Malaysian Journal of Computing and Applied Mathematics 2019, Vol 2(1): 25-33
©Universiti Sultan Zainal Abidin
(Online)

MALAYSIAN JOURNAL OF COMPUTING AND APPLIED MATHEMATICS

Dynamic Generalization of the Vigénère Table based on the Logistics Map

Hana Ali-Pacha, Naima Hadj-Said, Adda Ali-Pacha

Laboratory of Coding and Security of Information, University Science and Technology of Oran Mohamed Boudiaf, BP 1505 El M'Naouer Oran 31000, Algeria

*Corresponding author: hana.alipacha@univ-usto.dz

Received: 07/04/2019, Accepted: 12/06/2019

Abstract

The Vigenère cipher is a system that allows us to substitute a character with another character that is not always the same. The disadvantage of this system is the weak length of the encryption key, which is usually less than the length of the text to be encrypted, and can unfortunately be discovered by using Kasiski's and coincidence index methods.
To resolve this problem, we propose in this work to use a dynamic generalization of the Vigénère table based on the logistics map to generate a random keys as long as the length of the text.

Keywords: Vigénère cipher; Kasiski Test; Chaotic Theory; Logistic Map; Sensibility of Initials Conditions.

Introduction

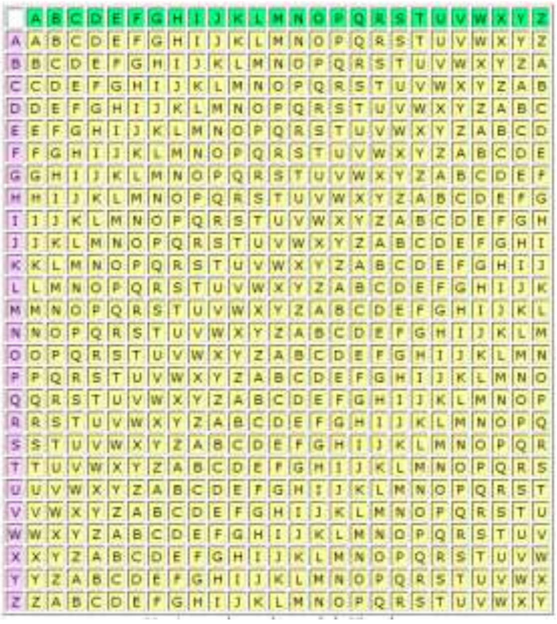
Cryptography has experienced a real explosion with the development of computer systems, moving from a traditional and confidential era to very high-tech systems requiring significant computing power. Several studies (Kocarev, 2001; Luca, 2006) have shown that chaotic signals can become the alterative of several common cryptographic systems due to their random behavior, their high sensitivity to initial conditions.

In recent years, several studies have been done on the Vigenère cipher (Singh, 2012; Rahmani et al., 2012; Ali and Sarhan, 2014; Al-Ahwal and Farid, 2015; Aliyu and Oliniyan, 2016; Wilson and Garcia, 2006; Saputra and Hasibuan, 2017; Subandi et al., 2017), which is a polyalphabetic encryption system and this is substitution cipher (Schneier, 1996), the same letter of the plaintext message may, depending on its position in the latter, be replaced by different letters.

In this work, we will propose a dynamic generalization of the Vigénère table of dimension of 256x256 boxes, the values are modulo 256. These values correspond to the integer values of the ASCII code, if we use them in the encryption of the texts. The dynamism of the Vigénère table is based on the logistics map (Devaney, 1989; Devaney, 1992; Pareek et al., 2006).

Vigénère Cipher

The Vigenère cipher, (Vigenère Tableau, Vigenère Table or Vigenère Square), uses a square matrix of 26 lines and 26 columns, each box of a row or each box of a column contains a letter of the Latin alphabet. The boxes in the first row (respectively column) contain the letters of the alphabet in increasing order (from A to Z). For the second row (respectively column), we start with the second letter of the alphabet (B) then by ascending order (from B to Z) and finally A. We do the same for all lines (respectively column), at each i^{th} line (respectively column) we start with the i^{th} letter of the alphabet and then in ascending and circular order until the $(i-1)^{th}$ letter of the alphabet (Figure 1a).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 1a. Vigenère Table

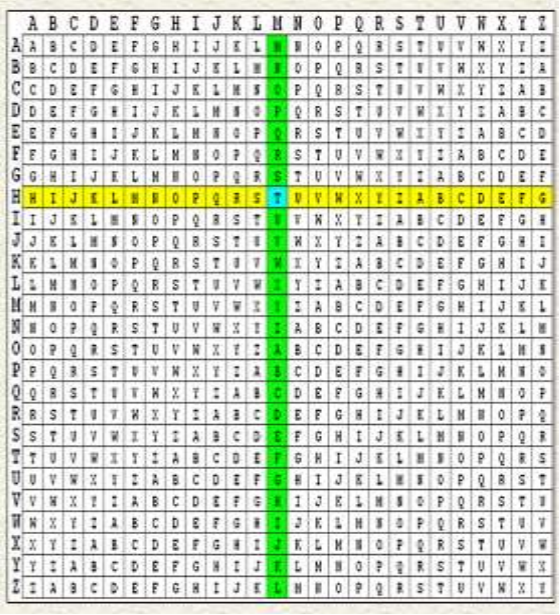
In addition to the plaintext, the Vigenère cipher also requires a keyword, which is repeated so that the total length is equal to that of the plaintext. It is agreed in the Vigénère cipher to use letters to uppercase and to encrypt with blocks of 5 letters.

For example, suppose the plaintext is **MEDITERRANEAN** and the keyword is **HELP-ME**. Then, the keyword must be repeated.

Plaintext: M E D I T | E R R A N | E A N
Key: H E L P M | E H E L P| M E H
Ciphertext: T I O X F | I Y V L C| Q E U

To encrypt, we write the text to be encrypted in a first line and in the second line, we write under each letter of the plaintext a letter of the keyword.

Figure 1a is used to encrypt the data (the row index is corresponding to the keyword letter, and the column index is corresponding to the plaintext letter). For example, the first letter in the plaintext is **M** and its corresponding keyword letter is **H**. This means that the row of **H** and the column of **M** are used, and the entry **T** at the intersection is the encrypted result (Figure 1b). Repeating this process until all plaintext letters are processed**.**



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 1b. Vigenère Table

To decipher, we take the letter line of the key that corresponds to the cryptogram to be deciphered, and we look at the column that carries this cryptogram, this column is the deciphered letter.

For example, to decrypt the first letter **T** in the ciphertext, we find the corresponding letter **H** in the keyword. Then, the row of **H** is used to find the corresponding letter **T** and the column that contains **T** provides the plaintext letter **M** (see the above figure 1.b). Consider the fifth letter **F** in the ciphertext. This letter corresponds to the keyword letter **M** and row **C** is used to find **F**. Since **F** is on column **M**, the corresponding plaintext letter is **C**.

Mathematically, we identify the letters of the alphabet with numbers from 0 to 25 (A = 0, B = 1 ...). The encryption and decryption operations are, for each letter, those of the Caesar figure. By designating the i^{th} letter of the plaintext by Text[i], the ith of the encrypted by C[i], and the i^{th} letter of the key, repeated enough times, by K[i], it is formalized by:

$$C[i] = (Texte[i] + K[i]) \text{ modulo } 26 \tag{1}$$

where, x modulo 26 designates the remainder of the integer division of x by 26. For encryption it is sufficient to add the two letters and then subtract 26 if the result exceeds 26.

$$Texte[i] = (C[i] - K[i]) \text{ modulo } 26 \tag{2}$$

The decryption is an operation identical to that of the encryption, for that, it suffices to subtract the value of the key from the value of the cryptogram and, to add 26 if the result is negative:

$$Clé'[i] = 26 - Clé[i] \tag{3}$$

Logistic Map

Logistics map (Devaney, 1989; Devaney 1992) is a well-known dynamic in non-linear systems theory, defined by equation (4):

$$y_{k+1} = r\, x_k\, (1-x_k) \tag{4}$$

It gives a perfect explanation of a dynamic system behavior. This system was developed by Prof. Pierre François Verhulst (1845) to measure the evolution of a population in limited environment, later used in 1976 by the biologist Robert May to study the evolution of insect population:

a. y_{k+1}: Generation in the future that is proportional to x_k.
b. x_k: Previous generation.
c. r: Positive constant incorporates all factors related to reproductive, successful overwintering eggs for example, etc.

In order to study this dynamic system and some asymptotic individuals' models, the first thing to do is to draw the parabolic graph $y= r.x\ (1-x)$, and the diagonal $y=x$. The operation that we will follow to draw the iterative form y_{k+1} according to x_k is simply summarized as following:

a. Starting from an initial value x_0 of the x-axis, we reach the function with a vertical; the function takes the value $y_1=r.x_0\ (1-x_0)$,
b. From horizontal $y_1= r.x_0\ (1-x_0)$ of the previous point, we join the line $y = x$;
c. We represent the abscissa of the intersection with the vertical line $x=x_0$; we have $y_1 = x_1$
d. From the x_1 value of the x-axis, we reach the function with a vertical; the function takes the value $y_2= r.x_1\ (1-x_1)$; and so on.

We take $r = 3.9$ and, $x_0=0.01$ for logistics map, the previous operations for 100 iterations are represented in Figure 2a and Figure 2b.

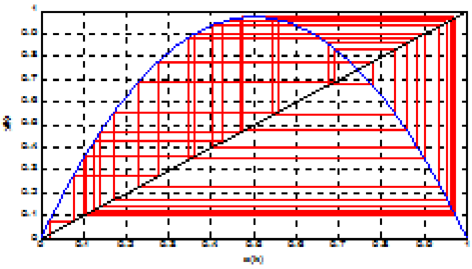


Figure 2a. Evolution of y_k in function of x_k

Figure 3 shows two signals generated from the chaotic logistic map (r = 3.9), one with an initial condition x_0= 0.1 and the other very close with x_0=0.099999999. We note that a very small error on the knowledge of the initial state x_0 in the space of the phases will be quickly amplified and gives us two widely different signals. Quantitatively, the growth of error is locally exponential for highly chaotic systems (sensitivity to initial conditions).

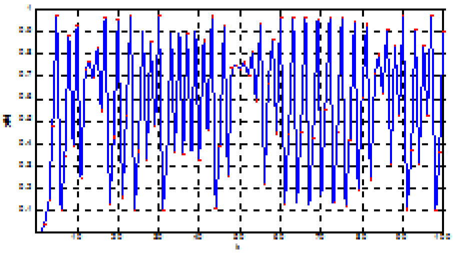


Figure 2b. Chaotic regime in function of k

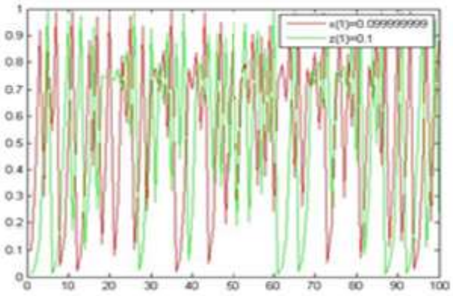


Figure 3. Sensitivity to initial conditions

It should be noted that the error on the initial conditions in this case is 10^{-15} and this is the smallest value because Matlab works with only 52 bits but the system can be sensitive to values smaller than 10^{-15} depending on the work environment.

New system of generalization of Vigénère table with Chaotic Key

The proposed system works in the following steps:

a. We generate data from the logistic map of equation 3a, with x0 = 0.2; r = 3.99;
b. We begin the identification of the data starting from a value N (97 for our result).
c. Then, we take inspiration from the generator Ali-Pacha et al. (2007) to realize a generalization of the table of Vigenere with integer values of ASCII code. We multiply each value by a scalar M; like an amplifier type to allow the decimal parts to be submerged (in this case M = 1000000).
d. We take the integer parts of these real values, and after the modulo 256 is applied to these parts.

$$y(i)=\text{Logistic\_map}\ (x_0,\ r,\ N); \qquad (5a)$$
$$z(i)=\text{floor}(\text{mod}(y*M,\ 256)); \qquad (5b)$$

If, we assume that an image M is square and 256 lines and 256 columns, can be represented by a 256x256 pixels dimension vector. We will have the generalization of the vigénère algorithm adapted to the images, from equation (5), as follows:

$$C[i] = (M[i] + z[i])\ \text{modulo}\ 256 \qquad (6a)$$
$$M[i] = (C[i] - z[i])\ \text{modulo}\ 256 \qquad (6b)$$

Length of the secrete Key of the cryptosystem.
The size of the encryption key space is the total number of different values that can be used in the encryption process. In the proposed algorithm, the secret key field is set as follows:

$$ST=\{x0,\text{r}\ ,N,\text{M}\} \qquad (7)$$

where x_0, r, are double precision numbers and N and M are integer constants (starting index value).

If the calculating precision of: x_0, r, is 10^{-15}, and $N \in [1, 1000]$. Therefore, the key space is larger, than $10^{15}\times10^{15}\times10^{3}\times10^{6} = 10^{39}(avec10^{3}\approx2^{10})$ in this case, we will have a key field of the order of ^{2130,} and it is huge. Therefore, the encryption algorithm has a very large key space to withstand all kinds of brute force attacks.

Cryptanalysis
The Vigenère cipher was broken by the Prussian major Friedrich Kasiski who published in 1863 an effective method (Kasiski test) to determine the size of the key, by identifying the repetition of certain motifs in the encrypted message.
Statistics based on the coincidence index, discovered in the twentieth century, are even more effective in breaking the Vigenère cipher.
a. Coincidence Index Method
Note that all spaced letters of k (where k is the length of the key) are shifted by the same constant. It is therefore sufficient to perform a frequency analysis for each of the sub-texts. The coincidence index represents the probability that two letters randomly selected in a text are identical. For the English language, the TH is approximately equal to 0.00302, this index does not vary if the text is coded with a monoalphabetic substitution. By testing different key lengths,

and keeping the lengths for which the TH is closest to 0.00302, we can deduce the length of the key.

b. Kasiski test method

If we know the number of symbols in the key, it becomes possible to proceed by frequency analysis on each of the sub-texts determined by selecting letters of the clear message at intervals the length of the key (as many sub-texts as the length of the key). This is the well-known attack on mono-alphabetic ciphers. Kasiski's method is to look for repetitions in the ciphertext. Usually we look for repetitions of minimum 3 letters. The idea is that the same sequence of letters of the plain text has been encrypted with the same part of the key and therefore the same sequence of letters has been repeated in the ciphertext. By noting the number of letters separating the redundant sequences, one can obtain a multiple of the length of the key.

For example, a repeating pattern is separated by 24 characters and another pattern separated by 16 characters. We search the common divisors, the key can be of lengths 1, 2, 4 or 8.

c. Cryptanalysis in our case

In our case, cryptanalysis is difficult; the two previous methods cannot apply. On the one hand, the length of the key is as long as the text to be encrypted; on the other hand, the key is random. With a few precautions, this system can be identified with the one-time pad (OTP), which is an ideal encryption system.

Results and Interpretations

We use BMP images (Lena) in our application of this new crypto system. We take as the fixed values of the encryption key: r = 3.9, x_0=0.95, with N=97 value starting of logistics map.

Histogram Images

For a monochrome image, that is to say, a single component, the histogram is defined as a discrete function that maps each value of the number of pixel intensity taking this value. The determination of the histogram is carried out by counting the number of pixel intensity for each of the image. The histogram can then be seen as a (Chen et al., 2004) probability density.

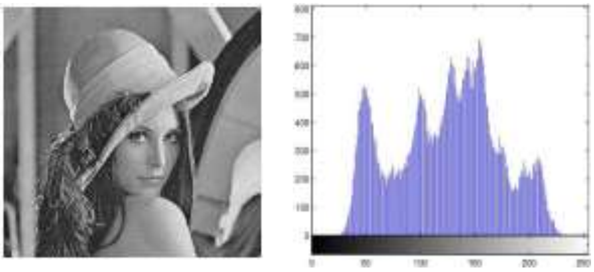


Figure 4. Image of Lena in plaintext and its histogram

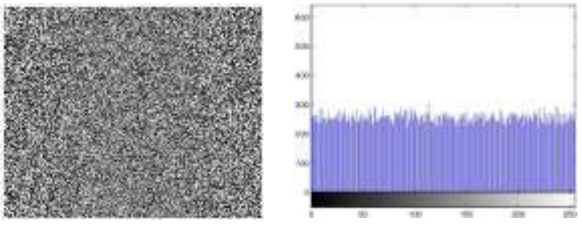


Figure 5. Image Lena encrypted and its histogram

Referring to the results obtained, we can clearly see that the simple image differs significantly from the corresponding encrypted. Moreover, the histogram of the encrypted image is fairly uniform which makes it difficult to extract the statistical nature of the simple image pixels. Histograms of plaintext and encrypted images of Lena showed that the proposed crypto system works correctly. It is found that:

a. Encryption changes the frequency of pixels with the same probability distribution for the entire image.
b. The pixels are highly correlated in the clear image in the encryption cancels any correlation between them in the encrypted image.
c. The image after encryption has become noisy and do not contain visible information that shows on the histogram of the two images is no information in the clear picture.

Correlation between two adjacent pixels

To test the correlation between two adjacent pixels horizontally, vertically and diagonally to the image is calculated correlation coefficient for a sequence of adjacent pixels (Chen et al., 2004. Let be x and y are the adjacent pixels. We assume that we have the following tables of values: X(x_1, …, x_n) and Y(y_1, …, y_n) and for each of the two series. A measure of this correlation is obtained by calculating the linear correlation coefficient of Bravais-Pearson. To know the correlation coefficient linking these two series, we apply the following formula (Chen et al., 2004):

$$Coef(X,Y) = \frac{cov(X,Y)}{\sqrt{D(X)}.\sqrt{DY}} \qquad (8)$$

The covariance between x and y is given as follows:

$$cov(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\big((X_i - E(X)).(Y_i - E(Y))) \qquad (9)$$

The average of X:

$$E(X) = \frac{1}{N}\sum_{i=1}^{N} X_i \qquad (10a)$$

The average of Y:

$$E(Y) = \frac{1}{N}\sum_{i=1}^{N} Y_i \qquad (10b)$$

The standard deviation of X is

$$D(X) = \frac{1}{N}\sum_{i=1}^{N}(X_i - E(X))^2 \qquad (11a)$$

The standard deviation of Y is

$$D(Y) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - E(Y))^2 \qquad (11b)$$

The correlation coefficient is between -1 and 1. The intermediate values tell us about the degree of linear dependence between the two variables. The correlation between the variables is strong, if the coefficient is close to -1 or 1; the term "highly correlated" is simply used to qualify the two variables. The correlation coefficient is close to 0 it means that the variables are not correlated.

We will be studying the distribution of 1,000 adjacent pixels that are randomly selected from the image clear and encrypted with the "randsrc". The "randsrc" randsrc (m, n, [symbols]) generates a matrix of size MxN equiprobable symbols, and obtaining a distribution of adjacent pixels.

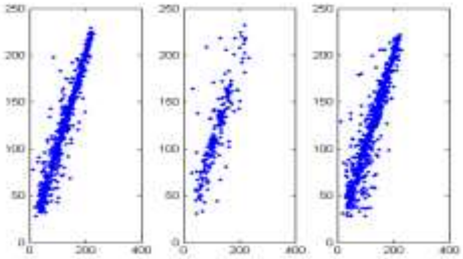


Figure 6. Distribution of adjacent pixels of the plaintext

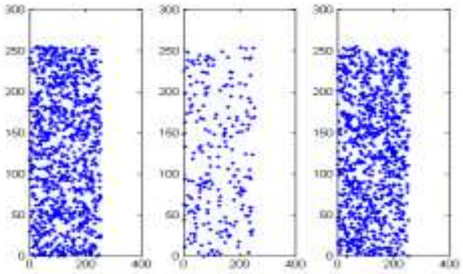


Figure 7. Distribution of adjacent pixels of the encrypted image

As an example, we found the following values for the horizontal correlation coefficient of plaintext image of Lena is (0.95247) and for her encrypted image is equal to (0.0037). Figures 6 and 7, which are three in number from left to right as follows: Correlation of horizontal pixels, vertical pixels Correlation, Correlation diagonal pixels:

a. The adjacent pixels are highly correlated on the encryption created a major mess.
b. The autocorrelation coefficients close to 1 for the images clear and encryption cancels proving the proper functioning of our system.
c. In addition, it is clear that in the plaintext image several lines can adjust to this cloud of points. Nevertheless, among all these lines we can retain one that enjoys a remarkable property giving rise to a linear line; affine form (Y = aX + b, the coefficient a, represent the correlation).

Entropy test

The average amount of information (Chen et al., 2004) associated with each symbol without memory source is defined as the expected value (denoted by E {.}) Specific information provided by the observation of each of the possible symbols $\{S_1, ..., S_n\}$:

$$H(s) = -\sum_{i=0}^{n} p_i \log_2(p_i) \tag{11}$$

This is the information that would be obtained in average by observing the parallel symbols output from very large number of identical source without memory. As the source is stationary and without memory, it is also the average information per symbol, which would be obtained by observing a series of very long symbol emitted by a single source.

One found the following values for the entropy of plaintext image of Lena is (7.4651) and for her encrypted image is equal to (7.9965). It is found that the entropy of the images increases to almost 8 bits showing that encryption creates a high level of disorder.

Conclusion

By making the generalization of the Vigenere table integer values of ASCII code, to produce keys as long as the text to be encrypted, we obtain the Verman system (one-time pad).

In our crypto-system, we use a polyalphabetic substitution of Vigénère type; where the Vigénère encryption key is produced by the logistics map (random values) and, it is the same size as the plaintext to be encrypted, in other words, it is a substitution of type of a one-time pad (OTP).

The one-time pad (OTP) is an encryption technique that cannot be cracked, but requires the use of a one-time pre-shared key the same size as, or longer than, the message is being sent. In this technique, a plaintext is x-ored with a random secret key. Therefore, it is an ideal encryption.

Conflict of Interests

There is no conflict of interest regarding the publication of this paper.

Reference

Kocarev, L. (2001). Chaos-based cryptography: a brief overview. IEEE Circuits and Systems Magazine, 1(3), 6-21.

Luca, M B. (2006). Apports du chaos et des estimateurs d'états pour la transmission sécurisée de l'information. l'Université de Bretagne Occidentale.

Singh, Y K. (2012). Generalization Of Vigenere Cipher. ARPN Journal of Engineering and Applied Sciences, 7(1), 39-44.

Rahmani, K I, Wadhwa, N and Malhotra, V. 2012. Alpha-Qwerty Cipher: An Extended Vigenère Cipher. Advanced Computing: An International Journal, 3(3), 107-118.

Ali, F M S, Sarhan, F H. (2014). Enhancing Security of Vigenere Cipher by Stream Cipher. International Journal of Computer Applications, 100(1), 1-4.

Al-ahwal, A, Farid, S. (2015). The Effect Of Varying Key Length On A Vigenère Cipher. Journal of Computer Engineering, 17(2), 18-23,

Aliyu, A M, Olaniyan, A. (2016). Vigenere Cipher: Trends, Review and Possible Modifications. International Journal of Computer Applications, 135(11), 46-50.

Wilson, P I and Garcia, M. (2006). A Modified Version of the Vigenère Algorithm. International Journal of Computer Science and Network Security, 6(3B), 140-143.

Saputra, I, Hasibuan, N A. (2017). Vigenere Cipher Algorithm with Grayscale Image Key Generator for Secure Text File. International Journal of Engineering Research & Technology, 6(1), 266-269.

Subandi, A, Meiyanti, R, Sandy, C L M, Sembiring, R W. (2017). Three-Pass Protocol Implementation in Vigenere Cipher Classic Cryptography Algorithm with Keystream Generator Modification. Advances in Science, Technology and Engineering Systems Journal, 2(5), 1-5.

Schneier, B. (1996). Applied Cryptography-Protocols, Algorithms and Source Code in C. John Wiley & Sounds, Inc, New York, Second Edition.

Devaney, R. (1989). An Introduction to Chaotic Dynamical Systems, 2nd ed. Redwood City, CA: Addison-Wesley.

Devaney, R. (1992). A First Course In Chaotic Dynamical Systems. Now published by Westview Press.

Pareek, N K, Patidar V and Sud, K K. (2006). Image encryption using chaotic logistic map. Science Direct, Image and Vision Computing, 24(9), 926-934.

Ali-Pacha, A, Hadj-Said, N, M'Hamed A and Belgoraf A. (2007). Lorenz's attractor applied to the stream cipher (Ali-Pacha generator). Chaos, Solitons & Fractals, 33(5), 1762-1766. 2007.

Chen, G, Mao, Y, Chui, C K. (2004). A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons and Fractals 21, 749–761.