Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 3, pp. 733–751
DOI : 10.47974/JDMSC-1867

Neuronal crypto system based on chaotic super-increasing sequence

Assia Merzoug [†]
Loboratory of Coding and Security of Information
University Center of Barika
Batna 05001
Algeria

Hana Ali-Pacha *
Quartz Laboratory
École Catholique d'Arts et Métiers - École des Productions et Méthodes Industrielles
13 Bd de l'Hautil
95092 Cergy Pontoise Cedex
France

Adda Ali-Pacha [§]
Loboratory of Coding and Security of Information
University of Sciences and Technology of Oran
Mohamed Boudiaf
P. O. Box 1505
El M'Naouer Oran 31000
Algeria

Özen Özer [‡]
Department of Mathematics
Faculty of Science and Arts
Kırklareli University
Kirklareli 39100
Türkiye

Abstract

Neural networks were originally developed as an alternative to artificial intelligence to solve problems such as control, shape or word recognition, decision-making, memory, and more. In this work, we propose to assign them to secure images. For this, we use data (which is defined from the logistics map) to generate a super-increasing sequence which can

[†] *E-mail:* assia.merzoug@cu-barika.dz
[*] *E-mail:* h.ali-pacha@ecam-epmi.com (Corresponding Author)
[§] *E-mail:* adda.alipacha@univ-usto.dz
[‡] *E-mail:* ozenozer@klu.edu.tr

attributed as a weight of the synapses. Also, we have inspired to encrypt the images by the Merkel-Hellman algorithm. The proposed system is called Neuronal Crypto-System which is a secret key system. It takes some steps of encryption and decryption of a specific public key crypto system and crypts a grayscale image to color image (RGB).

Subject Classification (2010): *11T71, 94A60, 97F60, 68P25, 92B20, 68T05, 90B06.*

Keywords: *Cryptography, Knapsack problem, Merkele-hellman Cipher, Neural network, Logistics map.*

Introduction

In [1,2], authors defined the Cryptography as follows:

"Cryptography is the set of locking processes designed to protect access to certain data in order to make it inapprehensible to unauthorized people [1, 2]. So, its role is to assure the confidentiality and the integrity of this information as well as their accountability. Modern encryption algorithms use a key to secure the data. This is the most secret and important information exchanged between the sender and the recipient. There are two classes of cryptography systems based on key as follows:

1. A Symmetric-key cryptography uses only one key for both decryption of cipher text and encryption of plain text. This key is called as Private key.
2. In asymmetric encryption, two different keys are used for encryption and decryption. The key used for encryption is called the public key and the key used for decryption is called the private key.

If we give an example for Asymmetric (public) cryptography, we can say Merkle-Hellman crypto system is based on the Knapsack problem given in [3]:

Imagine a collection of pebbles of known weight $\{a_1, a_2, ..., a_n\}$. Assume that we place some of them in a Knapsack and weigh all.

"Is it possible to know which pebbles are in the bag even we know total weight?"

Artificial neural networks [4], on the other hand, are computer models of automata networks. Their structure and behaviour are "copied" from that of real neurons. They can do things like shape recognition, data re-ordering and, more interestingly, learning like the brain.

They are found in:

- **Aerospace:** automatic piloting, flight simulation.
- **Automotive:** automatic guidance system ...
- **Defense:** missile guidance, target tracking, face recognition, radar, sonar, lidar, signal processing, data compression, noise suppression ...
- **Electronics:** prediction of the sequence of a code, machine vision, speech synthesizer, nonlinear model.

- **Finance:** Cost of living forecast.
- **Medical Sector:** EEC and ECG Analysis.
- **Telecommunications:** Data compression.

In cryptography, workings of several encryption based on the Neural Networks (RN) have been proposed in the references [5, 6, 7]. We consider them to be related to the solution of a problem of cryptanalysis such as the reference [8].

In these works, we prepared six (6) sections. In the first section, we give some explanations from the literature of the topic. Then, in section 2, we prepare neuronal networks by considering biological structure of Neuronal Cells, Formal Neuron and Activation Function. In section 3, we study on Logistic Map which is known by dynamical systems or nonlinear systems theory and Merkle Helman Cryptographic system. In section 4, we propose a new crypto system which is called by "**Neuronal Crypto System**". Also, we give some numerical examples to support our results. Using Neuronal Cryptographic System, we apply validation of test, obtain histograms of images and correlation of Adjacent Pixels as well as entropy in section 5. In the last section, we conclude our work.

2. Neural Networks

2.1 *Biological Model of a Neuronal Cell*

Figure 1 shows a simple model of biological neurons [4] that was used to set up the first neurons. In the brain, neurons are interconnected through axons and dendrites. In the first approach, it can be considered that these kinds of filaments are electrically conductive and can convey messages from one neuron to another. The dendrites represent the inputs of the neuron and outputs of axon. A neuron emits a signal based on signals from other neurons. In fact, it is observed that an integration of the signals received over time at the level of a neuron. It means that it is a kind of summation of the signals. Generally, if the sum exceeds a certain threshold, the neuron emits an electrical signal.

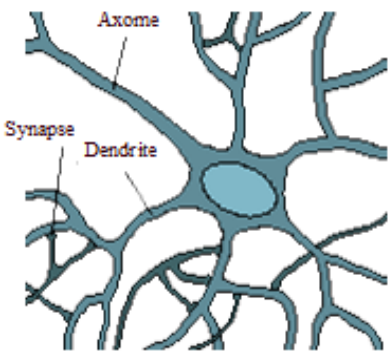

Figure 1
Biological Neuron

The notion of synapse explains the transmission of signals between an axon and a dendrite. At this juncture, there is an empty space that the electrical signal cannot propagate. The transmission is through chemical substances, neuro-mediators. When a signal arrives at the level of the synapse, it causes the emission of neuro-mediators that will attach to receptors on the other side of the inter-synaptic space. When enough molecules are fixed, an electrical signal is emitted on the other side. So, we have a transmission. In fact, the activity of a neuron can enhance or decrease the activity of these neighbours depending on the type of synapse. It is known as an excitatory or inhibitory synapse.

2.2 *Model of a Formal Neuron*

In reference [4], formal neural cobwebs have mathematical samplings of human being brains at the root of initiative. The first work was started with MM. Mac Culloch and Pitts in 1943.The main idea of "modern" neural networks is given as follows:

A simple unit (a neuron) is created. It can carry out a few elementary calculations. Many of these units are then connected to each other. In addition, an attempt is made to determine the computational power of the network which is the result.

First of all, it can be seen that the biological model involves a temporal notion, which is replaced by a simple summation of the signals and arrived to the neuron (these signals are commonly called by neuron inputs).

In our case (Figure 2), we consider a neuron with 8 excitatory synapses through eight bits of a given pixel, multiplying each synapse by a specific weight calculated by an appropriate algorithm.

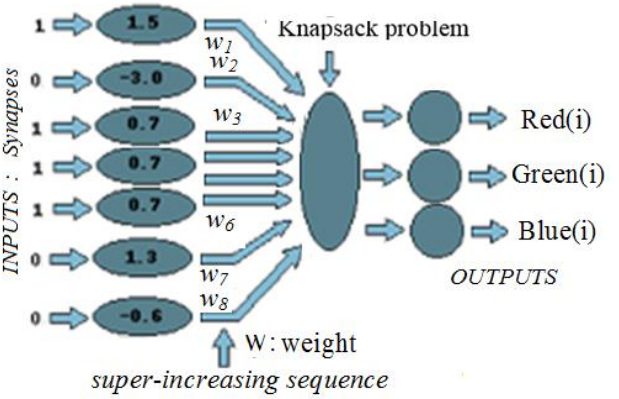

Figure 2
Formal Neuron Proposed

One may construct a super-increasing sequence of eight values for each neuron with the data logistic map. We generate three encrypted pixels that make up a color image RGB-type based on the Merkel Helman algorithm.

2.3 *Activation Function*

Activation function is used to define the internal state of the neuron according to total of inputs.

In this work, we propose to use the affine function g(x) which is one of the simplest activation functions. Activation function is given as

$$g(x) = (p * x + q). \bmod (256)$$

with p being a prime of 256.

2.4. *Logistic Map*

Logistics map which is given references [9, 10] is a polynomial mapping with degree two (or recurrence relation with iteration) and determined by following equation:

$$y_{k+1} = \mu\, x_k\, (1\text{-}x_k) \qquad \textbf{(1)}$$

For population dynamics and chaos theory, it has been used as a model.

Prof. Pierre François Verhulst was the first scientist developed the system in 1845. After him, it was used by mathematical biologist Robert May. He also explored a lot of properties of logistic map.

In the equation (1) which is nonlinear difference equation, we can define following notations:

- x_0**:** İnitial Value.
- y_{k+1}**:** Based on x_k as proportional and called as "future generation"
- x_k**:** Represent the ratio of previous population
- μ**:** It is called by "biotic potential" and it is a positive constant.

This equation (1) has two significant effects. One is reproduction and another is starvation.

So as to understand and work on the dynamic system with other asymptotic results' samples, we need to graph

$$y = \mu\, x(1\text{-}x) \qquad \text{and} \quad y = x$$

diagonal line. For drawing the graphic, we need to explain something as follows:

We obtain the function with a vertical if we start with x_0 initial value. We get following equation when we use initial value

$$y_1 = \mu.x_0\, (1\text{-}x_0)$$

for k=*0*. If we intersect $y = x$ with $y_1 = \mu.x_0\ (1\text{-}x_0)$ for the coordinate $x_0 = x$, we obtain $y_1 = x_1$. Then we get

$$y_2 = \mu.x_1\, (1\text{-}x_1)$$

from the x_1 value of x-axis. We continue iteration like this…

If we substitute μ = 3.9 and x_0=0.01 into the logistic map, Figure 3 is obtained for 100 iterations in the previous operations.

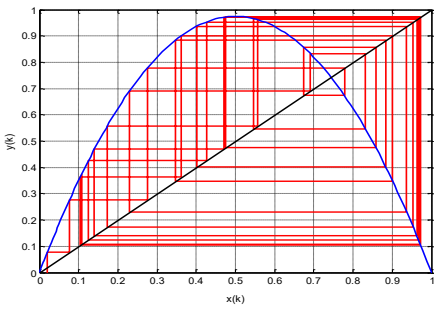

Figure 3a
Evolution of y_k in function of x_k

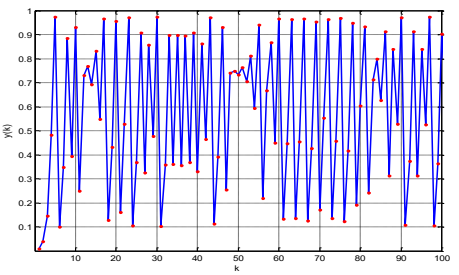

Figure 3b
Chaotic regime in function of k

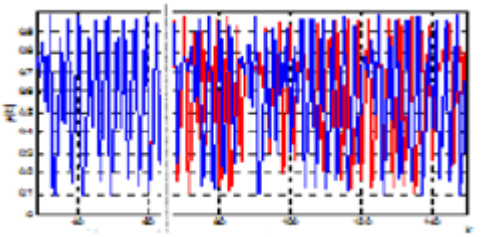

Figure 3c
Initial condition sensitivity

Additionally, Figure 3c shows two signals for y_k when μ = 3.9 with two different initial conditions such as x_0= 0.1 and x_0= 0.100000000000001 which is very close to 0.1.

Note that a small error in knowing x_0 is quickly amplified in phased space. Two different large-scaled signals can also be seen.

Quantitatively, the error (sensitivity to initial conditions) of a highly chaotic system grows exponentially in area.

Since MATLAB can only work on 52-bit, but the system can respond to values less than 10^{-15} depending on the working environment, we should also note that the initial state error is only 10^{-15}.

3. Merkle - Helman Cryptographic System

The cryptosystem Merkle and Hellman [3, 11, 12, 13, 14] uses the knapsack problem as per above. This cryptosystem can be given with details as follows:

3.1 *Generation of keys of crypto system*

- A positive integer n is chosen largely enough (Merkle and Hellman recommend taking n in the order of 100).
- Choose a sequence $\{b_1, b_2,..., b_n\}$ of positive integers checking the following property:

$$\forall\, i \in [2,n], b_i > \sum_{j=1}^{i-1} b_j \tag{2}$$

- Choose an integer M which is called module such as:

$$M > \sum_{i=1}^{n} b_i \tag{3}$$

- Choose an integer W from the closed interval [1, M-1] which is coprime with M. It means that gcd(W, M) = 1.
- Calculate:

$$a_i' = W \times b_i \bmod M \quad \text{for } i \in [1, n] \tag{4}$$

- Find the permutation π from {1, 2,..., n} such that $\{a_{\pi(1)}', a_{\pi(2)}',..., a_{\pi(n)}'\}$ is an increasing sequence.

  The public key is given as

  $$(a_1, a_2,..., a_n) = (a_{\pi(1)}', a_{\pi(2)}',..., a_{\pi(n)}')$$

  This key may be freely distributed to all potential correspondents or stored in a directory comparable to use for phone numbers.
- The private key consists of M, W, $(b_1, b_2,..., b_n)$ and the permutation π.

This private key must always be kept confidential but must not be provided to anyone because it is not needed to encrypt a message. However, it is indispensable to decrypt a message if the system is safe.

3.2 *Principle of Encryption*

1. Encrypted message is written in the terms of the sequence as the binary form: $m_1 m_2... m_n$, with $m_i \in \{0,1\}$.
   Note: if the message is too long, it is cut into blocks of n bits or less.
2. Calculate the encrypted C as:

$$C = \sum_{i=1}^{n} m_i . a_i \tag{5}$$

3. Transmit C.

3.3 *Decryption algorithm*

1. Calculate $d = W^{-1} \times (c \bmod M)$ by using the extended Euclidean algorithm.
2. Calculate $\varepsilon_1, \varepsilon_2,..., \varepsilon_n$ where d is given by:

$$d = \sum_{i=1}^{n} \varepsilon_i . b_i \tag{6}$$

It is very simple to solve this knapsack problem using the following property of b_i:

$$b_i > \sum_{j=1}^{i-1} b_j$$

3. Calculate $m_i=\varepsilon_{\pi(i)}$ where I is in the closed interval [1, n].
4. The decrypted message is written in the terms of the sequence as the binary form $m_1\ m_2 ...\ m_n$.

4. Neuronal Cryptosystem Proposed

We propose a crypto-system inspired from Figure 2 and based on the principle of the Merkle-Hellman cryptosystem [15] with a super-increasing sequence where :

- The inputs $\{I(I, j)_i, i\geq 1..., j=1 \text{ to } 8\}$ are the pixels of the clear image.
- Synapses $\{W_i, i\geq 1\ldots\}$ are positive real values from the logistics map.
- Sommation $CST= \sum W_i M_i$
- The outputs are the value C associated with the pixel of the encrypted image.

One encrypted it as an RGB color image for a monochromatic image.

4.1 *Encryption of the Neuronal Crypto-System*

The plaintext image is encrypted pixel by pixel (8-bits pixel). Let the pixel $\mathbf{i} = (\mathbf{i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7})$ be the value of $i \in \{0,1\}$.

Step 1: A program is used to retrieve data from the logistics map and with predefined initial values. We take the values higher than the initial values and we treat them for each time to find a packet of eight values. That is to say we order with a decreasingly absolute value. Also, we multiply them by one scalar ($K = 10^8$). This scalar is given by the eight values which are different between 0 and 255. Otherwise, this algorithm is rolled out to satisfy this last condition. One may remember the last value found to be set as initial value while using this algorithm again.

The predefined initial values of the logistics map are x(0), μ, N (indice of starting value of logistic map).

Let x(j) be the value of the logistic map and the treated value be defined as follows:

$$P(i, j) = \text{mod}(\ \text{floor}(x(j)*K), 256), j=1, \ldots 8.$$

Example: Let i be a pixel. The program to retrieve data is treated and ordered from the logistics map gives the following result: 21, 64, 103, 122, 193, 201, 224, 253.

j	P(i, j)	j	P(i, j)
1	21	5	193
2	64	6	201
3	103	7	224
4	122	8	253

Step 2: Construct a super-increasing sequence:

Let I be a pixel. We initialize SCT(i) = 0 and calculate the synapses or the eight values (SC(i, j), j = 1, ... 8) as follows for the eight values of j:

For j=1, the 1st synapse takes the small value as follows.

$$\begin{cases} SC(i,1) = P(i,1) \\ SCT(i,j) = SCT(i,j-1) + P(i,j) \end{cases}$$

For j = 2, the 2nd synapse takes the sum of two small values:

$$\begin{cases} \mathrm{SC}(2) = \mathrm{P}(\mathrm{i},1) + \mathrm{P}(\mathrm{i},2) \\ SCT(i,j) = SCT(i,j-1) + P(i,j) \end{cases}$$

For j = 3, the 3th synapse takes the sum of three small values; SC(i, 3)= P(i, 1) + P(i, 2) +P(i, 3)

If we continue like this until the 8th synapse (j = 8), the values of (SC(i, j)) are calculated as follows:

$$\begin{cases} SC(i,j) = SCT(i,j-1) + P(i,j) \\ SCT(i,j) = SCT(i,j-1) + SC(i,j) \end{cases}$$

At the end of the step, we have calculated 9 values such as the eight values of a super-increasing sequence which is the 1st value of synapse is strictly lower than the 2nd value of synapse which is strictly lower than the 3rd value synapse and so on until the 8th and their sum SCT(i, 8).

Example: The sequence values are given by the logistics map as: 21, 64, 103, 122, 193, 201, 224, 253.

1. The first value SC(i,1) of the sequence is equal to:

$$\begin{cases} SC(i,1) = 21 \\ SCT(i,j) = 21 \end{cases}$$

2. The second value SC(i,2) of the sequence is equal to:

$$\begin{cases} SC(i,2) = 21 + 64 = 85 \\ SCT(i,2) = 21 + 85 = 106 \end{cases}$$

The third value SC(i,3) of the sequence is equal to:

$$\begin{cases} SC(i,3) = 106 + 103 = 209 \\ SCT(i,3) = 106 + 209 = 315 \end{cases}$$

3. The fourth value SC(i,4) of the sequence is equal to:

$$\begin{cases} SC(i,4) = 315 + 122 = 437 \\ SCT(i,4) = 315 + 437 = 752 \end{cases}$$

4. The fifth value SC(i,5) of the sequence is equal to:

$$\begin{cases} SC(i,5) = 752 + 193 = 945 \\ SCT(i,5) = 752 + 945 = 1697 \end{cases}$$

5. he sixth value SC(i,6) of the sequence is equal to:

$$\begin{cases} SC(i,6) = 1697 + 201 = 1898 \\ SCT(i,6) = 1697 + 1898 = 3595 \end{cases}$$

6. The seventh value SC(i,7) of the sequence is equal to:

$$\begin{cases} SC(i,7) = 3595 + 224 = 3819 \\ SCT(i,7) = 3595 + 3819 = 7414 \end{cases}$$

7. The eighth value SC(i,8) of the sequence is equal to:

$$\begin{cases} SC(i,8) = 7414 + 253 = 7667 \\ SCT(i,8) = 7414 + 7667 = 15081 \end{cases}$$

Then, the super-increasing sequence is given as following table for the pixel i:

j	SC(i, j)	j	SC(i, j)
1	21	5	945
2	85	6	1898
3	209	7	3819
4	437	8	7667
SCT(i)=		15081	

We can also explain that SCT(i) is calculated as follows:

$$SCT(i) = \sum_{j=1}^{8} SC(i,j)$$

SCT(i)=21+85+209+437+945+1898+3819+7667=15081.

Step 3: On the basis of equation 5:

The first bit on the left of the pixel is associated with the 8th synapse. The 2nd bit of this pixel is associated with the 7th synapse, and so on, until the 8th bit is associated with the first synapse. according to the value of the bits of the plaintext image, the summation is multiplied by the values of the synapses associated with them as follows:

$$SY(i) = \sum_{j=1}^{8} SC(i,j) * i(i, 9-j)$$

This gives us

SY(i)= SC(i, 1)*i(i, 8)+SC(i, 2)*i(i, 7)+SC(i, 3)*
i(i, 6)+SC(i, 4)*i(i, 5)+SC(i, 5)*i(i, 4)+SC(6,i) *
i(i, 3)+SC(7,i)* i(i, 2)+ SC(8,i)*i(i, 1).

Example: if we have the pixel

i = 165 = i(i, j=1, …8) = (1, 0, 1, 0, 0, 1, 0, 1)

then,

SY(i)= 21 *i_8+ 85 *i_7+ 209*i_6+ 437*i_5+ 945*i_4+ 1898*i_3+ 3819*i_2+ 7667*i_1

SY(i)= 21 *1+ ~~85 *0~~+ 209*1+ ~~437*0~~+ ~~945*0~~+ 1898*1+ ~~3819*1~~+ 7667*1

SY(i)=21+209+1898+7667= 9795

is obtained.

1. The value of this sum is divided by the sum of all the values of the pixel's synapses. We calculate values of C(i) as follows:

$$C(i) = \frac{SY(i)}{SCT(i,8)} \leq 1$$

In our example, we get C(i)=0.6494927392082

In the case where C(i)=1, we set it to be equal to C(i)=0.9999999999999 and we apply the same procedure as we will explain.

From this step, we build the resultant encrypted RGB color matrix. So, we have 3 matrices where Cr(i) is a matrix of the red color, Cg(i) is a matrix of the green color, also Cb(i) is a matrix of the blue color.

The following steps are followed:

2. C(i) is multiplied by 256.

CXr(i) = S(i)*256

We keep the integer part of the value found in the 1st matrix (which represents the red space):

In our example:

C(i)*256=0.6494927392082*256=166.27014123731

$$Cr(i) = \lceil SXr(i) \rceil = floor(S(i) * 256)$$

Therefore, we get **C_r(i)= 166**.

The decimal part of (CXr(i)= 0.27014123731) is multiplied by 256. This last value shows that we keep its integer part in the 2nd matrix (which represents the green space):

CXg(i)=SXr(i)-Cr(i)

$$Cg(i) = \lceil SXg(i) * 256 \rceil = \lceil \big((S(i) * 256) - Cr(i)\big) * 256 \rceil$$

$$Cg(i) = \lceil SXg(i) * 256 \rceil = \lceil \big((S(i) * 256) - Cr(i)\big) * 256 \rceil$$

Cg(i)=floor(0.27014123731*256) = floor(69.15615675353), **Cg(i)=69**.

The decimal part of (CXg(i)= (0.15615675353) is multiplied by 256. So, last value implies that we keep its integer part in the 3nd matrix (which represents the blue space):

CXb(i)= (CXg(i)*256)-Cg(i)

$$Cb(i) = \lceil CXb(i) * 256 \rceil$$

Cb(I)= floor(0.15615675353*256) = floor(39.976128903918), **C_b(i)=39**

For the pixel i = **165** it is encrypted with the three values (**166, 69, 39**) which represents the values of the pixels in the three basic colors RGB respectively.

1. But to perfect the encryption and before the transmission of the encrypted pixels, one chooses three numbers (u, v, w) coprime with 256 to transmit these data to the following step:
   - $C_r(i) = u*C_r(i) \bmod (256)$,
   - $C_g(i) = v*C_g(i) \bmod (256)$,
   - $C_b(i) = w*C_b(i) \bmod (256)$.

   In our case, we have u= v = w = 1.
2. The three values found ($C_r(i)$, $C_g(i)$, $C_b(i)$) will be calculated according to the affine system $(p.x + q) \bmod 256$.

   In this case, we get p = 1, q = 0.
3. We proceed to the encryption of another pixel by following the same steps above, but the last value of the logistic map used in the previous pixel is set as initial value when using the chaotic algorithm.
4. We can represent which one can be the first matrix (red / blue / green), and which one can be the second matrix. In this case, we must choose among these six possibilities.

bits	order	bits	order
000	BVR	100	VRB
001	RVB	101	VBR
010	RBV	110	RVB
011	BRV	111	RVB

4.2 *Decryption of the Proposed Neuronal Crypto-System*

First, we follow the same steps to find the super-increasing sequence as the encryption for a given pixel i. Besides, same steps are used to decipher (SC(i, j) / j=1 to 8). Besides, we calculate CST(i) and receive three encrypted pixels (i_r, i_g, i_b) that can be associated with it.

The obtained value is calculated according to the affine system $(px+q) \bmod 256$.

The received cryptogram is decrypted by the affine system $(p^{-1}(y-q) \bmod 256)$ and following equations are obtained;

$$C_r(i) = u^{-1}*C_r(i) \bmod (256),$$
$$C_v(i) = v^{-1}*C_v(i) \bmod (256),$$
$$C_b(i) = w^{-1}*C_b(i) \bmod (256).$$

1. According to this formula:

$$SY(i)=((Cr(i)+(Cv(i)+Cb(i)/256)/256)/256)*SCT(i, 8)$$

   the encrypted pixel is recovered from the RGB color matrices.

   If we have a decimal number, then we take the whole number that follows it.

   In our case, if we receive:

$$(Cr(i), Cg(i), Cb(i)) = (166, 69, 39),$$

We can follow the same steps to find the super-increasing sequence as the encryption for this pixel (SC(i, j) / j=1 to 8). Additionally, we calculate.

SCT(i)= 15081,

$$SY(i) = \left(166 + \frac{69+\frac{39}{256}}{\frac{256}{256}} \right) * 15081$$

SY(i) =9794.9991225600, then SY(i) =9795

From this moment we use the Merkel Helman decryption (or Knapsack Problem) to find the desired pixel.

N	Sc(i,j)	Merkel Helman decryption	Coordinate of decoded pixel
1	7667	9795≥7667→9795-1*7667=2128	128*1=128
2	3819	2128<3819→2128-0*3819=2128	64*0=0
3	1898	2128≥1898→2128-1*1898=230	32*1=32
4	945	230<945→230-0*945=230	16*0=0
5	437	230<437→230-0*437=230	8*0=0
6	209	230≥209→230-1*209=21	4*1=4
7	85	21<85→21-0*85=21	2*0=0
8	21	21≥21→21-1*212=0	1*1=1
decoded pixel		128+32+4+1=165	

4.3 Encryption Key of the Proposed Neuronal Cryptosystem

The cryptosystem has a behavior of a secret key system. Almost the same steps are used in the encryption and the decryption. The encryption key can be shared in fields as follows:

- The predefined initial values of the logistic map are x_0, μ, N starting index to collect values from the logistics map where x_0, μ are double precision numbers (10^{-16}) and N is a 8-bit encoded takes values from 1 to 256).
- The multiplier K is in single precision (10^{-8}).
- The four numbers (p, u, v, w μ) coprime with 256 plus the shift value q. The five values will be 8-bit encoded (take values from 1 to 256).
- The three bits are used to represent the RGB type image's matrix.

So we have:

$$10^{16} \text{ x } 10^{16} \text{ x } 2^8 \text{ x } 10^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^3 =$$

$$10^{40} \text{ x } 2^{51} .= 2^{133} \text{ x } 2^{51} .= 2^{184}$$

(with $10^3 \approx 2^{10}$),

$40= 13*3+1$, $10^{40}=(10^3)^{13}*10 \approx 2^{133}$.

Then, we obtain a key from Proposal Neuronal Crypto-System as 133+51=184 bits.

5. Validation Tests of this Cryptographic System

We work on the plaintext and encrypted images of Lena with the following conditions:

Initial values as $x_0 = 0.01$ and μ = 3.9, K = 10^8, N = 25,
p = u = v = w = 1, q= 0, RGB = 000.

5.1 *Histogram of the Images*

For monochrome images, the histogram is defined in Reference [16, 17] as a discrete function representing the number of pixels per intensity value. By counting the number of intensity pixels for each image, the histogram is determined. A histogram can be viewed as a probability density function. Additionally, histograms are resistant to certain transformations of the image. They are invariant to rotations, translations, and to a lesser extent changes in perspective or scale. From the obtained results, we can clearly see that there are significant differences between the plaintext image and the corresponding ciphertext image. In addition, the histogram of the ciphertext image is uniform. This makes it difficult to extract the statistical properties of pixels from text-only images.

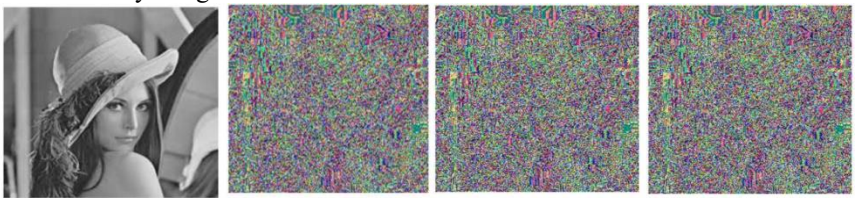

Figure 4

Plaintext image of Lena and Encrypted image in space of // red // green // blue.

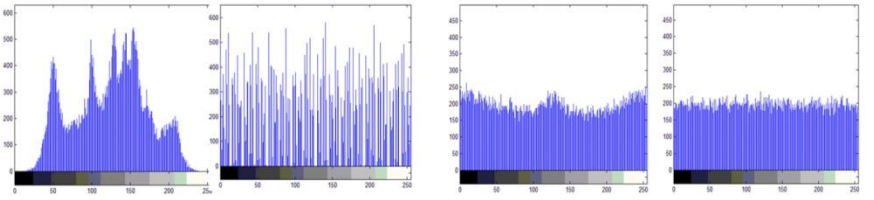

Figure 5

Histogram of Plaintext image of Lena and Encrypted image in space of // red // green // blue

In summary, the histograms of plaintext and encrypted images of "Lena" show that the proposed cryptosystem works properly.

5.2 *Correlation of the Adjacent Pixels*

In probability theory or statistics, the study of the correlation between two random variables, or in numerical statistics, the study of the strength of the connection between these variables. Affine relationships and linear regression are the connections we are looking for. For example, we want to

calculate the correlation coefficient between two data sets of the same length (typical case: regression). Suppose we have the following table of values:

$$X(x_1, \ldots, x_n) \text{ and } Y(y_1, \ldots, y_n)$$

For both series. The measure of correlation can be calculated based on the Bravais-Pearson linear correlation coefficient in Ref. [16].

We apply the following formula:

$$Coef(X,Y) = \frac{cov(X,Y)}{\sqrt{D(X).\sqrt{DY}}} \tag{7}$$

for the correlation coefficient of these sets.

Covariance between x and y is given as follows:

$$cov(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\big((X_i - E(X)\big)(Y_i - E(Y))) \tag{8}$$

The average of X is given by :

$$E(X) = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{9a}$$

The average of Y is defined by :

$$E(Y) = \frac{1}{N}\sum_{i=1}^{N} Y_i \tag{9b}$$

The standard deviation of X is determined as:

$$D(X) = \frac{1}{N}\sum_{i=1}^{N}(X_i - E(X))^2 \tag{10a}$$

Also, the standard deviation of Y is given as :

$$D(Y) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - E(Y))^2 \tag{10b}$$

Value of the pixel at the position (x, y)

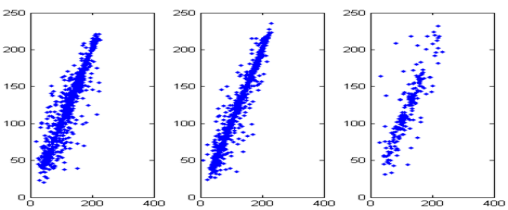

Figure 6

Correlation between adjacent pixels: of the plaitext image (horizontal, vertical, and diagonal)

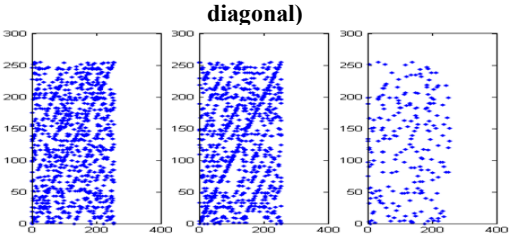

Figure 7.a

Correlation between adjacent horizontal, vertical, and diagonal pixels: from the encrypted image to the red color space.

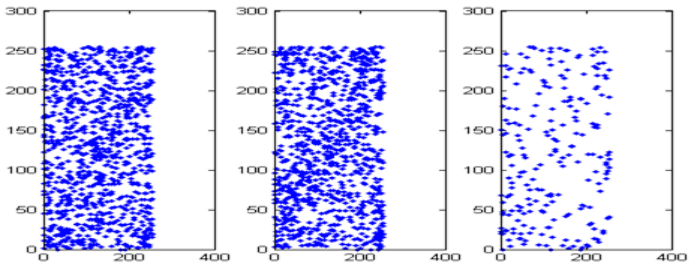

Figure 7.b

Correlation between adjacent horizontal, vertical, and diagonal pixels: from the encrypted image to the green color space.

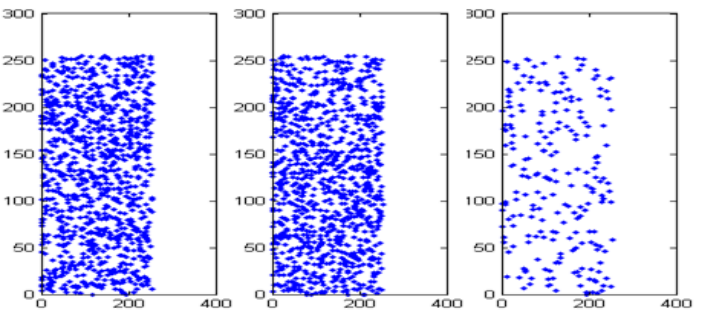

Figure 7.c

Correlation between adjacent horizontal, vertical, and diagonal pixels: from the encrypted image to the blue-collar space

The correlation coefficient is between -1 and 1. The intermediate value represents the degree of linear correlation. We simply use the term "highly correlated" to describe two variables because the closer the coefficients are to the extreme values -1 and 1, the stronger the correlation between the variables. A correlation value of 0 means the variables are not correlated. To test the correlation coefficient, we randomly selected 1500 pairs of two adjacent pixels in encrypted and unencrypted images.

Figures 6 and 7 show the correlation between two adjacent pixels of the plaintext image and the encrypted image.

We see that adjacent pixels in the plaintext image are correlated (coefficient = 0.94024), while in the encrypted image they have a lower correlation (coefficient = 0.0031). The low correlation between two adjacent pixels in an encrypted image makes attacks on cryptographic systems more difficult.

Furthermore, it is obvious that the plaintext image may have several scattered lines, but between all these lines the salient property appearing on the right side of the shape Y = aX + b, which represents a linear dependence, can be maintained.

5.3 *Entropy*

Shannon entropy, as defined in reference [16, 17], is a mathematical function. Intuitively, it corresponds to the amount of information contained or provided by the information source. For a source of discrete random variables X consisting of n symbols, each symbol xi has a probability Pi of occurrence. The entropy H of source X is defined as

$$H(x) = -\sum_{i=1}^{n} P_i . log_2 (P_i) \quad (11\ a)$$

We pose

$$P_i = \frac{k_i}{n} \quad (11b)$$

Where i varies between 0 and 255. n is the number of values generated in our case (n = 256 * 256 = 65536). k_i is the frequency with which each number i occurs. Usually the logarithm based on 2 is used because entropy has bit/symbol units.

This is the information that would be obtained on average by observing the parallel symbols output by a very large number of identical sources without memory. It is also the average information per symbol that would be obtained by observing a series of very long symbols, since the source is stationary but without memory.

It comes from a single source. Consider a source with a 256-character alphabet. If all these characters are equally likely, the entropy associated with each character is $\log_2(256) = \log_2(2^8) = 8$ bits. This means that 8 bits are required to transmit one character. Therefore its entropy is 8 bits.

The following table shows the entropy of different images through our neuronal cryptosystem:

Image	Plaintext	Red space	Green Space	Blue Space
Lena	7.4651	7.3338	7.9903	7.9968
Cameraman	7.0917	7.0124	7.9789	7.9941
Barbara	7.4189	7.3316	7.9906	7.9966

Our proposal cryptosystem gives us a ratio between the image crypted and a source for the both spaces green and blue. It delivers equiprobable characters is 99.79% Therefore, our proposal cryptosystem passes the entropy test.

6. Conclusion

We built a cryptographic system from a simple model of artificial neurons. We know that the connections between the neurons describe the topology of the model. It is possible to distinguish a certain regularity in the network with complete connection.

In our model we used the structure of a monolayer network. Input-organized neurons are fully connected to other neurons organized as outputs. Their synapses have variable weights by selecting the random values from the logistics map.

The three indicators that make it possible to estimate, This algorithm have been calculated by three indicators as follows;

1. Histogram of Images,
2. Adjacent pixel correlation,
3. Entropy,

Not only can encrypted images be visualized, but the efficiency of this cryptographic system can also be tested. All these tests provide a good evaluation of this cryptosystem. In addition, this cryptosystem has a very large key space (184 bits) and is resistant to all types of brute force attacks.

The novelty of our system is to create greyscale images into RGB type images. The proposed system is called as Neuronal Crypto-System a secret key system that considers steps of encryption with decryption of a public key cryptosystem.

Conflict of interest

All authors have had a role in (a) conception and design or analysis and interpretation of data; (b) drafting or critical revision of the article for important intellectual content; and (c) approval of the final version.

This manuscript has not been submitted to any other journal or publication, nor is it currently under consideration by any other journal or publication.

The authors are not affiliated with any organisation having a direct or indirect financial interest in the topic discussed in the manuscript.

References

[1] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York: John Wiley & Sons, Inc. (1996).

[2] D. Stinson, Cryptography: Theory and Practice, Paris: Vuibert Informatics (2001).

[3] R. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsacks," *IEEE Trans. Inform. Theory,* vol. IT-24, pp. 525-530, Sept. (1978).

[4] J. Hertz, A. Krogh, and R. G. Palmer, An Introduction to the Theory of Neural Computation, Addison-Wesley (1991).

[5] B. Chandra and P. P. Varghese, "Applications of Cascade Correlation Neural Networks for Cipher System Identification," *World Academy of Science, Engineering and Technology*, vol. 26 (2007).

[6] A. Merzoug, A. A. Pacha, N. Hadj-Said, and H. A. Pacha, "Production of a super-increasing sequence based on the Fibonacci sequence," *Malaysian Journal of Computing and Applied Mathematics,* vol. 2, no. 1, pp. 16-24 (2019).

[7] A. Merzoug, A. A. Pacha, N. Hadj-Said, M. Mamat, and M. A. Mohamed, "Generating a random sequence based on PWLCM map applied to RC4 algorithm," *Int. J. Eng. Technol.,* vol. 7, no. 2.14, pp. 182-185, Apr. (2018), DOI: 10.14419/ijet.v7i2.14.12821.

[8] C. Li, S. Li, D. Zhang, and G. Chen, "Cryptanalysis of a chaotic neural network-based multimedia encryption scheme," in Proc. PCM (3), pp. 418-425 (2004).

[9] J. Gleick, Chaos: Making a New Science, Albin Michel, 1987, ISBN: 0-7493-8606-1.

[10] L. Devaney, A First Course in Chaotic Dynamical Systems, Westview Press, Oct. 21 (1992), ISBN: 9780813345475.

[11] E. Karnin and M. Hellman, "The largest super-increasing subset of a random set," *IEEE Trans. Inform. Theory,* vol. IT-29, no. 1, pp. 146-148, Jan. (1983).

[12] L. Bournon, "Comparative study of several families of knapsack," DEA thesis, Université d'Aix Marseille II, Sept. (1991).

[13] A. Clark, E. D. Dawson, and H. Bergen, "Combinatorial optimization and the knapsack cipher," Cryptologia, vol. 20, no. 1, pp. 85-93, Jan. (1996).

[14] B. Chor and R. L. Rivest, "A knapsack-type public-key cryptosystem based on arithmetic in finite fields," *IEEE Trans. Inform. Theory,* vol. 34, no. 5, pp. 901-909 (1988).

[15] S. M. Stefanov, "Characterization of the optimal solution of the convex separable continuous knapsack problem and related problems," *Journal of Information and Optimization Sciences*, vol. 42, no. 1, pp. 1-16 (2021), DOI: 10.1080/02522667.2019.1624048.

[16] H. Ali-Pacha, N. Hadj-Said, and A. B. Ali-Pacha, "Data security based on homographic function," *Pattern Recognition Letters*, vol. 129, pp. 240-246 (2020).

[17] H. Ali Pacha, N. Hadj-Said, and A. Ali-Pacha, "Image encryption using a specific adaptation of Lehmer's algorithm," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, no. 5, pp. 949-971 (2020), DOI: 10.1080/09720529.2019.1652402.

Received July, 2023