Neural Cryptography in Artificial Intelligence

KEYWORDS: coding and encryption; artificial intelligence; multiple algorithms; personalised encryption engines; artificial neural networks; evolutionary computing

Twinkle Pardeshi
Maharashtra Institute Of Technology, Pune University, Pune

Apoorva Patil
Maharashtra Institute Of Technology, Pune University, Pune

Shreyal Gajare
Maharashtra Institute Of Technology, Pune University, Pune

ABSTRACT A method of generating encryption algorithms using neural networks and evolutionary computing is described in this paper. Based on the application of natural noise sources obtained from data that can include atmospheric noise , radioactive decay, electronic noise and so on, we 'teach' a system to approximate the input noise with the aim of generating an output nonlinear function. This output is then treated as an iterator which is subjected to a range of tests to check for potential cryptographic strength in terms of metric such as a (relatively) large positive. Lyapunov exponent, high information entropy, a high cycle length and key diffusion characteristics, for example. This approach provides the potential for generating an unlimited number of unique Pseudo Random Number Generator (PRNG) that can be used on a I-to-l basis. Typical applications include the encryption of data before it is uploaded onto the Cloud by a user that is provided with a personalized encryption algorithm rather than just a personal key using a 'known algorithm' that may be subject to a 'known algorithm attack' and/or is 'open' to the very authorities who are promoting its use.

I. INTRODUCTION

Protective software is as an integral part of an operating system, e.g. file managers, browsers, text processors etc. However, even with use of the latest protective software products there is no guarantee that the computer and/or computing environment is 'safe'. Such infamous mal ware for cyber espionage as Stuxnet, Flamer, Duqu, Gauss, Regin and others gives proof that up-to-date protective solutions are powerless in the face of sophisticated cyber-weapons. The listed malware may not only still undetectable for years but also perform cyber espionage by controlling infected computers. The existing situation opens a possible arena in regard to personalized encryption algorithms and data protection systems. Personal encryption algorithms allow users to protect data even if protective software fails to detect a threat.

II. ST ATE-OF-THE-ART

Neural cryptography is a direction in cryptography that applies Artificial Neural Networks (ANN) for encryption and Crypt analysis. Recent works in the use of neural networks in cryptography can be categorized into three major parts:

a) Implementation of synchronized neural networks.
b) Cryptography based on use of chaotic neural networks.
c) Cryptography based on multi-layer neural networks.

A. Synchronization neural networks:

The first approach of applying an ANN in cryptography is through the creation of two topologically identical neural networks. Neural networks can synchronize by learning from each other and full synchronization can be achieved in a finite number of steps. The main idea is as follows: the user A wants to communicate with user B, but they cannot exchange a secret key through a secure channel; so they create two topologically identical neural networks but with different random weights and evaluate them with the same inputs until the weights of both ANN's match [3] - [5]. The main algorithm of such a system consists of the following steps:

1) initialize random weight values.
2) Execute the following steps until full synchronization is achieved.

a) Generate a random input vector.
b) Compute the values of the hidden and output neurons.
c) Compare the values of both neural networks. If the outputs are different then go to 2(a) else update the weights of the networks according to learning rules.

3) After full synchronization is achievd, A and B can use their weights as keys.

The process of synchronization of the networks can be considered as the key generation process in cryptography. The common identical weights of synchronized networks for two partners can be used as a key for encryption.

B. Chaos-based neural cryptography:

Chaos-based cryptography combines two research fields, i.e., chaos (nonlinear dynamic systems) and cryptography and becomes more practical in the secure transmission of large multi-media files over public data communication networks. In this field, the implementation of neural networks composed of chaotic neurons opens a wide area for developing strong cryptosystems. For example, W. Yu proposed an encryption technique based on the chaotic Hopfield neural network with time varying delay [6]. The proposed chaotic neural network is used for generating binary sequences for masking the plaintext.

The binary value of the binary sequence chooses the logistic map randomly, used for generated the binary sequences. The plaintext is masked by switching the chaotic neural network maps and the permutation of generated binary sequences. In the area of image cryptography, A Triple Key means that three parameters are used as control parameters producing a hexadecimal sequence. The triple parameters are used to perform the various operations on an image so as to scramble the data.

C. Multilayer neural networks in the cryptography:

Here a neural network is used to construct an efficient encryption system by using a permanently changing key. The encryption function consists of the three steps:

1) the creation of the keys;

2) the input message is broken into blocks equal to the number of keys and processed, one block at a time, as input to the neural network;

3) the neural network is composed of three layers (each layer consisting of a number of neurons, depending on the process type) and is used to encrypt each block of data producing the encrypted message. The encryption process divides the input message into 3 - bit data sets, and produces 8- bit data after the encryption process. The simulation results provide a relatively better performance than traditional encryption methods.

III. EVOLUTIONARY COMPUTATION IN CRYPTOGRAPHY

Evolutionary Computing is associated with the fIeld of Computational Intelligence, and, like ArtifIcial Intelligence, involves the process of continuous and combinatorial optimizations. It is inspired from biological mechanisms of evolution and uses iterative processes in a parallel manner to achieve the goal. Evolutionary algorithms are a part of Evolutionary Computation and simulate different biological mechanisms such as reproduction, mutation, recombination, natural selection and so on. For designing Pseudo Random Number Generators (pRNGs), an evolutionary algorithms based approach can be used in which a population-based, stochastic search engine is required that mimics natural selection. Due to their ability to fmd excellent solutions for conventionally difficult and dynamic problems within acceptable time, evolutionary algorithms have attracted interest from many areas of science and engineering.

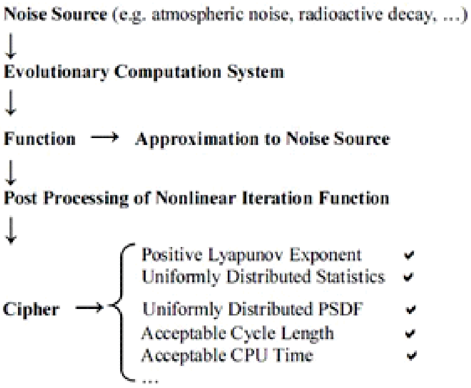

Noise Source (e.g. atmospheric noise, radioactive decay, ...)
Evolutionary Computation System
Function
Approximation to Noise Source
Post Processing of Nonlinear Iteration Function
Cipher
Positive Lyapunov Exponent ✓
Uniformly Distributed Statistics ✓
Uniformly Distributed PSDF ✓
Acceptable Cycle Length ✓
Acceptable CPU Time ✓
...

Fig. 1. Schematic of the processes for evolving a cipher.

Evolutionary computing systems, which use mechanisms such as reproduction, mutation, recombination, selection etc., try to find the best equation for an optimal approximation to a given data stream. We use the Eureqa system developed by

Nutonian Inc. for detecting equations and hidden mathematical relationships in the data. This system uses evolutionary computations. For this study, we use the data available from RANDOM. ORG which, to date, has generated 1.92 trillion random bits for the Internet community [12]. Fig. 2 shows an example screen shot of the Eureqa system used to generate the following iteration function for cipher generation.

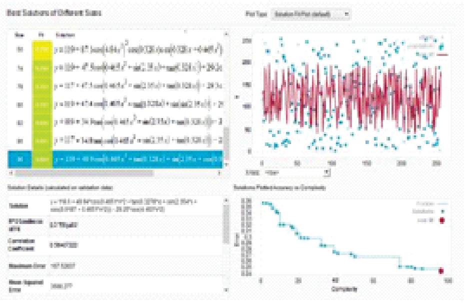


Fig. 2. Screen shot of *Eureqa* used for evolving nonlinear functions suitable for cipher generation.

The non-linear iteration function given by equation (Fig. 3 demonstrates some cryptographic strength characteristics, e.g. uniform distribution, power spectrum and autocorrelation function) is the result of Eureqa undertaking over 100 iterations (using 255 noise samples randomly selected from the data bases available at RANDOM. ORG) to evolve the result, taking approximately 27 hours using an Intel Core i3 1.7x2 GHz CPU to do so.

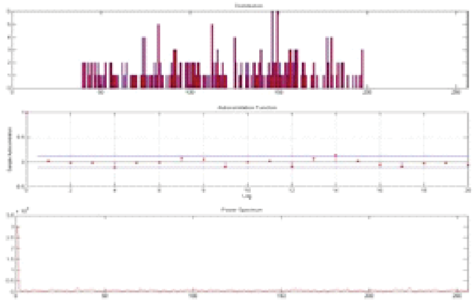


Fig. 3. Cryptographic strength of the iterated function.

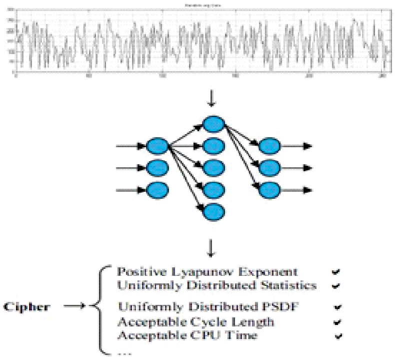
In this way, the proposed approach pays no attention to the algorithmic complexity of the iterator which is one of the main problems in the application of chaos to cryptography. Neither does it consider the structural stability of the iterator or its algorithmic complexity.

IV. NEURAL CRYPTOGRAPHY

Artificial Neural Networks (ANN) as well as Evolutionary Computation are a subfield of Artificial Intelligence and involve various architectures of neural networks and rules of training. ANNs were inspired by biological neural networks (the central nervous systems, in particular, the brain) and can be implemented in various complex control engineering problems such as function approximation pattern recognition, data mining, prediction, machine learning and so on. Generally, ANNs consist of several layers of interconnected nodes or "neurons" that form a network that mimics a biological neural network. Through iterative training process an ANN computes a set of optimal weights between neurons that determines the flow of information (the amplitude of a signal at a given node) through a network that simulates a simple output subject to a complex input.

In this paper we propose different technique that is based on ability of an ANN to simulate a high entropy input with the aim of transforming the result into a low entropy output.

This demonstrates that an ANN can produce highly nonlinear behavior. Given this statement, the precise ANN algorithm becomes analogous to a PRNG in conventional cryptography and the weights are equivalent to the key.


Cipher
Positive Lyapunov Exponent ✓
Uniformly Distributed Statistics ✓
Uniformly Distributed PSDF ✓
Acceptable Cycle Length ✓
Acceptable CPU Time ✓
...

In comparison with the Evolutionary Commutating approach, described in the Section II, an ANN provides analogous results (uniformly distributed statistics, uniformly distributed power spectrum, positive Lyapunov exponent, information entropy etc.).

V. CONCLUSIONS

Practical cryptography is based on passing known statistical tests pseudo-random sequences being taken to be used instead of truly random sequences in most cryptographic applications. This paper introduces a way of designing algorithms for generating pseudo-random (chaotic) sequences using truly random strings to evolve an iterator that is taken to be an approximation to these sequences. This approach pays no attention to the algorithmic complexity of the iterator which is one of the main problems in the application of chaos to cryptography. Neither does it consider the structural stability of the iterator or its algorithmic complexity. However, it does provide a practical solution to the problem of developing a large database of PRNGs for the application of personalizing encryption algorithms for strictly 'one-to-one' communications or 'one-to-Cloud' (encrypted) data storage. However, the one-step unpredictability does not guarantee that the output sequence will be unpredictable when an adversary has access to a sufficiently long sequence.

REFERENCES

1. Blackledge, S. Bezobrazov, P. Tobin and F. Zamora, "Cryptography using Evolutionary Computing", lET ISSCI3, L YIT Letterkenny, 2013.
2. Eureqa, "A software tool for detecting equations and hidden mathematical relationships in your data", Cornell Creative Machine Labs, USA,20\3,
3. W. Kinzel and I. Kanter, "Neural Cryptography," TH2002 Supplement, vol. 4, pp. 147-153, 2003.
4. E. Klein, R. Mislovaty, I. Kantor, A. Ruttor and W. Kinzel, "Synchonization of neural networks by mutual learning and its aoolication to cryptography", Adwances in Neural Information Processing Systems, NIPS, 2004.
5. R. Jogdand and S. Bisalapur, "Design of an efficient neural key generation", International Journal of Artificial Intelligence and Applications (IJAIA), vol. 2, no. 1,2011, pp. 60-69.
6. W. Yu and 1. Cao, "Cryptography based on delayed chaotic neural networks", Physics Letters A, Vol. 356, (4) Elsevier, pp. 333-338, 2006.
7. A. EI-Zoghabi, A. Yassin and H. Hussien, "Survey Report on Cryptography based on Neural Network" International Journal of Emerging Technology and Advanced Engineering, (rSSN 2250-2459), Vol. 3, Issue 12, December 2013.
8. S. Suryawanshi and D. Nawgaje, "A triple-key Chaotic neural network for cryptography in image processing", International Journal of Engineering Sciences & Emerging Technologies, Vol. 2, Issue. 1, pp. 46-50,2012.