for example, the work of Taiwanese scientists (Tai-Wen and Suchen, 2001) suggested the use of Hopfield neural network for image encryption. Unfortunately, however, they are largely theoretical and yet are of limited practical use.

EXPERIMENTAL

Cryptographic protocol based on neural networks: To give an idea of how useful neural networks are, let's consider the example described by Oscar and Karl-Heinz, 2010). They describe an interesting change of Diffie-Hellman algorithm, which is used for the exchange of two keys. Reyes-Zimmermann model consists two Tree Parity Machines (TPM), which are synchronized independently and thus they get a key. Synchronization mechanism is constructed similar to the synchronization of two chaotic oscillators (it's mathematical description is very complex and beyond to the scope of this study).

So, suppose that there is some TPM that contains three levels, as shown in Fig. 1.

As is seen in the diagram it is neural network of direct distribution and it contains three layers-levels. The input level is composed of $K \cdot N$ binary neurons, which are described by the formula:

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

Between the output neuron and an array of input neurons there is the so-called "hidden" level, which isn't binary and has the following weight:

$$w_{ij} = \{-L, \dots, 0, \dots, +L\} \quad (2)$$

The value of each of the hidden neurons is calculated as the sum of the input values and weights. Note that the sign is calculated separately since zero is equated to negative.

$$\sigma_i = sign(\sum_{j=1}^{N} w_{ij} x_{ij}) \quad (3)$$

$$sign(x) = \begin{cases} -1, x =< 0 \\ 1, x > 0 \end{cases} \quad (4)$$

An output neuron also has an integer format and is the multiplication of all the hidden neurons. It is binary:

$$\tau = \prod_{i=1}^{K} \sigma_i \quad (5)$$

Now imagine that each of two parties A and B has a TPM. To synchronize them, use a bidirectional learning:

- Initialize random weight values
- Generate a random input vector
- Compute the values of the hidden neurons
- Compute the value of the hidden neuron
- Compare the outputs of TPM: if outputs are different, repeat the iteration

From step 2. If they are the same, we save the calculated weight

Let us compose a block diagram of such training (considering that the maximum weight is equal to L): (Fig. 2).

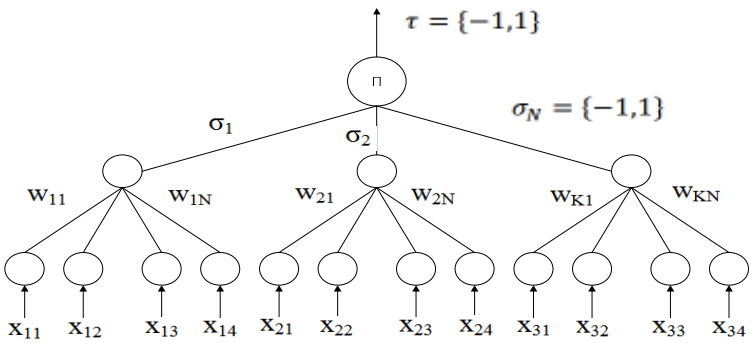

Fig. 1: Neural network TPM

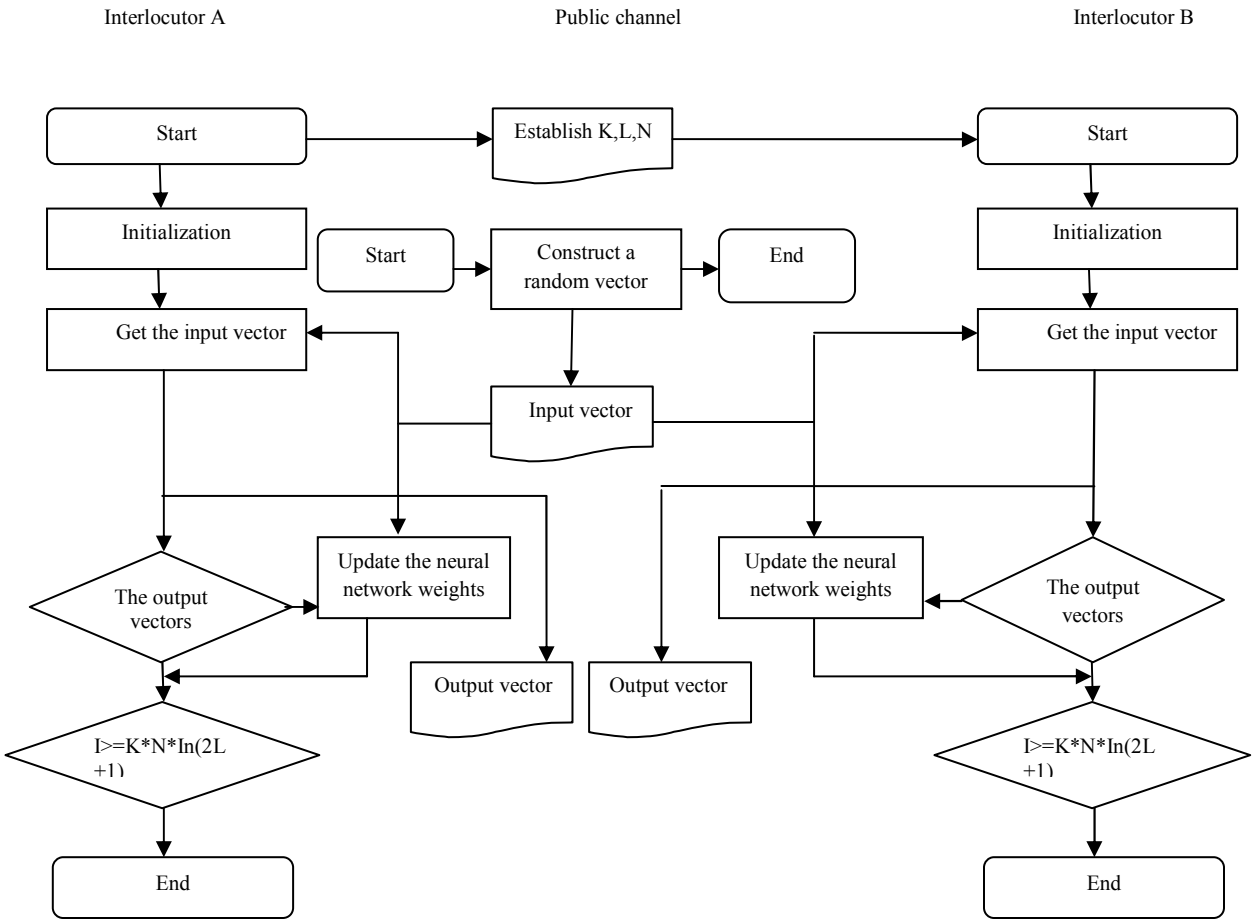

Fig. 2: Block diagram of learning

If there is achieved a full synchronization, that is the weights of hidden neurons in both TPM are identical than both parts can use these weights as the key. To update the weights in the process or periodic key change, you can use one of three options:

а) Hebbian learning rule:

$$w_i^+ = w_i + \sigma_i x_i \Theta(\sigma_i \tau)\Theta(\tau^A \tau^B) \quad (6)$$

б) Anti-Hebbian learning rule:

$$w_i^+ = w_i - \sigma_i x_i \Theta(\sigma_i \tau)\Theta(\tau^A \tau^B) \quad (7)$$

в) Random walk:

$$w_i^+ = w_i + x_i \Theta(\sigma_i \tau)\Theta(\tau^A \tau^B) \quad (8)$$

Thus, we have a relatively simple neural cryptographic protocol.

Attacks on neurocryptographic protocol: It is natural that neurocryptographic system, as well as any code protocol, is liable to attacks. Let's consider some of them supposing that a certain cryptanalyst has an access to messages between interlocutors, but can't change them. Let's also suppose that a cryptanalyst is informed of the fact that he deals with a neurocryptographic protocol and even has TPM at his disposal.

«Frontal attack»: All possible key combinations should be checked for the similar attack. The number of keys for K hidden neurons, K*N input neurons and maximum weight L is:

$$S = (2 \cdot L + 1)^{K \cdot N} \quad (9)$$

Let's calculate this rate for a neurocryptographic system on the Fig. 1. Let's suggest that K=3, L=3, N=10:

$$S = (2 \cdot 3 + 1)^{3 \cdot 10} = 7^{30} = 225393402906922580878632 49$$

As we see the quantity of searching is vast even for the smallest neurocryptographic systems. But if the neural network consists of at least 100 input elements, the frontal attack on the neurocryptogramme will require enormous computer timing.

Training of your own TPM: Let's suppose that a cryptanalyst has the same TPM as both interlocutors and this cryptanalyst reads their correspondence. By using dispatch patterns as initial vectors he can train his own TPM with the help of algorithm which is described in the previous chapter and continuously comparing his TPM's log off with the interlocutors' ones. In this case three situations appear (let's define the cryptanalyst with a letter E and analyze conclusions):

- A<>B. Nobody renews significance
- A = B = E. All three renew significance
- A = B<>E. A and B renew significance whereas E can't do this

According to the frequency of significance renewals (i.e., periodical keys change) it may be empirically shown that the cryptanalyst will always renew significance more slowly than requestors. Therefore, he can identify only the part of the key, but not entirely. This attack may be effective only in the case of a big quantity of analyzed messages or when interlocutors don't renew significance of their TPM at all.

Genetic attack: This method was offered by Klimov *et al*. (2002) and is effective only for small neural networks (N, K< = 3), because for the big ones (K>6) its complication and resource intensity increase exponentially. However, this is the only attack method which may be applied to quantum calculations and become a serious danger for the neurocryptography.

In fact this is a modify of a previous attack method which is based on the training of a TPM by its cryptanalyst and combined with genetic algorithms. The certain variety (which may be compared with the quantity of analyzed messages) of neural networks with a TPM structure is being created. After the iteration each of them exchanges its significance with its neighbors at random. After that those TPM which have failed are being canceled out. In such a way, in the process of cryptanalysis only those TPM which represent requestors' keys sufficiently will be always selected.

Let's consider the simplest case described in Andreas *et al*. (2006):

- **Initialization phase:** There is only one TPM with a random initial vector in the variety. A certain numeral M which is a "population" limit is also being fixed
- **Cryptanalysis phase:** It is divided into three situations as well as in the previous attack method:
  - Output values A and B are not equal, significance renewal doesn't happen, the population is not changing
  - A = B, thereby the quantity of TPM doesn't exceed the numeral M. All TPM are replaced for the new ones each of which becomes the significance substitution of one of the hidden neurons for the opposite rate. After that training takes place according to the Hebb's rule (6)
  - If A = B and the population limit exceeds the numeral M then all the TPM which are not equal to the received log off for A and B are being canceled out from the population

In Tai-Wen and Suchen (2001) it is shown that a similar approach has sense only for the small neurocryptographic systems and with the meanings N>100, K>100, L>10 it becomes nonsense since resource intensity will increase exponentially. At the same time this defect can be removed on the quantum computer (this is just an empiric assumption which has no facts yet).

Except the genetic attack (Klimov *et al*., 2002) mark geometric attack which is based on the principles of neural networks and probabilistic attack as tolerable ones. However both of them are theoretical insights and even have no formal algorithm.

This, in its turn, proves that currently neurocryptographic systems are the only postquantum code systems which may be realized and will be quite cryptosecure in order to withstand attacks even from the quantum computer.

CONCLUSION

This study had a task to find the postquantum encryption method which would allow to create a steady cryptosystem, at the same time quite simple for mass realization.

Various postquantum cryptographic systems were analyzed and their realization at the present time was estimated. As it was determined in the process of evaluation, neurocryptographic systems have the highest perspective among them.

With the help of existing developments the variant of a neurocryptographic protocol was shown which ensures sufficient cryptosecurity (as it was analyzed in the second chapter) and at the same time doesn't suppose complicated realization. This confirms the mentioned above hypothesis that neurocryptographic systems are exactly the most perspective among all postquantum encryption algorithms.

REFERENCES

Andreas, R., K. Wolfgang, N. Rivka and K. Ido, 2006. Genetic attack on neral cryptography. Phys. Rev. E, 73(3).

Bernstein, D., T. Lange and C. Peters, 2008. Attacking and defending the McEliece cryptosystem. Proceeding of the 2nd International Workshop on Post-Quantum Cryptography (PQCrypto'08). Belin, Cryptology ePrint Archive, Report 318, Springer-Verlag, Heidelberg, pp: 31-46.

Colin, B., 2009. First Universal Programmable Quantum Computer Unveiled, New Scientist. Retr ieved from: http://www.newscientist. com/ article/ dn18154-first-universal-programmable-quantum-computer-unveiled.html.

IBM Research, 2012. IBM Research Advances Device Performance for Quantum Computing. Retrieved from: http://www-03.ibm.com/press/us/en/ press release/36901.wss.

Khalil, S., 2012. A backpropagation neural network for computer network security. J. Comput. Sci., 2: 710-715.

Klimov, A., A. Mityagin and A. Shamir, 2002. Analysis of neural cryptography-advances in cryptology. Asiacrypt, 2501: 288-298.

Micciancio, D. and O. Regev, 2009. Lattice-based Cryptography, Post-Quantum Cryptography, pp: 147-191. Retrieved from: http://www.math.uni-bonn.de/~saxena/courses/WS2010-ref5.pdf.

Oscar, M.R. and Z. Karl-Heinz, 2010. Permutation parity machines for neural cryptography. Inst. Comput. Technol. Hamburg Univ., Technol., 81(2).

Pellegrini, A., V. Bertacco and T. Austin, 2010. Fault-based attack of RSA Authentification. Prceeding of the Conference on Design, Automation and Test in Europe, pp: 855-860.

Sébastien Dourlens Neuro-applied cryptography, 1995-1996. Computer Control Option Mcroinformatique Microelectronics, Year (French).

Shor, P., 1994. Algorithms for quantum computation: discrete logarithms and factoring. Proceeding of the 35th Anual Symposium on Foundations of Computer Science. Santa Fe, NM, pp: 124-134.

Tai-Wen, Y. and C. Suchen, 2001. The general neural-network paradigm for visual cryptography. Proceedings of the 6th International Work-Conference on Artificial and Natural Neural Networks: Connectionist Models of Neurons (IWANN '01), Learning Processes and Artificial Intelligence-Part I, 2084: 196-206.