International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 10 No. 1 Jan. 2015, pp. 305-312

http://www.ijias.issr-journals.org/

A New Algorithm for Dynamic Encryption

Youssef Harmouch and Rachid El Kouch

Dept. Mathematics, Computing and Networks,
National Institute for Post and Telecommunication,
Rabat, Morocco

ABSTRACT: Encryption is a cryptography method that makes it difficult to understand a document for any person without the decryption key. Our contribution in this paper is to introduce a new concept of encryption to secure a document by using a pair of keys. Generally the security of a cryptosystem relies on the secret encryption key but not on the algorithm. This has led many scientists and researchers to develop multiple encryption algorithms that are differentiated by their suppleness, complexities, resistance to the attacks and their quick execution. Thus, we develop a new dynamic encryption algorithm based on the use of two keys, static and dynamic, in different sizes. The static key is combined with a vector product to encrypted data while the other dynamic key will be added as an additional jamming, which is only a chaotic vector that changes values after each use. Our immaculate algorithm will be tested and simulated in MATLAB to visualize and verify the results so that to deduce its effectiveness of resistance to cryptanalysis attacks.

KEYWORDS: key encryption, jamming update, dynamic algorithm and cryptanalysis attacks.

1 INTRODUCTION

After the huge universal evolution of computer, phone or telecommunications networks in recent years, accesses from many implemented applications and that causes a big data exchange.

The need for confidentiality will appear under a new name “Cryptography” [1], which major function is taking clear data (example text, any kind of many areas such as “banking, military, health, education, government ...” are working on a platform with multiple multimedia document...) and turning it into unintelligible data through an encryption process, or vice versa for decryption. This process is composed of two essential elements which are the algorithm and the key.

The algorithm is a transformation applied to data for encryption or decryption using a key which is the secret parameter for this transformation with only the “transmitter, receiver” possessing information.

2 CRYPTOGRAPHY

2.1 USUAL ENCRYPTION METHOD

In cryptography domain, there are two encryption methods:

- Symmetric encryption: the same key is used to encrypt and decrypt data.
- Asymmetric encryption: different keys are used to encrypt and decrypt data.

In this paper, we look at the development of a new symmetric encryption algorithm, which uses a pair of keys with different sizes for dynamic encryption.

Corresponding Author: Youssef Harmouch

2.2 CIPHER BLOCK AND STREAM CIPHER

In general, data D is divided into bits fixed length blocks $D = D_1D_2 \dots D_N$. Each block D_i is encrypted into C_i by a function named E_K as follow $C_i = E_k (D_i)$ and the result is added to the cipher text $C = C_1C_2 \dots C_N$. There are two main types of encryption: block ciphers and stream ciphers [2].

In block cipher, the transformation function $E_k (D) = C$ is the same for each block which requires little memory and it is relatively fast for execution. In the stream cipher, the blocks are encrypted sequentially and each block is encrypted by a separate transformation that depends on:

- previous encrypted blocks, and / or
- the previous transformation, and / or
- the number of blocks

Therefore the same data D will not give the same cipher text C [3]. Our algorithm combines these two modes of encryption [4]. It will initially use the same transformation function with a static key, and then comes the stream cipher represented by using the dynamic key.

2.3 CHAOS CIPHER

For scientists [5], chaos does not mean the absence of order, but it relates preferably to a notion of unpredictability and inability to predict a long-term evolution since the final state depends directly and very sensitive to the initial state. Chaotic systems are a special class of nonlinear systems, so it is possible to apply all methods for nonlinear systems. [6]

3 ALGORITHM BLOCK

Let set ķ as a classes of equivalence for congruence modulo p noted $(\mathbb{Z}/p\mathbb{Z})$ and E_p is a commutative ring with the addition "+" and modular multiplication "×" laws $(\mathbb{Z}/p\mathbb{Z},+,\times)$ define by [7]:

$\forall$ v, u $\in (\mathbb{Z}/p\mathbb{Z})^n$ we have:

- Law "+" is defined by:
$$w = u + v = ((u_1+v_1)[p], (u_2+v_2)[p],\dots, (u_n+v_n)[p]) \quad (1)$$
- Law "×" is defined by:
$$w = u \times v^t = \Sigma(u_i \times v_i)[p] \quad (2)$$

With v^t is the v transposed.

3.1 CHAOS ADDER FUNCTION

We define J as a mapping from $\mathbb{R}^n$ into $\mathbb{R}^n$ by:

$$J : \mathbb{R}^n \rightarrow \mathbb{R}^n$$
$$\forall\, x,y \in \mathbb{R}^n \;\; J(x,y) = (x_1 \oplus y_1, x_2 \oplus y_2,\dots, x_n \oplus y_n) \quad (3)$$

This function will be used for stream cipher with a dynamic key.

3.2 ENCRYPTION FUNCTION

We define f as a mapping from $(\mathbb{Z}/p\mathbb{Z})^n \times (\mathbb{Z}/p\mathbb{Z})^{n\times m}$ into $(\mathbb{Z}/p\mathbb{Z})^m$, allowing not only the encryption of the data value xi with a static key and a pseudo-random jamming (dynamic key), but also to mixing it with subsequent values by a row vector transformation as an upper triangular matrix caused by data right shift:

$$f : (\mathbb{Z}/p\mathbb{Z})^n \times (\mathbb{Z}/p\mathbb{Z})^{n\times m} \rightarrow (\mathbb{Z}/p\mathbb{Z})^n$$
$$Y = f(x) = J(C \times X, P) \quad (4)$$

C : row vector of n dimension, present the static encryption key.

X : n×m matrix dimension containing data signal x (x_0, x_1, x_2, ... x_{n-1}) arranged by a shift right in each line, it is given as follows:

$$X = \begin{pmatrix} x_0 & x_1 & x_2 & \dots & x_{n-1} & \dots & x_{m-2} & x_{m-1} \\ 0 & x_0 & x_1 & \dots & x_{n-2} & \dots & x_{m-3} & x_{m-2} \\ 0 & 0 & x_0 & \dots & x_{n-3} & \dots & x_{m-4} & x_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & x_0 & \dots & x_{m-n-2} & x_{m-n-1} \end{pmatrix}$$

P: row vector of m dimensions m ≥ n, present the dynamic encryption key and also the chaos jamming vector.

Note:

From now on, in this article, we will consider C as the static key and call it "Key" and P as dynamic key and call it "jamming".

4 ALGORITHM OPERATION PRINCIPLE

4.1 DECRYPTION FUNCTION

After jamming subtracting by the same function J, decryption becomes purely a linear problem to solve, it is represented as a Gaussian system like this:

$$\begin{cases} x_0 = \{c_0^{-1} \times (y_0 - \sum_{i=1}^{n-1} c_i)\}[p] \\ x_1 = \{c_0^{-1} \times (y_1 - \sum_{i=2}^{n-1} c_i - x_0 \times c_1)\}[p] \\ x_2 = \{c_0^{-1} \times (y_2 - \sum_{i=3}^{n-1} c_i - \sum_{i=1}^{0} x_i \times c_{2-i})\}[p] \\ x_3 = \{c_0^{-1} \times (y_3 - \sum_{i=4}^{n-1} c_i - \sum_{i=2}^{0} x_i \times c_{3-i})\}[p] \\ \vdots \\ x_{n-1} = \{c_0^{-1} \times (y_{n-1} - \sum_{i=n-2}^{0} x_i \times c_{n-1-i})\}[p] \\ \vdots \\ x_{m-2} = \{c_0^{-1} \times (y_{m-2} - \sum_{i=m-3}^{0} x_i \times c_{m-2-i})\}[p] \\ x_{m-1} = \{c_0^{-1} \times (y_{m-1} - \sum_{i=m-2}^{0} x_i \times c_{m-1-i})\}[p] \end{cases} \quad (5)$$

x_i represents bits respectively bytes of data decrypted from y_i, which are bits respectively bytes of data encrypted, and the ci are in reality the values of the static key C.

4.2 MODULAR INVERSE

It is shown from the decryption system that the first value of C c_0 must admit a modular inverse. To correct this problem a method of adding the initial value c_0 to initial data x using the adder chaos function J (x, c_0) will be applied before the construction of the data matrix X, subsequently this value c0 will be replaced by the number '1' (since $c_0^{-1} = c_0 = 1$) throughout the course of encryption.

4.3 DATA ENCRYPTED REARRANGEMENT

4.3.1 THE PROBLEM

In decryption, each xi depends on the values x_{i-l} with $0 < l \leq i$, i.e. x_{i-l} becomes an essential factor to decrypt x_i and so recursively x_0 is the main key to all data decryption x, hence the need to rearrange the encrypted vector Y in order to clear the risk for decryption by a third person other than the transmitter and receiver.

4.3.2 PERMUTATION FUNCTION

We define $\mathcal{G}$ as a mapping from $\mathbb{R}^m$ into $\mathbb{R}^m$ allowing the permutation of $f(x)$ values by a vector $V \in (\mathbb{Z}/m\mathbb{Z})^m$ containing distinguished and unique values defined by:

$$\mathcal{G} : \mathbb{R}^m \to \mathbb{R}^m$$
$$\forall\, x \in \mathbb{R}^m, \forall V \in (\mathbb{Z}/m\mathbb{Z})^m$$
$$\mathcal{G}(x,V) = (x(V_i)),\ \forall\, i \in [0, m-1] \quad (6)$$

4.3.3 PERMUTATION INVERSE

Since $\mathcal{G}$ is reversible by itself, its inverse function is defined as mapping from $\mathbb{R}^m$ into $\mathbb{R}^m$ by :

$$\begin{gathered}\mathcal{G}: \mathbb{R}^m \rightarrow \mathbb{R}^m \\ \forall\, x \in \mathbb{R}^m, \forall U \in (\mathbb{Z}/m\mathbb{Z})^m \\ \mathcal{G}(x,U) = (x(U_i)),\ \forall\, i \in [0,m\text{-}1] \\ \text{with } \mathcal{G}(V,U) = (V(U_i)) = (0,1,2,3,\ldots,\text{m-3, m-2, m-1})\end{gathered} \tag{7}$$

4.3.4 ENCRYPTION FUNCTION

The encryption function $\hbar$ is defined then as mapping from E_p into E_p by:

$$\hbar: E_p \rightarrow E_p$$

$$\hbar(x) = \mathcal{G}(f(\mathcal{J}(x, c_0))) \tag{8}$$

This function respects the **Shannon** theorem (that says, to be a good encryption system, it is necessary that the size of the key is large enough than the size of the data) because the dimension (C) + dimension (P)> dimension (x).

4.3.5 COMPLEXITY

We can realize encryption system by successive blocks, where each block contains a predefined function, as follows:

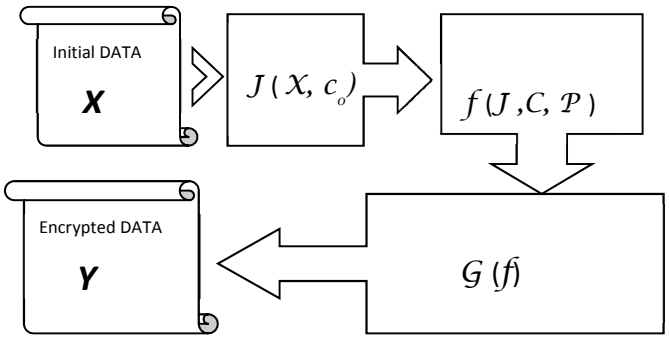

Fig. 1. Algorithm Block function

Each one of these blocks admits deferential complexity compared to the other:

- f has a matrix product, so its complexity order is an O (n×m).
- $\mathcal{G}$ is a permutation function, therefore it admits a complexity order of O (m).
- $\mathcal{J}$ has a "XOR" function only, so it has a complexity order of O (m).

5 SIMULATION

To visualize the algorithm behavior under different types of data "text, voice and image", we start a MATLAB simulation with a same given key and jamming for those data, with modulo = 256:

- Key : dimension=10

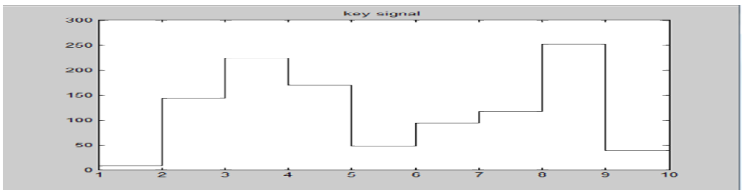

Fig. 2. Encryption Key "C" value

- Jamming : dimension=20

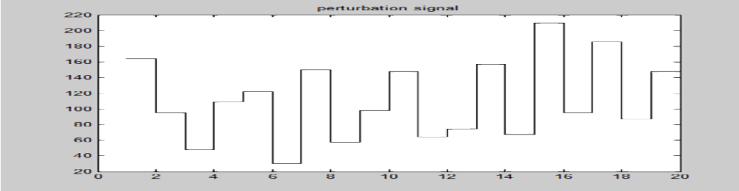

Fig. 3. Jamming vector "P" value

Note:

In the rest of simulation, the size of the jamming P will be set as 20 and the key C as 10.

5.1 TEXT ENCRYPTION

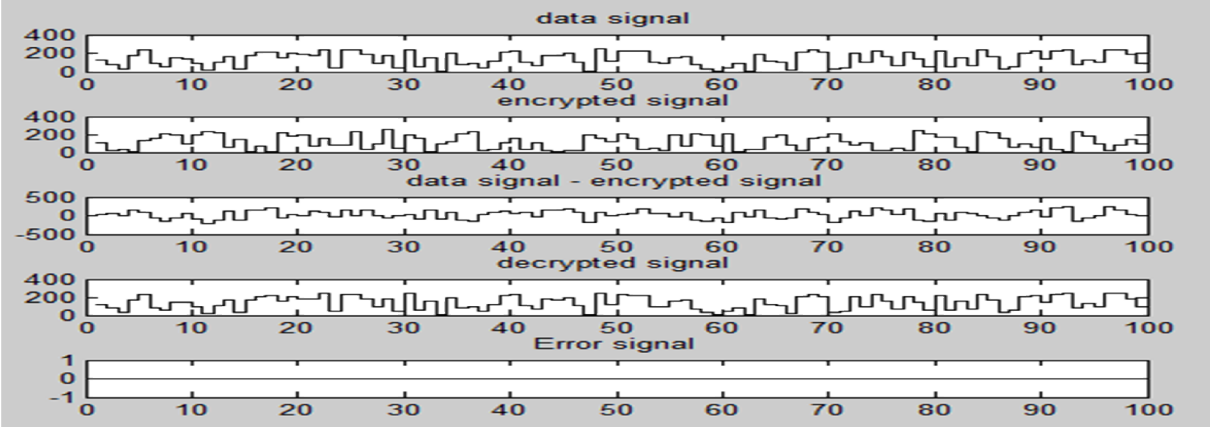

Fig. 4. Text Encryption simulation

This figure simulate a random text signal encoded with "AINSI" presented by 256 modulus data vector, the success encryption are verify firstly by subtracting the original signal with the encrypted one "data signal - encrypted signal" which must be non-zero vector, and secondly by checking the error signal which is the difference between the original signal and the decrypted one define by "Error signal = signal data - decrypted signal" which must be a null signal.

5.2 VOICE ENCRYPTION

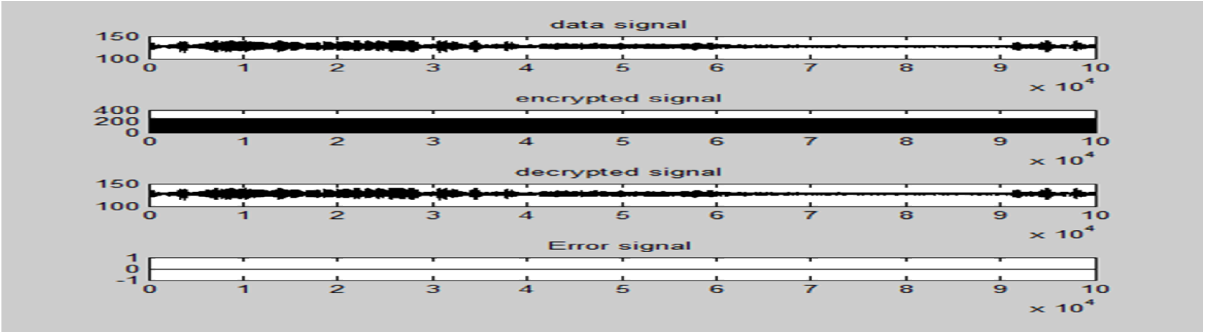

Fig. 5. Voice Encryption simulation

According to an audio file recorded by using the MATLAB function "wavrecord" with 105 of 8-bit sample (sample value are between 0 and 255) and 8000 Hz sampling frequency, this simulation show us a strong signal voice encryption that cause

a major difficulty to reconstruct the original signal from the encrypted one without knowing the initial encryption key value which changes dynamically after each use.

5.3 IMAGE ENCRYPTION

Fig. 6. Image Encryption simulation

This figure illustrates the encryption of a GIF image. From the simulation, we can easily see a strong image encryption.

Those simulation of different multimedia type "text, voice, image" show us that our algorithm can support all data type for encryption and so deduce it flexibility and strangeness to secure all type of information. This kind of Characteristics will help for single implementation to the algorithm to multiple services types of communications and applications for multiple users.

6 CRYPTANALYSIS ATTACKS RESISTANCE

6.1 RESISTANCE TO BRUTE FORCE ATTACK

Since the algorithm works in a vector space E_p into E_p, it parameters "Key and jamming" contain numbers modulo p independent of each other, the entire probability to break the algorithm by brute force attack is the product of its different functions probabilities, then we have:

$$\left.\begin{array}{l} Prob(x_i) = 1/p \\ Prob(C) = P_C = 1/(n!\, p^n) \\ Prob(P) = P_P = 1/(m!\, p^m) \\ Prob(Réarangement) = P_R = 1/m! \end{array}\right\} \Rightarrow \begin{array}{l} P_{total} = P_C \times P_P \times P_R \\ P_{total} = 1/(n!\, m!^2\, p^{n+m}) \end{array} \quad (10)$$

For example, for AINSI encoding and a key dimension respectively jamming is 10 respectively 1024:

$$\left.\begin{array}{l} p = 256 \\ n = 10 \\ m = 1024 \end{array}\right\} \Rightarrow P_{total} \cong 7{,}1178 \times 10^{-7777}$$

To a computer (CPU: Intel Core 2 Duo 2GHz, 2GB RAM), the execution time for encryption a 1MB of data is 2.22 s, which would say that for such a probability, the time to break the algorithm with 365 days/year is almost equal to $9{,}891 \times 10^{7768}$ years.

6.2 RESISTANCE TO ATTACK PLAINTEXT / ENCRYPTED

The permanent change in the jamming makes a strong resistant for the algorithm to attacks types:

- attack using statistical analysis.
- attack using encrypted text only.
- attack using clear text.
- attack with the chosen plaintext.

The following figure shows an average curve of each new jamming vector for a 1000 bytes data with jamming dimension equal 20:

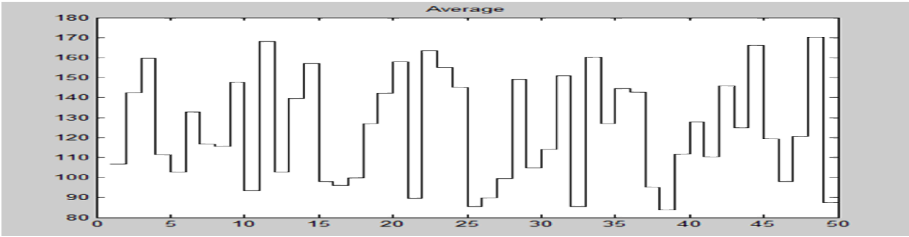

Fig. 7. Average jamming vectors values

Now let's examine the behavior of the algorithm to a constant unit function as initial data to be encrypted, ie:

$$x(t)=\text{cte}\times u(t) \tag{11}$$

Example:

$\text{Cte} = 100 \Rightarrow x(t) = 100 \ \forall \ t \geq 0$

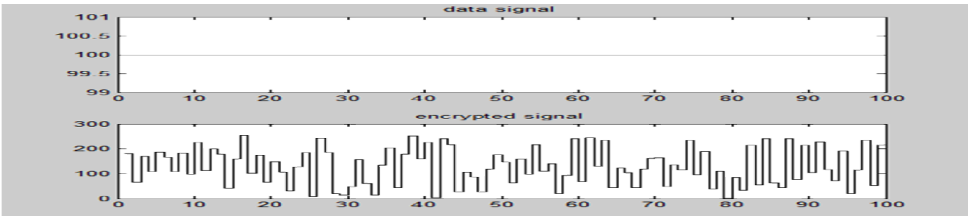

Fig. 8. Constant data text value encryption

There is a lack of a cyclic redundancy check (message dimension = 100), making it difficult to break the algorithm with those types of attacks.

7 ALGORITHM OPERATION METHOD

The algorithm is based on a secret shared only between the sender and receiver. This secret is a data vector that the couple "transmitter, receiver" must have in initialization phase, its allows : the receiver to get information about the jamming added to the data by the sender, the sender authenticates , the encrypted data sign, the encrypted data rearranges, verify the correctly data transmission..., in this paper we are interested only in jamming synchronization and its update between "transmitter, receiver", the rest of functions and the possibilities offered by this "secret" with exchange method for "key + secret" will be addressed in the development phase of a new protocol using this algorithm in the next paper.

During initialization phase, the transmitter and receiver synchronize the "secret +key" (key and secret exchange method), the sender encrypts the secret then creates and adds a chaotic random jamming into, rearranges secret encrypted data and then sends it to the receiver. The receiver rearranges inversely this data and calculates the deference between the received secret data from the sender and his encrypted secret data by itself with only the static key C, in order to infer the jamming P.

After each new data encryption, a modulus value is added from the secret vector to the jamming vector in order to modify it and update it for the next use, ie:

```
i=0
Repeat until the end of data:
        Message encryption "key + jamming"
        For (k=0 ; k<m ; k++)
                Jamming_k = (Jamming_k + secret_i )[p]
        end for
        i = i+1
        if( i > m-1)
                then i=0
        end if
end repeat.
```

8 CONCLUSION AND FUTURE WORK

After using a pair of keys for encryption, a static and another one dynamic which changes its values after each use due to a secret vector data shared only between transmitter and receiver and after the results obtained during the simulation, we deduce that our algorithm is clearly much stronger in resistance to cryptanalysis attacks. This shared secret gives us many procedures such as authentication, confidentiality, signature, that could be put it into a new protocol inheriting the present algorithm. This protocol will describe the exchange methods of the couple "keys and secrets" for the initialization phase and operate in multiple functions as an error detection, security policy and tolerance rate…

Just as the high resistance to cryptanalysis attacks by our algorithm, this protocol will be dedicated to resist to different types of network attacks such as Denial of Service (DoS), Man in the Middle, package network ejection..., to finally get a powerful security protocol when it is used in an unsecured network or platform.

REFERENCES

[1] A. Young, "the future of cryptography: practice and theory", pp. 64, 62 – 63, 2003.

[2] S. Orn er Sharif, S.P. Mansoor, "performance analysis of stream and block cipher algorithms", vol. 1, pp. 522 -525, 2010.

[3] "Neuro-applied cryptography" by S. Dourlens, Paris University 8 Micro Computer Micro-Electronics Departement, ch. 3,p. 4, 1996.

[4] D.E. Goumidi , F. Hachouf "Hybrid chaos-based image encryption approach using block and stream ciphers" Vol. 1,pp. 522 – 525, 2010.

[5] X. WANG, L. MA, X. DU, "an encryption method baser on dual-chaos system" pp. 217 – 220, 2009.

[6] "realization and study of a secure system based on chaos " by M. Ouerdia Mouloud Mammeri, Tizi-Ouzou University, Electrical Engineering and Computer Faculty, Automatic Control Science Department, ch.1, p. 3, 2013.

[7] S. Vollala, Varadhan V, K.Geetha, N.Ramasubramanian, "efficient modular multiplication algorithms for public key cryptography"pp. 74 – 78, 2014.