CRYPTOGRAPHY BASED ON NEURAL NETWORK

Eva Volna
Martin Kotyrba
Vaclav Kocian
Michal Janosek
Department of Informatics and Computers
University of Ostrava
Dvorakova 7, Ostrava, 702 00, Czech Republic
eva.volna@osu.cz
martin.kotyrba@osu.cz
vaclav.kocian@osu.cz
michal.janosek@osu.cz

KEYWORDS
Cryptography key, encryption system, encryption algorithm, artificial neural network.

ABSTRACT

The goal of cryptography is to make it impossible to take a cipher and reproduce the original plain text without the corresponding key. With good cryptography, your messages are encrypted in such a way that brute force attacks against the algorithm or the key are all but impossible. Good cryptography gets its security by using incredibly long keys and using encryption algorithms that are resistant to other form attack. The neural net application represents a way of the next development in good cryptography. This paper deals with using neural network in cryptography, e.g. designing such neural network that would be practically used in the area of cryptography. This paper also includes an experimental demonstration.

INTRODUCTION TO CRYPTOGRAPHY

The cryptography deals with building such systems of security of news that secure any from reading of trespasser. Systems of data privacy are called the cipher systems. The file of rules are made for encryption of every news is called the cipher key. Encryption is a process, in which we transform the open text, e.g. message to cipher text according to rules. Cryptanalysis of the news is the inverse process, in which the receiver of the cipher transforms it to the original text. The cipher key must have several heavy attributes. The best one is the singularity of encryption and cryptanalysis. The open text is usually composed of international alphabet characters, digits and punctuation marks. The cipher text has the same composition as the open text. Very often we find only characters of international alphabet or only digits. The reason for it is the easier transport per media. The next cipher systems are the matter of the historical sequence: transposition ciphers, substitution ciphers, cipher tables and codes.
Simultaneously with secrecy of information the tendency for reading the cipher news without knowing the cipher key was evolved. Cipher keys were watched very closely. The main goal of cryptology is to guess the cipher news and to reconstruct the used keys with the help of good analysis of cipher news. It makes use of mathematical statistics, algebra, mathematical linguistics, etc., as well as known mistakes made by ciphers too. The legality of the open text and the applied cipher key are reflected in every cipher system. Improving the cipher key helps to decrease this legality. The safety of the cipher system lies in its immunity against the decipher.
The goal of cryptanalysis is to make it possible to take a cipher text and reproduce the original plain text without the corresponding key. Two major techniques used in encryption are symmetric and asymmetric encryption. In symmetric encryption, two parties share a single encryption-decryption key (Khaled, Noaman, Jalab 2005). The sender encrypts the original message (P), which is referred to as plain text, using a key (K) to generate apparently random nonsense, referred to as cipher text (C), i.e.:

$$\boldsymbol{C} = \textbf{Encrypt}\ (\boldsymbol{K},\boldsymbol{P}) \tag{1}$$

Once the cipher text is produced, it may be transmitted. Upon receipt, the cipher text can be transformed back to the original plain text by using a decryption algorithm and the same key that was used for encryption, which can be expressed as follows:

$$\boldsymbol{P} = \textbf{Dencrypt}\ (\boldsymbol{K},\boldsymbol{C}) \tag{2}$$

In asymmetric encryption, two keys are used, one key for encryption and another key for decryption.
The length of cryptographic key is almost always measured in bits. The more bits that a particular cryptographic algorithm allows in the key, the more keys are possible and the more secure the algorithm becomes. The following key size recommendations should be considered when reviewing protection (Ferguson, Schneier, Kohno, 2010):
Symmetric key:

- Key sizes of 128 bits (standard for SSL) are sufficient for most applications

Proceedings 26th European Conference on Modelling and Simulation ©ECMS Klaus G. Troitzsch, Michael Möhring, Ulf Lotzmann (Editors)
ISBN: 978-0-9564944-4-3 / ISBN: 978-0-9564944-5-0 (CD)

- Consider 168 or 256 bits for secure systems such as large financial transactions

Asymmetric key:

- Key sizes of 1280 bits are sufficient for most personal applications
- 1536 bits should be acceptable today for most secure applications
- 2048 bits should be considered for highly protected applications.

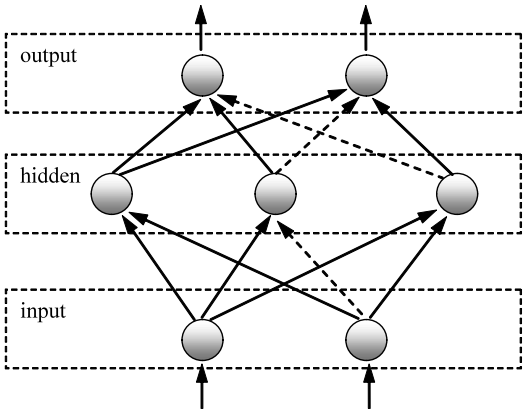
Hashes:

- Hash sizes of 128 bits (standard for SSL) are sufficient for most applications
- Consider 168 or 256 bits for secure systems, as many hash functions are currently being revised (see above).

NIST and other standards bodies will provide up to date guidance on suggested key sizes.

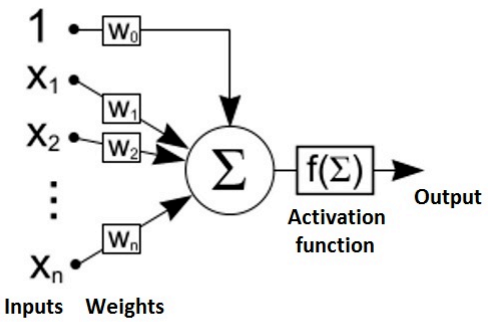
BACKPROPAGATION NEURAL NETWORKS

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well.



Figures 1: A general three layer neural network

Backpropagation network is one of the most complex neural networks for supervised learning. Regarding topology, the network belongs to a multilayer feedforward neural network. See Fig. 1 (Volna 2000), usually a fully connected variant is used, so that each neuron from the *n-th* layer is connected to all neurons in the *(n+1)-th* layer, but it is not necessary and in general some connections may be missing – see dashed lines, however, there are no connections between neurons of the same layer. A subset of input units has no input connections from other units; their states are fixed by the problem. Another subset of units is designated as output units; their states are considered the result of the computation. Units that are neither input nor output are known as hidden units.



Figures 2: A simple artificial neuron (http://encefalus.com/neurology-biology/neural-networks-real-neurons)

A basic computational element is often called a neuron (Fig. 2), node or unit (Fausett 1994). It receives input from some other units, or perhaps from an external source. Each input has an associated weight ***w***, which can be modified so as to model synaptic learning. The unit computes some function *f* of the weighted sum of its inputs (3):

$$y_i = f\left(\sum_j w_{ij} x_j\right) \tag{3}$$

Its output, in turn, can serve as input to other units. The weighted sum is called the net input to unit *i*. Note that w_{ij} refers to the weight from unit *j* to unit *i* (not the other way around). The function *f* is the unit's activation function. Backpropagation algorithm usually uses a logistic sigmoid activation function (4) for values of *t* in the range of real numbers from $-\infty$ to $+\infty$.

$$f(t) = \frac{1}{1+e^{-t}} \tag{4}$$

Backpropagation algorithm belongs to a group called "gradient descent methods". An intuitive definition is that such an algorithm searches for the global minimum of the weight landscape by descending downhill in the most precipitous direction. The initial position is set at random selecting the weights of the network from some range (typically from -1 to 1 or from 0 to 1). Considering the different points, it is clear, that backpropagation using a fully connected neural network is not a deterministic algorithm. The basic backpropagation algorithm can be summed up in the following equation (the *delta rule*) for the change to the weight w_{ji} from node *i* to node *j* (5):

$$\begin{array}{lcccccc} \text{weight} & & \text{learning} & & \text{local} & & \text{input signal} \\ \text{change} & & \text{rate} & & \text{gradient} & & \text{to node } j \\ \Delta w_{j\,i} & = & \eta & \times & \delta_j & \times & y_i \end{array} \qquad (5)$$

where the local gradient δ_j is defined as follows (Seung 2002):

1. If node j is an output node, then δ_j is the product of $\varphi'(v_j)$ and the error signal e_j, where $\varphi(\_)$ is the logistic function and v_j is the total input to node j (i.e. $\Sigma_i w_{ji} y_i$), and e_j is the error signal for node j (i.e. the difference between the desired output and the actual output);
2. If node j is a hidden node, then δ_j is the product of $\varphi'(v_j)$ and the weighted sum of the δ's computed for the nodes in the next hidden or output layer that are connected to node j.

The actual formula is $\delta_j = \varphi'(v_j)\&\text{Sigma}_k\, \delta_k w_{kj}$ where k ranges over those nodes for which w_{kj} is non-zero (i.e. nodes k that actually have connections from node j. The δ_k values have already been computed as they are in the output layer (or a layer closer to the output layer than node j).

NEURAL CRYPTOGRAPHY

Neural cryptography (Kanter and Kinzel 2002, Kinzel 2002) is based on the effect that two neural networks are able to synchronize by mutual learning (Ruttor *et al*. 2006). In each step of this online procedure they receive a common input pattern and calculate their output. Then, both neural networks use those outputs present by their partner to adjust their own weights. This process leads to fully synchronized weight vectors.

Synchronization of neural networks is, in fact, a complex dynamical process. The weights of the networks perform random walks, which are driven by a competition of attractive and repulsive stochastic forces. Two neural networks can increase the attractive effect of their moves by cooperating with each other. But, a third network which is only trained by the other two clearly has a disadvantage, because it cannot skip some repulsive steps. Therefore, bidirectional synchronization is much faster than unidirectional learning (Ruttor 2004).

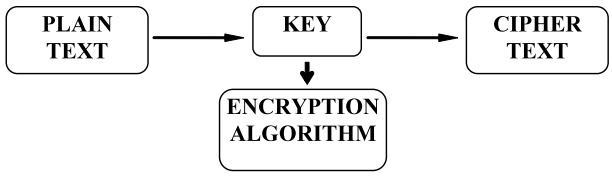
Two partners *A* and *B* want to exchange a secret message over a public channel. In order to protect the content against an attacker *T*, who is listening to the communication, *A* encrypts the message, but *B* needs *A*'s secret key over the public channel (Kinzel 2002). This can be achieved by synchronizing two TPMs (Three Parity Machines), one for *A* and one for *B*, respectively. After synchronization, the system generates a pseudorandom bit sequence which passes test on random numbers. When another network is trained on this bit sequence it is not possible to extract some information on the statistical properties of the sequence. The TPMs generate a secret key and also encrypt and decrypt a secret message (Prabakaran, Loganathan and Vivekanandan 2008).

In this paper we present an encryption system based on an Artificial Neural Network (ANN). ANN is used to construct an efficient encryption system by using a permanently changing key. The ANN topology is an important issue, as it depends on the application the system is designed for. Consequently, since our application is a computation problem, we have used a multi-layer topology. In the present paper, Backpropagation network is proposed for the encryption-and-decryption process. Neural networks offer a very powerful and general framework for representing non-linear mapping from several input variables to several output variables. The process to determining the values of these parameters on the basis of a data set is referred to as learning or training, and so the data set is generally referred to as a training set. A neural network can be viewed as suitable choice for the functional forms used for encryption and decryption operations.

DESIGN OF THE PROPOSED ANN-BASED ENCRYPTION SYSTEM

Every practical encryption system consists of four fundamental parts (Garfinger 1998), see Figure 3:

- The message that you wish to encrypt (called the *plain text*).
- The message after it is encrypted (called the *cipher text*).
- The encryption algorithm.
- The key (or keys), which is used by encryption algorithm.



Figures 3: A simple example of the encryption system

In this paper, we conducted an experimental study with using neural network in cryptography. Thus, it means

- to design the topology of the neural network;
- to design the method of training algorithm of the neural network;
- to design the training set for training.

We successfully used neural networks as an encryption and decryption algorithm in cryptography. Parameters of both adapted neural networks were then included into cryptography keys. We used multilayer neural networks, which were adapted by backpropagation. Topology of each neural network is based on their training sets (see Table 1). In the encryption process, the input message is divided into 6-bit data sets and also 6-bit sets are produced after the encryption process. Thus, both systems were designed as follows: 6 units on the input

layer and 6 output units. There is no predetermined number of units in the hidden layer, but we also used 6 units. Both networks were trained on binary representations of symbols. In each training set, chains of numbers of the plain text are equivalent to binary values of their ASCII code, chains of letters of the plain text are equivalent to their binary value, which are 96 less than their ASCII code, each chain of some punctuation symbol of the plain text is equivalent to a binary value of ASCII code of space (e.g. 32), and chains of others chars of the plain text are equivalent to zero. Then, the cipher text is a random chain of 6 bits.
The security for all encryption and decryption systems is based on a cryptographic key. The SIMPLE systems use a single key for both encryption and decryption. The good systems use two keys. A message encrypted with one key can be decrypted only with the other key. If we use the neural network as encryption and also decryption algorithm, their keys have adapted neural networks´ parameters; which are their topologies (architecture) and their configurations (weight values on connections in the given order). Generally, each key is written as follow:

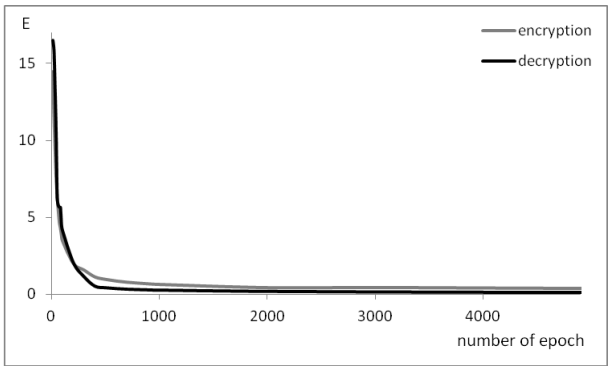
[*Input, Hidden, Output, Weights coming from the input units, Weights coming from the hidden units*]

where
Input is the number of input units;
Hidden is the number of hidden units;
Output is the number of output units;
Weights coming from the input units are weight values coming from the input units to hidden units in a predefined order;
Weights coming from the hidden units are weight values coming from the hidden units to output units in a predefined order

Parameter values of both ANNs in our experimental study are the following:

- each input layer consists of 6 nodes, which represents the 6-bit blocks;
- each hidden layer consists of 6 nodes;
- each output layer consists of 6 nodes, used to define the decrypted output message;
- fully connected networks;
- a sigmoid activate function;
- a learning rate equals 0,3.

History of both Error functions (E) is shown in Fig. 4. There are shown average values of error function, because adaptation with backpropagation algorithm was applied 10 times in each calculation. Other numerical simulations give very similar results.



Figures 4: The Error function history

SENDING AND RECEIVING MESSAGES

In this model, a 6-bit plain text is entered ($N = 6$) and a 6-bit cipher text is the output (2^N).

Imagine that we want to send the following message:
"*We are in zoo, call us.*"

The first steps of our encryption process are to convert all uppercase letters to lowercase and to replace all punctuation symbols by a space. After this process, our message is the following:
"*we are in zoo call us*"

Then we replace all two spaces by one space, we get the following plain text:
"*we are in zoo call us*"

The plain text is coded into the chain:
(010111000101100000000001010010000101100000001001001110100000011010001111001111100000000010000001001100001100100000010101010011)

Now, we break it down into blocks (N =6), thus:
010111 000101 100000 000001 010010 000101 100000 001001 001110 100000 011010 001111 001111 100000 000010 000001 001100 001100 100000 010101 010011

The corresponding cipher text is the following:
(100100100000101111000010010111100000101111001000011100101111001101101000101000101111001011000010011010111010111101010010100111)

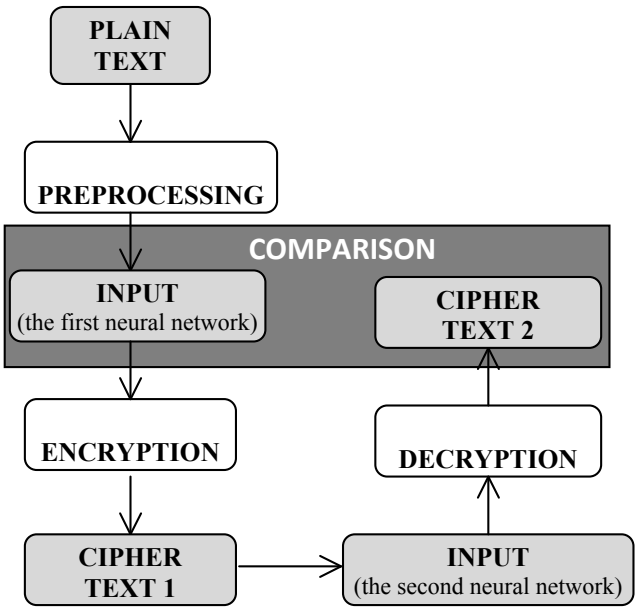
The encrypted data will then be transmitted to the recipient.

RESULTS AND DISCUSSION

We have tested the behavior of the neural network described in the previous section so that we have generated messages (plain text) that were encrypted via the first adapted neural network. Then we have received some cipher text, which represented some input into the decryption process carried out via the second adapted neural network. Each obtained cipher text was compared with the original message after its pre-

processing. The whole procedure is demonstrated in Fig. 5. We found that:

- the neural network works reliably and absolutely no errors are found in the outputs during encryption;
- the neural network also works reliably during the decryption process, which is the reverse of the encryption process.



Figures 5: Tested process of a behavior of neural networks

This model presents an attempt to design an encryption system based on artificial neural networks of the backpropagation type. The proposed ANN has been tested for various numbers of plain text. The simulation results have shown very good results.

Table 1: Table of experiment constants

THE PLAIN TEXT			THE CIPHER TEXT
Char	ASCII code (DEC)	The chain of bits	The chain of bits
0	48	110000	111111
1	49	110001	110010
2	50	110010	101100
3	51	110011	111010
4	52	110100	101010
5	53	110101	100011
6	54	110110	111000
7	55	110111	000111
8	56	111000	010101
9	57	111001	110011
punct.	32	100000	101111
others	0	000000	011101
a	97	000001	000010
b	98	000010	100110
c	99	000011	001011
d	100	000100	011010
e	101	000101	100000
f	102	000110	001110
g	103	000111	100101
h	104	001000	010010
i	105	001001	001000
j	106	001010	011110
k	107	001011	001001
l	108	001100	010110
m	109	001101	011000
n	110	001110	011100
o	111	001111	101000
p	112	010000	001010
q	113	010001	010011
r	114	010010	010111
s	115	010011	100111
t	116	010100	001111
u	117	010101	010100
v	118	010110	001100
w	119	010111	100100
x	120	011000	011011
y	121	011001	010001
z	122	011010	001101

CONCLUSION

The neural net application represents a way of the next development in good cryptography, but we can ask question. What are the limitations of the system? The limitations of this type of system are few, but potentially significant. This is effectively a secret-key system, with the key being the weights and architecture of the network. With the weights and the architecture, breaking the encryption becomes trivial. However, both the weights and the architecture are needed for encryption and decryption. Knowing only one or the other is not enough to break it. What are the advantages to this system? The advantages to this system are that it appears to be exceedingly difficult to break without knowledge of the methodology behind it, as shown above. In addition, it is tolerant to noise. Most messages cannot be altered by even one bit in a standard encryption scheme. The system based on neural networks allows the encoded message to fluctuate and still be accurate.

ACKNOWLEDGEMENT

The research described here has been financially supported by University of Ostrava grant SGS2/PřF/2012. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

Fausett, L.V. 1994 *Fundamentals of Neural Networks*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey

Ferguson, N., Schneier, B., Kohno, T. 2010 *Cryptography Engineering: Design Principles and Practical Applications.* Wiley Publishing ISBN:0470474246 9780470474242

Garfinger, S. 1998. *PGP: Pretty Good Privanci*. Computer Press, Praha.

Kanter, I., Kinzel, W., 2002. Neural cryptography. In: *Proceedings of the 9th International conference on Neural Information Processing*. Singapore.

Khaled, M. Noaman, G., Jalab, H.A. 2005. Data security based On neural networks. *TASK Quarterly* 9 No 4, pp. 409–414

Kinzel, W., 2002 Theory of Interacting Neural Network. Preprint [cont.-mat/020454].

Prabakaran, N., Loganathan, P. and Vivekanandan, P. 2008. Neural Cryptography with Multiple Transfers Functions and Multiple Learning Rule. *International Journal of Soft Computing, 3: 177-181.*

Ruttor, A., Reents, G., Kinzel, W. 2004. Synchronization of random walk with reflecting boundaries. J. Phys. A: Math.Gen, 37: 8609 [cont-mat/0405369].

Seung, S. 2002. Multilayer perceptrons and backpropagation learning. 9.641 Lecture 4. 1-6. Available from: http://hebb.mit.edu/courses/9.641/2002/lectures/lecture04.pdf

Volná, E. 2000. Using Neural network in cryptography. In P. Sinčák, J. Vaščák, V. Kvasnička, R. Mesiar (eds.): *The State of the Art in Computational Intelligence.* Physica-Verlag Heidelberg. pp.262-267. ISBN 3-7908-1322-2, ISSN 1615-3871.

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

AUTHOR BIOGRAPHIES

EVA VOLNA is an Associative Professor at the Department of Computer Science at University of Ostrava, Czech Republic. Her interests include artificial intelligence, artificial neural networks, evolutionary algorithms, and cognitive science. She is an author of more than 50 papers in technical journals and proceedings of conferences.

MARTIN KOTYRBA is a Ph.D. student at the Department of Computer Science at University of Ostrava, Czech Republic. His interests include artificial intelligence, formal logic, soft computing methods and fractals. He is an author of more than 15 papers in proceedings of conferences.

VACLAV KOCIAN is a Ph.D. student at the Department of Computer Science at University of Ostrava, Czech Republic. His interests include artificial intelligence, artificial neural networks, and soft computing methods. He is an author of more than 10 papers in proceedings of conferences.

MICHAL JANOŠEK is a Ph.D. student at the Department of Computer Science at University of Ostrava, Czech Republic. His interests include artificial intelligence, multi-agent systems, modeling and simulations. He is an author of more than 10 papers in proceedings of conferences.