Computers and Electrical Engineering 72 (2018) 179–190

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Design and investigation of a chaotic neural network architecture for cryptographic applications☆

A. Ruhan Bevi, Sriharini Tumu*, N. Varsha Prasad

Department of ECE, SRM Institute of Science and Technology, Kattankulathur 603203, India

ARTICLE INFO

Article history:
Received 30 August 2017
Revised 13 September 2018
Accepted 13 September 2018

Keywords:
Chaotic neural network
Memristor
Nonlinear equations
Cubic map
2D Logistic map
Encryption

ABSTRACT

Artificial neural networks are an integral part of emerging technologies, and ongoing research has shown that they can be applied to a variety of applications. This paper proposes a new cryptographic algorithm using chaotic neural networks, whose function is enhanced by construction with polynomials that exhibit chaos, namely, nonlinear Hermite and Chebyshev polynomials. These polynomials incorporate a memristor conductance, which is used as an activation function in the chaotic neural networks. Further, a function of the weights obtained from the chaotic neural networks, is used to generate the initial values that are used in the cryptographic process. The encryption algorithm employed here is inspired by the Lai–Massey block cipher with cubic and two-dimensional logistic maps, and the evaluation of these chaotic equations is performed using correlation values. The correlation values between the cipher and plain text are also examined to determine the undecipherability of the message to be sent on a public channel.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

There is a growing need for secure transmission of information, to ensure confidentiality between the sender and recipients. Cryptography deals with the study of developing and analyzing processes that improve the level of security and protect the information from reaching the hands of adversaries. The construction of a chaotic neural network (CNN) with the help of a memristor conductance equation to enhance the performance of the network has been inspired by Wang et al. [1].

A memristor is a nonlinear, passive device having two terminals, which exhibits nonvolatile properties as its electrical resistance is dependent on the previous current values. The physical model of a memristor was introduced by a team at HP laboratories, 37 years after the theoretical proposal was made by Chua in 1971 [2,3]. This model consists of two layers with platinum contacts and a thin film of TiO_2, placed between them. One of the layers acts as a semiconductor, due to the doping of oxygen vacancies and the other exhibits insulating properties. The width of the doped region can be determined by the electrical charge passing through the memristor. There is a drift in the oxygen vacancies due to an external excitation, causing the boundary between the regions to shift in the same direction. Memristors have been used in a variety of applications since its model came into existence and some of its potential applications include simulations that can recall patterns,

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. G. G. Devadhas.

* Corresponding author.
E-mail addresses: ruhan.b@ktr.srmuniv.ac.in (A.R. Bevi), sriharini12@gmail.com (S. Tumu).

https://doi.org/10.1016/j.compeleceng.2018.09.015
0045-7906/© 2018 Elsevier Ltd. All rights reserved.

in a manner analogous to the human brain. The use of memristor conductance and chaotic polynomials for synaptic weight updation contributes to faster convergence and constitutes an integral part of the system.

The use of memristors for data encryption, has interested scientists around the world. Du et al. [4] demonstrated the use of nonlinear resistance change in $BiFeO_3$ memristors for the generation of higher harmonics to develop a novel encoding system. Further, Thomas [5] explained the use of memristors in neural networks due to their unique 'memory' property, which can be used to imitate the synapse as in biological neural networks. The nonvolatile nature of memristors can also be exploited to develop physical unclonable functions (PUF), which provide higher security by preventing the extraction of secret keys [6]. Shi et al. [7], proposed the use of memristors in an artificial neural network (ANN) architecture by the synaptic weight updation process using Hermite polynomials as the activation functions.

In this work, a neural network is constructed, using the chaotic properties of nonlinear equations like Hermite and Chebyshev polynomials and theoretical equations concerned with the fourth circuit element, namely memristor, which is used to catalyze the process of convergence. The encryption and decryption process that is proposed is a quasi-Feistel cipher. This algorithm makes use of the chaotic maps that produce a sequence, using which, the plaintext is encrypted. The use of the 1D cubic map as well as 2D map (two dimensional logistic) is explained using the given process. The faster convergence rate reduces the overall time taken to complete the process. These weights, which are obtained from the neural network once it converges to the output with zero error, can neither be computed manually, nor predicted without knowing the architecture of the CNN and the initial value, thereby making the process secure.

This paper is organized as follows: In Section 2, the overall architecture of the proposed CNN, with the key generation scheme targeted for cryptographic applications, is explained. Sections 3, 4, and 5 explain the developmental stages involved in the encryption process using the CNN. The results and discussion on the performance of the proposed system are addressed in Section 6 with the conclusions detailed in Section 7.

2. Architecture of the proposed CNN

The process flow of the CNN architecture for cryptographic application is shown in Fig. 1. The work is organized in three stages that together constitute the entire process from the processing of the key till the generation of the ciphertext. The input consists of the plaintext and the key value that has to be transmitted across the secure communication channel. The input key A is of any value within the boundary (0,1) and the plaintext P is the sequence of characters that has to be encrypted. The architecture of the proposed network targeted for encryption comprises three stages as shown in Fig. 1.

- Stage I: Learning of CNN
- Stage II: Chaotic series generation
- Stage III: Cryptographic process

Stage I of the CNN is a learning stage wherein the memristive conductance equation aids in the synaptic weight updation. It is made up of two functional units, namely, the memristive conductance equation for synaptic weight updation and the polynomials as the complement simulator of the error function integral. The derivation of the memristor conductance and the usage of polynomials capable of exhibiting polynomial chaos form a part of the key function of stage I. These generated weights are taken note of, and in stage II, they are used as chaotic initial values in the series generator containing the chaotic equations. The generated series are used as components in the encryption algorithm, which is a part of stage III.

3. Stage I: Learning of CNN

Stage I is the learning stage of the proposed CNN. The input key A is of any value within the boundary (0,1) and the plaintext P is the sequence of characters for the input. The integral of the error in each epoch is an integral of the chaotic polynomial function of the error, which in turn is a function of the input variable x. This integral of the error is fed back into the weight updation (w_i) equation by which the weights are updated.

The Fig. 2 depicts the network made up of three layers, namely, an input layer x having a single neuron, a hidden layer (h_1, h_2...h_{10}) having 10 neurons and an output layer y having one neuron. The 10 weights between the hidden and output layers are represented as w_i. The number of neurons chosen for the hidden layer is 10, because the number of keys generated from this architecture with 10 weights is associated with 10 iterations in the encryption process. The equation corresponding to the change in weight Δw is derived in the following section, wherein the calculated change in weight is added and supplied to the input for the next epoch. Further, the weights between the input and hidden layers are 1, whereas the weights between the hidden and output layers are calculated from the chaotic polynomials under consideration. Once the 10 weights from the input to the hidden layer and the 10 weights from the hidden to the output layer are established, the network converges to the output and is ready to be used in the encryption process.

Memristor conductance. The memristor conductance equation for synaptic weight updation is derived from the relationship between the voltage $v(t)$ and current $i(t)$ represented by the following equation.

$$v(t) = \left(R_{on}\left(\frac{w(t)}{D}\right) + R_{off}\left(1 - \frac{w(t)}{D}\right)\right) i(t) \tag{1}$$

Download English Version:

https://daneshyari.com/en/article/11030096

Download Persian Version:

https://daneshyari.com/article/11030096

Daneshyari.com