CRYPTOGRAPHY CHAOS THEORY

Bulai Rodica and Victor Fanari[1]

DOI: 10.24989/ocg.v331.37

Abstract
The development of information society, which has led to an impressive increase in the volume of information, mainly economic, circulated in computer networks, accelerated the development and mostly the use of modern cryptography tools. In the last years, researchers have pointed out that there is a possible similarity between chaos and cryptography, many of the properties of chaotic dynamic systems having correlation among the cryptographic systems that are based on computational methods.

Studies carried out on chaotic dynamic systems usage in digital crypto-systems have determined the occurrence of similar to classic techniques, but also of some specific techniques and methods that have been analyzed and evaluated. The attempts to develop new encryption algorithms based on chaos theory have evolved gradually from simple solutions, which suppose the iteration of a dinamic system to obtain binary sequence used for text masking, to methods that imply coupled dinamic systems and hybrid techniques that would combine the chaos advantages with classical methods.

In this article there are presented 3 encryption algorithms based on chaos theory: RC4, Fractal Encryption and Cellular Automata, implemented in a system of encryption and operation mode analysis for each algorithm separately.

1. Introduction

The theory of chaos is one of the ways we can study nonlinear phenomena. More specifically, chaos is a state of nonlinear dynamic systems in which seemingly random events are actually predictable using simple deterministic equations. Thus, a phenomenon that seems unpredictable locally can actually be stable globally, can have well-defined boundaries and may have sensitivity to initial conditions. Small differences in the initial states can produce significant differences over time in the final states.

The theory of chaos teaches us that even very simple rules can lead to extremely complex and unpredictable behavior. Water droplet dispersion from a dripping tap is not the same if it occurs twice, even if each drop is almost exactly the duplicate of the last one. Changes in the microscopic environment have a dramatic effect on individual particle pathways in water.

Optimization problems in uncertain and dynamic environments are complex and difficult, and often classical algorithms based on dynamic programming or mathematical approaches manage to solve only small instances of problems. Attempts to develop new cryptographic algorithms based on chaos theory evolved gradually from simple solutions involving the iteration of a dynamic system to

[1] Technical University of Moldova, Faculty of Computers, Informatics and Microelectronics, Str. Studentilor 7, Chisinau, MD-2012, Republic of Moldova, Tel:(37322) 509908; E-mail:rodica.bulai@ati.utm.md, fanarivictor@mail.ru

obtain the binary sequence used to mask the text to methods involving coupled dynamic systems and hybrid techniques combining the chaos advantages with classical methods.

The design and digital implementation of chaotic cryptosystems involves the use of digital chaotic functions for building flow or block algorithms.

From a technical point of view, the term "chaos" defines a particular state of a system characterized by the following:

- is never repeated (looks irregular);
- there is a dependence of sensitivity in relation to the initial conditions: extremely small differences in the values of different parameters may lead to divergent results;
- it is less ordered and can be characterized by unpredictable determinism .

Unpredictable determinism means that even a perfect chaotic system (identical motion equations and the same initial conditions) can lead to unpredictable outcomes [1].

Chaotic systems are therefore ordered, deterministic and unpredictable. It is true that "very simple" systems follow perfectly deterministic rules and yet their behavior is totally unpredictable. Deterministic, because the effects can be precisely measured and located, determining the continuation of events. Chaos, because we do not know everything that will happen, despite the fact that we know all the data that determines the events.

2. The principle of cryptography based on the theory of chaos

The principles of "chaos theory" are used to secret communications. The basic idea is that a message can be "buried" inside a chaotic signal - a sound of solar, meteorological origin, and so on. - as a screen that makes the message inaccessible to those who can not break down chaos into component elements.

Chaos-based cryptosystems use deterministic chaotic dynamic systems, either continuous or discreet, sensitive to the initial conditions. By their motion law, these dynamic systems uniquely determine the state of the cryptosystem and allow for non-catastrophic decoding of the encoded sequence. These dynamic systems are described by state functional equations (formula 1) in which a linear or nonlinear 'dynamic' ***f*** function of the system is used:

$$x+(t) = f(x(t),t) \qquad (1)$$

By ***x***, we understand the state variables vector dependent on the continuous time variable ***t***, and + represents the system state change operator.

Similarly, for discrete systems (formula 2), the state equation is written according to the discreet variable of time ***n*** in form:

$$x[n+1] = f(x[n],n) \qquad (2)$$

It is preferred to use non-linear dynamic systems that may have more than one set of boundaries in a permanent regime, with different attraction bases, very dependent on the initial condition, so that long-term prediction of their condition becomes impossible.

In the case of mixed discrete chaotic cryptosystems, the encryption and decryption procedure is performed by multiple, inverse and direct iterations, and the encoded sequence corresponds to the number of iterations performed. These systems prove to be particularly robust against statistical attacks.

The principle of cryptography based on the theory of chaos is given by the diffusion and confusion of trajectory parameters generated on the basis of the encryption key and the transmitted message. With small variations of the transmission key, extreme changes of the phase path trajectory for the dynamic system used must occur. This ensures the cryptosystem resistance against raw attacks based on the testing of all possible transmission keys.

Chaotic trajectories are neither periodic nor quasi-periodic, and they have a random appearance with a "white noise" (wideband) power spectrum [2].

No computer or software can predict the trajectory of a chaotic dynamic system, because the algorithmic complexity of the trajectories is positive, given by the Kotulski-Szczepanski entropy of the system. This is based on the idea of designing efficient data encryption techniques based on the theory of chaos so that the entropy of the system grows through coding and exceeds the computational capabilities of the cryptanalyst.

Optimization of encryption algorithms aims at reducing data processing time, reducing memory capacity, diversifying potential transmission keys, and decreasing the efficiency of cryptographic attacks. Applying a precision to compress the information source to reduce its redundancy reduces the risk of interception of the transmission key and the effectiveness of any attack.

The value of a cryptosystem is appreciated on the basis of several factors: degree of secretion, encryption key size, error propagation, uniqueness distance.

Developing a powerful cryptosystem involves maximizing the amount of work required for cryptanalysis by any method. When the cryptanalysis of an encryption algorithm is performed, the general assumption is that the cryptanalyst knows exactly how the cryptosystem works.

3. Ciphering and deciphering methods of chaotic cryptographic systems

There are two ways to use chaos to encrypt information.

The add-on method consists of separately creating the chaotic system and the information and then adding the two signals. In turn, the interlocutor has the system keys (initial conditions and system equations sent in advance) with which he can in turn create a chaotic system like the one from the broadcast. When he receives the additional message of the chaotic system, he has nothing to do but recover the message by extracting the "mask".

The inclusion method not only drowns in the chaos of the message, it is even deep inside the structure of the chaotic system, still a hindrance for a spy in his attempt to decipher the message.

The message is therefore not transported by the chaotic wearer through the transmission line, but it is its own bearer. This line only contains the transmitter data that will allow the recipient to discover the transmitted message.

The difference between the addition and inclusion methods is that in the latter case the message is not "retrieved" in the receiver but is "reconstituted" by the receiver.

Attempts to develop new cryptographic algorithms based on chaos theory evolved gradually from simple solutions involving the iteration of a dynamic system to obtain the binary sequence used to mask the text to methods involving coupled dynamic systems and hybrid techniques combining the chaos advantages with classical methods.

The most promising encryption systems based on chaotic dynamic systems have proven to be the ones using linear functions on portions. The discrete representation of the chaotic system values can lead to the loss of intrinsic properties of the continuous dynamic systems, appearing problems related to the dynamic degradation of the behavior of digital chaotic functions [3].

By using simple disruptive methods, good performance can be achieved for the chaotic digital functions used to implement random (pseudo) sequence generators, but also for building encryption algorithms.

The development and implementation of an encryption algorithm pursues aspects related to the provision of chaotic features throughout the entire system operation period. For this purpose, rules were used to define the dynamic system's initial parameter and condition to fully exploit the key's size and to ensure sensitivity to its modification. Obtaining a real-time workflow for real-time applications is determined both by the implementation mode used for chaotic dynamic systems and by the way the algorithm is defined, which is why it has been proposed to use a number of fixed iterations of small size, but determined by the key, through a very sensitive relationship to changes [4].

The approach to many variants of chaos-based encryption algorithms has so far been limited to software, primarily due to ease of use, enhancement, portability and flexibility. But with technological development and increased demands on high-speed work and key safety, hardware implementations become more suited both in terms of physical security and encryption / decryption speed.

By implementing chaotic generators and encryption system, it has been demonstrated that digital hardware structures can be used with good performance to protect information using chaos- specific techniques.

4. Analyzed cryptographic systems based on the theory of chaos

Three algorithms based on chaos theory have been selected and analyzed: RC4 (Rivest Cipher 4), Cellular Automata and Fractal Algorithm, which are part of a stream cipher system and have been integrated into a single encryption entity and Decryption.

4.1. The *RC4* algorithm

Rivest Cipher 4 is a flow cipher. While it's remarkable for its simplicity and speed in software, more vulnerabilities have been discovered, making it unsafe. RC4 is the most commonly used cipher-

stream software in protocols such as SSL or WEP. Unfortunately, RC4 does not meet the current high security standards and some methods of using it lead to very unsafe cryptosystems.

The cipher consists of two major components, the Key Scheduling Algorithm (KSA) and the Pseudorandom Generation Algorithm (PRGA). The internal state of RC4 contains a permutation of all 8-bit words, i.e., a permutation of N = 2^8 = 256 bytes, and the KSA produces the initial pseudorandom permutation of RC4 by scrambling an identity permutation using the secret key k. The secret key k of RC4 is of length typically between 5 to 32 bytes, which generates the expanded key K of length N =256 bytes by simple repetition. If the length of the secret key k is l bytes (typically $5 \leq l \leq 32$), then the expanded key K is constructed as $K[i] = k[i \bmod l]$ for $0 \leq i \leq N-1$. The initial permutation produced by the KSA acts as an input to the next procedure PRGA that generates the keystream (Figure 1).

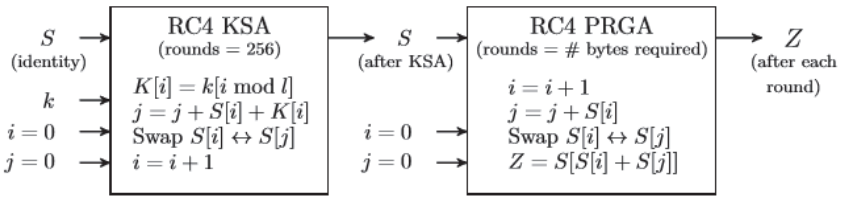

Figure 1: Description of RC4 stream cipher [5]

For round r = 1,2,... of RC4 PRGA, we denote the indices by i_r,j_r, the keystream output byte by Z_r, the output byte-extraction index as $t_r = S_r[i_r] + S_r[j_r]$, and the permutations before and after the swap by S_{r-1} and S_r respectively. After r rounds of KSA, we denote the state variables by adding a superscript K to each variable. By S_0^K and S_0, we denote the initial permutations before KSA and PRGA respectively. The S_0^K 0 is the identity permutation and $S_0 = S_N^K$ is the permutation obtained right after the completion of KSA[5].

4.2. The *Cellular Automata* algorithm

Proposed by John Conway, as "The Game of Life", played on a grid, divided into cells. Each cell can be "live" or "dead," and a set of four rules determines whether any given cell will live, die or be born at each iteration. The simple set of rules of the game has led to surprisingly complex and convincing behavior, and a new field of research called "Cellular Automata" has emerged around.

One interesting point that can be extracted from this area is that any simulation of cellular automates, no matter how complex they are, is completely determined by the state of starting the cells on the grid.

The *Cellular Automaton* algorithm is currently not widely deployed, such as the *RC4* algorithm. Most often it is used in image encryption, as it provides the user with a variety of encryption methods. There are some useful aspects to this: from a single configuration of cells in a grid, a huge volume of unpredictable and complex information can be built. After a thousand generations, who could predict which cells would be active without knowing the initial state and without running the whole simulation? If a single cell was different in the initial configuration, after sufficient generations, the condition of each cell will ultimately be different (Figure 2).

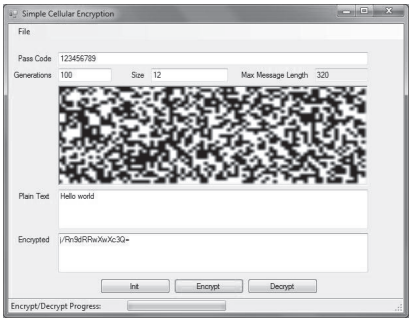

Figure 2: Automated Cellular Algorithm

Generating pseudo-random numbers through Cellular Automats.

Cellular automata are dynamic systems where space and time are discreet. A CA (Cellular Automated) consists of a series of cells, each of which can be in one of a finite number of possible states, updated in a discrete time synchronously, according to a local and identical rule.

Consider only Boolean automata for which the cellular state is $s \in \{0, 1\}$.

The condition of a cell at the next time step is determined by the current state of a neighborhood around the cells. Cellular matrix (grid) is d-dimensional, where d = 1, 2, 3, which are used in practice. In our case d = 1 and d = 2, that is a one- and two-dimensional grid. The same rule contained in each cell is essentially a finite state, usually specified as a rule table (also known as the transition function), with an entry for each possible neighborhood of configurations. Neighboring cell of a cell is formed by its own state and neighboring cells (adjacent) [6].

For unidimensional CA, a cell is connected to a local neighbor r (cell) on each side, where r is called radius (so each cell has 2r + 1 neighbors). For two-dimensional CA, two types of cellular districts are usually considered: 5 cells, consisting of its own cell together with the four non-Dionygon evecs (also known as von Neumann neighborhood) and 9 cells, consisting of its own cell with The eight surrounding neighbors (also known as the Moore neighborhood). When a finite dimensional grid is analyzed, periodic spatial conditions are frequently applied, resulting in a circular grid for the unidimensional case, and the toroidal grid for the two-dimensional case.

S. Wolfram first proposed CA one-dimensional as a pseudo-random number generation (PRNG). In particular, he extensively studied the bit sequences generated by rule 30 in his numbering scheme for unidimensional, r = 1 rules, if the rule number represents the decimal format of the binary coding number in the rule table.

In Boolean format, Rule 30 can be written as in Formula 3:

$$s_i(t + 1) = s_{i-1}(t) \text{ XOR } (si(t) \text{ OR } s_{i+1}(t)) \qquad (3)$$

where s_i (t) is the state of the cell *i* at time *t*. The formula gives the state of the cell *i* in time step *t* + *1* as a boolean function of the states in the vicinity of the cells at time *t*. Pseudo-random bit sequences are obtained by sampling the values that a particular cell (usually the central one) touches as a function of time [7].

An uneven randomizer was presented consisting of two rules, 90 and 150, arranged in a specific order in the grid (Figure 3).

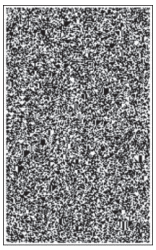

Figure 3: Random Number Generator uneven unidimensional

In the boolean form, rule 90 can be written as:

$$s_i(t + 1) = s_{i-1}(t) \text{ XOR } s_{i+1}(t) \quad (4)$$

and rule 150 can be written as:

$$s_i(t + 1) = s_{i-1}(t) \text{ XOR } s_i(t) \text{ XOR } s_{i+1}(t) \quad (5)$$

Generating a string of numbers based on cellular automata.

Let P be a clear message and E is a cipher algorithm. Fundamental transformation to get C-chipherd text is therefore:

$$C = E_k (P) \quad (6)$$

where k is the key of transformation that distinguishes a particular encryption in a transformation family using the same decryption algorithm. To recover the original message, a D_k decryption function using the same key is defined as the inverse of E:

$$P = D_k(C) = D_k(E_k(P)) \quad (7)$$

Encryption algorithms that operate with clear text on a single bit at a time are called flux cipher algorithms. A flow cipher breaks the P message into a bit stream or successive bytes $p_1, p_2, ..., p_q$ and encrypts each pi with a bit stream (or bytes) $k_1, k_2, ..., k_q$ generated by a key generator so that:

$$E_k (P) = E_{K1} (p_1) E_{K2} (p_2) \quad (8)$$

A common encryption operation used is the XOR-exclusive operation:

$$c_i = k_i \text{ XOR } p_i \quad (9)$$

where c_i is the i-th bit of the cipher text. Applying the same operation on cipher text allows recovery of the original text:

$$p_i = c_i \text{ XOR } k_i = (k_i \text{ XOR } p_i) \text{ XOR } k_i \quad (10)$$

4.3. The *Fractal* algorithm

The *Fractal* Algorithm uses the famous *Mandelbrot* fractal to convert the encryption key (provided by the user) to a longer key, which is then XORed with the clear text, resulting in encrypted text.

Many famous encryption algorithms extend to some extent the encryption key and then, after moving, move and replace the bits in plain text, they use the XOR operation with the extended password, and this process is usually repeated a number of times.

The *Fractal* Algorithm tries to create a random key extended using the Mandelbrot fractal instead of using a fixed rule [8].

Moreover, the *Fractal* algorithm encrypts the entire file as a single large block instead of encrypting it divided into blocks of 256 bits, so it does not use the same encryption key on each block but uses only one large encryption key to It encrypts the entire text (which should mean fewer repetitions - fewer chances of attacking successfully).

Although it is more complex than the other two algorithms, *Fractal* is used in encryption of visual images, and, at the same time, the encryption information using its iterating.

Principles of Fractal Encryption Algorithm.

Encryption is the repetition of binary operations inside a loop, between which the fractal encryption key is calculated [9].

Suppose we have a series of messages *M(j)* for $j = 1$ up to N, we want to send safely to the recipient. We will need a reversible encryption function *E*:

$$E(M(j), k) \dashrightarrow X(j) \quad (11)$$

where *k* is an encryption key and *X(j)* is the properly encrypted message. Then the message is sent to our receiver, which has a complementary function E' to decrypt the encrypted message:

$$E'(X(j), k) \dashrightarrow M(j) \quad (12)$$

However, both *E()* and *E'()* function can not be performed using Fractals. On the other hand, there are some functions, such as XOR (or-exclusive) that are their own complementaries:

$$(M(j) \text{ XOR } k) \dashrightarrow X(j) \quad (13)$$

$$(X(j) \text{ XOR } k) \dashrightarrow M(j) \quad (14)$$

But *XOR* is also a weak encryption function, and although it is perfectly sure of a single message, but if we use it more than once with the same key *(k)* it becomes very easy to perform reverse engineering, thus making the operation Unsure *XOR* for single key encryption systems. This can be solved by using another key at each iteration:

$$M(j) \text{ XOR } K(j) \dashrightarrow X(j) \text{ și } X(j) \text{ XOR } K(j) \dashrightarrow M(j) \quad (15)$$

Most often we want to generate a series of identical keys on both sides: sender and receiver. But we must be able to generate a series of keys that are secure in cryptography. That is, even if an external observer knows all of the previous keys, he would not be able to predict the next key in the series with precision. And because we'll need a different set of keys each time, in fact, we need actual serial key to the basic key [10].

The solution is to use *a Master Key - MK*, and another *H*-encryption function, to generate the specific keys for each message:

$$H(MK, j) \dashrightarrow K(j);\ M(j)\ XOR\ K(j) \dashrightarrow X(j) \text{ și } H(MK, j) \dashrightarrow K(j);\ X(j)\ XOR\ K(j) \dashrightarrow M(j) \quad (16)$$

In this case fractals are used, because as we can see above, the H function does not need a complementary function *H'*. So we can freely use a basic *Fractal* function with a master key to generate the local key series [11].

5. Security and performance of algorithms usage

Unlike a modern stream cipher (such as eSTREAM), *RC4* does not take just one random number (nonce) along with the key. This means that if a single long-term key is used to safely encrypt multiple streams, the protocol must specify how to combine this arbitrary number (nonce) and the long-term key to generate the key flow for RC4. To address this operation, it is necessary to generate a "fresh" RC4 key by hashing a long-term key with a nonce. However, many applications that use RC4 simply hook the key and nonce.

Because the RC4 is a flux cipher, it is more malleable than the common block ciphers. If it is not used along with a strong message authentication (MAC) code, then encryption is vulnerable to a bit flipping attack. The method is vulnerable to a stream cipher attack if it is not implemented correctly. Moreover, the double encryption of a message with the same key may accidentally decrypt the encrypted text, since the involuntary nature of the XOR function would result in the second operation inversion of the first.

If the keyflow of the Cellular Automation algorithm is truly unpredictable, then we have the so-called «one-time pad», the system that is perfectly safe (assuming the keys are not stolen). The one-time-pad system was invented by J. Mauborgne, and is based on a variation of the Vernam cipher in which the key is not repeated. However, the encryption system is impracticable because the sender and receiver must be in possession and protect the random key. In addition, the total amount of data that can be encrypted is limited by the key length available.

Thus, the security of a flow cipher system, of the given algorithm, is based on the predictability of bits in the key stream. A good pseudo-random statistic of the key stream is not enough in cryptographic applications: a perfect RNG may be completely inappropriate if the next random bit can be predicted from the previous sequence. From this point of view, ACs are more appropriate than classical RNGs, which are very easy to break, taking into account the given algorithm and a small portion of the sequence.

By analyzing the performance of these encryption and decryption algorithms, after the required time (seconds) for encrypting a data volume (number of characters), then we can mention that the Fractal algorithm requires the smallest time, regardless of the amount of encrypted (figure 4) or decrypted (figure 5) information.

For developing these 3 algorithms we used C# language - no additional libraries - just standard functions and standard "Windows Form" compiler (used in Visual Studio). We used random characters (numbers / letters / signs). The maximal length that we used is 500.000 characters. Cause as you can see from the diagram, RC4 algorithm is taking much more time to do all the stuff in the background (almost 350 seconds).

First of all, we started with 1000 characters - and as you can see the encryption and decryption is around 0-2 seconds. After this, we started to increase the number of characters to see what is the difference between these algorithms (RC4, Cellular Automaton, and Fractal). So as you can see from the graph - Fractal algorithm is less affected by the length of the encrypted text. And for 100000-500000 characters it can take a long time to encrypt/decrypt for RC4 and Cellular Automaton. This is caused by the fact that it needs to know the value for each "neighbor". Startup time is almost the same for all these algorithms, but as we go with a multi characters text, the most efficient is Fractal of course.

All these algorithms are using the physical memory to do all the encryption/decryptions and to generate all pseudo-random numbers and they are using also the CPU to maintain a fast solution for the end user.

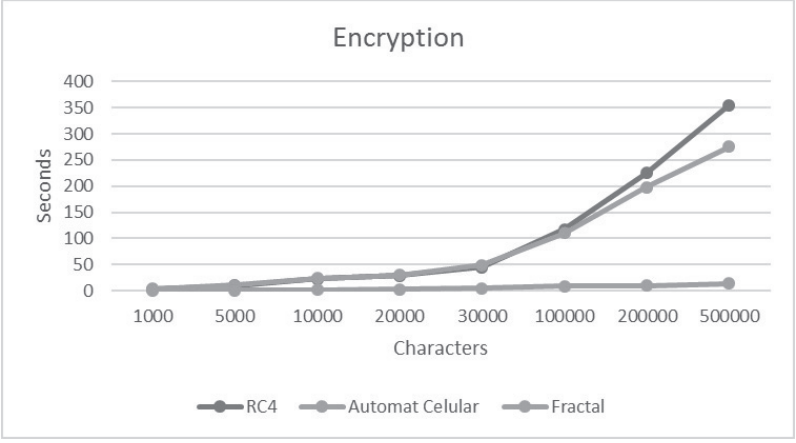

Figure 4: Encryption Performance

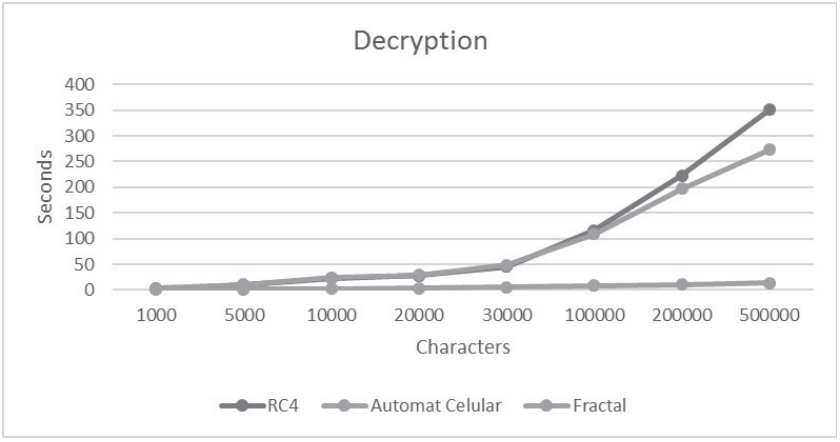

Figure 5: Decryption Performance

6. Conclusions

Studies on the use of chaotic dynamic systems in digital cryptosystems have led to the emergence of similar techniques to classical ones, but also to specific techniques, methods that have been analyzed and evaluated. Attempts to develop new cryptographic algorithms based on chaos theory evolved gradually from simple solutions involving the iteration of a dynamic system to obtain the binary sequence used to mask the text, to methods involving coupled dynamical systems and hybrid techniques combining the chaos advantages with classical methods.

The application of chaotic dynamic systems in the development of new cryptographic algorithms is in the process of development along with the technological evolution. Many of the proposed methods are still in their early stages, due to the relatively slow implementation technology and insufficient cryptographic resilience, but it should be noted that chaos can be a source that could be exploited to obtain robust cryptosystems for attacks based more and more on the high calculation ability of the new performance processors.

By enrolling in the attempts made to exploit the intrinsic characteristics of chaotic dynamic systems, cryptography based on the theory of chaos constitutes a new direction of research in the field of data protection in the last period. Studies conducted in this direction were followed by the proposal of specific solutions for the use of dynamic systems with chaotic behavior for the realization of robust and secure communication systems. Pseudo-random generators, block ciphers, and hash functions are three of the best-known security service delivery methods in which chaos-based solutions have been proposed.

The development and implementation of an encryption algorithm pursues aspects related to the provision of chaotic features throughout the system's operating period. Obtaining a working speed for real-time applications is determined by both the deployment mode used for chaotic dynamic systems and the way the algorithm is defined, which is why it has been proposed to use a number of fixed iterations of small size, but determined by the key, through a very sensitive relationship to changes.

By using simple disruptive methods, good performance can be achieved for the chaotic digital functions used to implement (pseudo) random sequence generators, but also for building encryption algorithms.

The present work, through modeling, simulation and, in particular, through the concrete implementation of digital chaos based cryptographic systems, has attempted to respond to new trends by proposing specific solutions and presenting the results obtained.

7. References

[1] MOGOLLON, M., Cryptography and Security Services: Mechanisms and Applications, New York, Cybertech Publishing, 2008, 26-27.

[2] KONHEIM, A. G., Computer Security and Cryptography, John Wiley & Sons, Inc., 2007, 99, 350.

[3] MAO, B., Современная криптография (теория и практика), М.: Вильямс, 2005.

[4] BREDIN, S., Chaos theory and Cryptography, https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxicmVkaW5jcnlwdG8yfGd4OmRkZmRlOWVhNmFkMThjYg

[5] GUPTA, S. S., Analysis and Implementation of RC4 Stream Cipher, Indian Statistical Institute, India, 2013.

[6] WOLFRAM, S., Cryptography with Cellular Automata, http://www.stephenwolfram.com/publications/academic/cryptography-cellular-automata.pdf

[7] HENRIQUES, M. A. A., New Possibilities for Cellular Automata in Cryptography, http://www.criptored.upm.es/cibsi/cibsi2011/info/Ponencias/5.%20New%20Possibilities%20for%20Cellular%20Automata%20in%20Cryptography.pdf

[8] AL-AKAIDI, M., Fractal Speech Processing, http://assets.cambridge.org/97805218/14584/frontmatter/9780521814584_frontmatter.pdf

[9] Fractal-Based Encryption, http://www.techbriefs.com/component/content/article/ntb/tech-briefs/photonics/2579

[10] GUPTA S., BANSAL, N., Image Encryption Techniques using Fractal Geometry, http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue5/Version-1/F016513135.pdf