New Symmetrical Cryptosystems Based on a Dynamic Optimization Model: two-Dimensional non-Uniform Cellular Automata

Ismahane Souici
Computer sciences department, Jijel university, Algeria
LabSTIC, Guelma university, Algeria

Hamid Seridi
LabSTIC, Guelma university, Algeria
seridihamid@yahoo.fr

Herman Akdag
LIASD, Paris 8 university, France
akdag@ai.univ-paris8.fr

Received 27 October 2014
Accepted 17 July 2015

Abstract

Today, computer networks are complex and illegal wiretapping is possible. This poses a real problem for the security during transmission of data. For ethical reasons, the transfer of sensitive data cannot be done with such a danger and must protect themselves. Protection best suited for this type of communication is the cryptography. However, despite all its developments, it is still hampered by some flaws citing in particular the key size and resistance against advanced attacks.

Thus, in this paper we present two new cellular automata-based optimization algorithms aiming to solve problems of images encryption: S2CA, Symmetrical Cellular Automata-based Ciphering Algorithm, and S2CA-EX, Symmetrical Cellular Automata-based Ciphering Algorithm Extension. The main idea is to operate on the pixels positions or pixels RGB coding positions in order to alter the original image so as to have the maximum of disorder in the corresponding encrypted image. The proposed algorithms were tested on images of different sizes where they have shown a good confusion level and resistibility against the most advanced attacks. A comparison between these algorithms and the best-known cryptographic standards conclude this work.

Keywords: Security, symmetrical crypto-systems, advanced attacks, optimization, cellular automata, neighbors.

1. Introduction

Digital networks have so much evolved that they have become an essential communication mechanism. They can transmit any kind of text, sound and mainly images information. The exponential growth of traffic images is enhanced by the emergence of large digital photo cameras and mobile phone use. Images representing two-dimensional information are specific data because of their large amount of information. Thus, image transmission raises a significant number of problems that are not yet all resolved. In addition, computer networks are complex and numerous illegal wiretapping, which continually require new ways to increase the safety and particularly safe encoding of the exchanged data. To assure these needs, cryptography is one of the safest methods.

Two basic categories in cryptographic systems are distinguished: Secret key cryptosystems and public key cryptosystems. The first ones (secret key cryptosystems, also called symmetrical cryptosystems), known for their speed, require the sharing of a secret between two communicating parties; it is the encryption key which must be kept secret. However, public key cryptosystems (also called asymmetrical cryptosystems) that are quite slow compared to the first ones present the advantage of
using a pair of public/private key used for encryption/decryption, which eliminates the problem of securing the transmission of secret (secret key). A state of the art in the principles and the most known symmetrical and asymmetrical cryptosystems was proposed in [1]. These systems were based on various techniques. One of the promising techniques uses cellular automata (CAs). These latter are simple models of highly parallel and distributed calculation wearers of complex behaviours (a state of research art on CAs and their applications is available in [2]).

On the side of asymmetrical cryptosystems, Guan [3] and Kari [4], for example, have come to use CAs to develop such systems. For symmetrical cryptosystems, Wolfram [5] was the first who succeeded in the development of a safe method using CAs. Then several other studies that fall under the same category (symmetrical cryptosystems) were established. We cite as examples the work of Habutsu et al. [6], Nandi et al. [7] and those of Gutowitz [8]. Tomassini & Perrenoud [9] and Tomassini & Sipper [10] also came to develop symmetrical cryptosystems using one-dimensional (1D) and two-dimensional (2D) automata. Similarly, in [11], the authors present a symmetrical cryptosystem based on CAs where the cryptographic Vernam encoding based was considered, and following the work of Kari [12, 13] and of Toffoli and Margolus [14] showing the undecidability of the problem trying to find out if a cellular automaton in dimension at least two is invertible, asymmetrical cryptosystems based on cellular automata have been proposed [15, 12, 16]. The confidentiality of these cryptosystems is usually based on the difficulty of finding the inverse of the encryption function. The computation of this inverse may however be very easy if one knows secret information in the construction of the encryption function. Regarding the cryptosystems based on cellular automata, the public key used to encrypt the message is a cellular automaton. To decrypt the message, the inverse automata of that used for encryption was simply applied to the encrypted message.

Recently, they were a subject of study by Bruno Martin [17], where starting from the demonstration that there is no elementary rule of non-linear cellular automata, robust to correlation, this allowed him to infer that this result strongly limit the use of cellular automata for the construction of pseudo-random sequences representing keys used in secret key cryptography. Especially for such mechanisms generating pseudo-random sequences, Meier and Staffelbach [18] proposed an effective cryptanalysis technique.

In our work, the problem of encryption is otherwise considered. We formalized it as an optimization problem having as goal the calculation of the most possible different solution (encrypted data) of the original data. It deals with the application of two-dimensional nonuniform (also called heterogeneous) cellular automata for secret key cryptography where two cryptosystems have been proposed. More details will be given in the following sections.

2. Cellular automata

2.1. History

Interested by the evolution of graphic constructions based on a two dimensional space divided into cells and generated from simple rules, the mathematician Stanislas Ulam was the creator of the notion of cellular automata [19,20]. A given combination of cells, which could have two different states: ON or OFF, is called cells configuration. The next one is calculated according to neighborhood rules. Ulam quotes that these latter representing generally a simple mechanism (such as, a cell in contact with two lit cells, will lit otherwise it will die) are able to generate very complex figures that can, in some cases, self-reproduce. In addition, it is Ulam who was suggested to John Von Neumann to use what he called "cellular spaces" to accomplish the design of its self-replicative machine which should be able, from materials found in the environment, to produce any machine described in its program, including a copy of itself [21]. He could, thus, overcome the real physical conditions to work in a highly simplified world yet capable of generating a high complexity. On this basis, he developed a cellular automaton of some 200,000 cells in 29 states containing a universal copier, a description of itself and a Turing machine for supervision.

Cellular automata have emerged from laboratories in 1970 with the famous Life Game of John Horton Conway [22, 23].

2.2. Presentation

A cellular automaton is a tuple \( A = (Q; Z^n; f; V) \), where \( Q \) is a finite set called the set of states, \( Z \) is the set of integers, \( Z^n \) is the cellular space, \( f \) is the local transition function, \( n \) is the automaton dimension, and \( V \) is the neighborhood vector. An automaton cell at time \( t \) is characterized by a pair \((i; z(t))\), where \( i = (i_1; i_2; \ldots ; i_n) \) is an element of \( Z^n \) which can localize a cell in the cell space and \( z(t) \) is an element of \( Q \) that represents the state of this cell.
The neighborhood vector $V$ is a vector that defines the neighborhood of any cell. If $V$ is equal to $\{v_1; v_2; \ldots; v_i; \ldots; v_m\}$, where $1 \leq i \leq m$, then the neighborhood of the cell having the coordinate $i$ is $i + v_1; i + v_2; \ldots; i + v_m$. In two dimensions, the most famous neighbourhoods are Von Neumann neighbourhood given by the neighborhood vector : \{(0;1); (-1;0); (1;0); (0;1); (0;0)\} and Moore neighbourhood given by the neighborhood vector : \{(-1;-1); (-1;1); (1;-1); (0;-1); (-1;0); (0;0); (0;1); (1;0); (1;1)\}.

The local transition function $f$ is an application of $Q_m$ to $Q$, it means, to compute the new cell state, we consider the state of each cell in the neighborhood of the cell in question and we applied to the m-tuple obtained the local transition function. This latter may be the same for all cells, it is called uniform cellular automaton (or homogeneous); it can be different from one cell to another, it is called non-uniform cellular automaton (heterogeneous).

All cells change state asynchronously. Let $Z(t)$ the states set of all cellular automaton cells at time $t$. We say that $Z(t)$ is the cellular automaton configuration at time $t$. We can thus define a function $F$ which associates, the automaton configuration at time $t+1$ to the automaton configuration at time $t$, (i.e., $Z(t+1) = F(Z(t))$). $F$ is the global transition function of the cellular automaton.

3. Proposed approach

As already mentioned, in this work and to solve the encryption problem, we are interested in particular to a dynamic system: cellular automata. These are discrete dynamical systems whose phase space, which means all states which are accessible, are composed of finite automata arranged in a regular grid. Each of these automata is, at a given time, in a certain state and all of these local states provide the overall state (the configuration) of the cellular automaton. All these components are identical and are updated synchronously in successive iterations. Their new state is calculated at each iteration using a local rule applied to the states of a finite set of neighboring automata. The same rule is applied simultaneously to all the network automata. The main model features are:

- local interaction,
- synchronous update (parallel) of local states,
- uniformity.

Despite the simplicity of this definition, a large number of behaviors, sometimes very complex, appear in the study of cellular automata dynamics. Moreover, a detailed classification of cellular automata based on their properties has been an open problem for a long time [24, 25, 26]. This wide variety of behaviors and characteristics of these systems make them particularly relevant for the modeling of some natural phenomena (when many identical elements are in short-range interaction).

However, for our proposed work, the concept of cellular automata is coupled to the concept of optimization, or in other words, the cellular automaton is only a means to optimize the resolution of the studied problem (problem of encryption). Thus, the proposed encryption method, using a law of evolution, can describe the future system states depending on the current state. It is opposed to stochastic systems falling to the theory of probabilities. The law of evolution is often presented in the form of a change in the system state in a short time period which may be infinitesimal (differential equations, partial differential equations) or discrete (recurrence, iteration process). To solve the system is equivalent to find an expression of the system state at any time, given an initial condition.

Thus, we used non-uniform cellular automata. Such models have been used in the literature and are also known under the name of hybrid cellular automata (HCA) [27, 28, 29]. To relax the constraint of uniformity, while preserving the other advantages of the original model (massive parallelism and locality), has indeed several advantages. This allows more flexibility in modeling and increases the cryptosystem confusion level. Indeed, it has been shown in [30] that the generation of pseudo-random sequences for cryptography using elementary cellular automata was of low quality. One way to increase the quality of these sequences is the use of non-uniform cellular automata based on several different rules [31]. However, we noticed that few theoretical studies have been conducted in the case of non-uniform cellular automata.

To summarize, our work will focus on the development of new crypto-systems using optimization discrete dynamical models: non-uniform cellular automata. The following section provides definitions of a dynamic system, a discrete dynamic system and a non-uniform cellular automaton. Next, details of the proposed approach give rise to two new cryptosystems, S2CA and S2CA-EX, will be illustrated.
3.1. Discrete dynamical system and non-uniform cellular automata

**Definition 3.1.1:**
A **dynamical system** is a triple $(X, T, F)$, where $X$ is a set, $T$ a monoid noted additively and $F : X \times T \to X$ a function such that:
1. $\forall x \in X, F(x, 0) = x$.
2. $\forall x \in X, \forall s, t \in T, F(F(x, s), t) = F(x, s + t)$.

For a dynamical system $(X, T, F)$, the notation $F^n(x)$ is usually used for $F(x, n)$. Then, the conditions for a dynamical system can be rewrite [32]:

$$F^0 = \text{id} \text{ et } F^t = F^s \circ F^t.$$

(1)

The set $T$ is a **set time**; this is usually one of the sets: $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}$ or $\mathbb{N}$. The set $X$ is **phase space**; it means all possible system states. In classic cases, $X$ has a certain structure (topological space, differential variety, complex analytic variety, measured space, etc) and the functions $F^n$ follow this structure.

A dynamical system $(X, T, F)$ is discrete if $T$ is discrete (i.e., $T = \mathbb{N}$ or $T = \mathbb{Z}$). These systems are also called **iterative systems**.

Indeed such a system is fully determined by a continuous function $F : X \to X$:

$$\forall n \in \mathbb{N}, F^n = \underbrace{F \circ F \circ \ldots \circ F}_{\text{n fois}}.$$  

(2)

If $T = \mathbb{Z}$, the entire dynamical system is calculated by successive iterations of the function $F^1$ (and eventually of its inverse). In this case, we can simplify the definition of a discrete dynamical system as follows:

**Definition 3.1.2:** A **discrete dynamical system** is a pair $(X, F)$ where $X$ is a metric space and $F : X \to X$ a continuous function for the topology induced by the metric.

These systems are very common. It suffices to know the function which allows moving from one state to the next to define the entire dynamic.

**Definition 3.1.3:** A **non-uniform cellular automata** is a couple $(A, (\theta_i, r_i)_{i \in \mathbb{Z}})$ where $A$ is a finite set called alphabet and where, for any integer $i$, $r_i$ is a positive integer and $\theta_i : A^{2r_i+1} \to A$ is a local rule of radius $r_i$.

This structure induces a global rule $H_\theta : A^2 \to A^2$ defined by:

$$\forall x \in A^2, \forall i \in \mathbb{Z}, H_\theta(x) = \theta_i(x_{[r-i, i+r]}).$$  

(3)

A non-uniform cellular automaton and its global transition function can be identified in the same way as a cellular automaton. An element $x \in A^2$ is called automaton configuration and the sequence $\theta$ is called automaton distribution (of local rules). Given a distribution $\theta$, we note $H_\theta$ the non-uniform cellular automaton induced by $\theta$ and we say that $\theta$ generates $H_\theta$.

3.2. Description of S2CA

3.2.1. Terminology

The following table shows the correspondence between the concepts of cellular automata and those of the cryptographic process adopted by the first proposed algorithm S2CA.

<table>
<thead>
<tr>
<th>Cellular automata</th>
<th>Cryptography</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Image</td>
</tr>
<tr>
<td>cell</td>
<td>Pixel</td>
</tr>
<tr>
<td>neighbors</td>
<td>neighboring pixels</td>
</tr>
<tr>
<td>state</td>
<td>pixel color</td>
</tr>
<tr>
<td>transition rule</td>
<td>calculating the encrypted image</td>
</tr>
</tbody>
</table>

3.2.2. Problem definition

Let $P_1$, $P_2$, $\ldots$, $P_m$ the various pixels in an image $I$ of size $n \times m$.

We note by $(R_{i,j}, G_{i,j}, B_{i,j})$ the coding on RGB space of the pixel $P_{i,j}$.

Consider this encoding mode of pixel values, the objective of S2CA is to dynamically change (iteratively) the maximum of the positions of the various pixels of the image $I$ by performing permutations of the positions of the pixels and their most different neighboring pixels. This becomes a combinatorial optimization problem that can be solved using the dynamical system: cellular automata.

The following algorithm summarizes the steps of the proposed encryption scheme based on cellular automata:
Algorithm S2CA
begin
1) Coding of original image,
    For i = 1 to Nbr of iterations Do
2) Generating the set of pixels to be processed,
    For j = 1 to Nbr of pixels resulting from step 2 Do
    3) Choosing the type of neighborhood,
    4) Study of the neighborhood of each of the pixels,
    5) Permutation of pixels locations (treated pixel, neighboring pixel),
end
end.

3.2.3. S2CA algorithm

Step 1 : Adopted coding
The coding adopted by the first algorithm is simple. It is the representation on RGB space of image pixels. Three components (matrices) R, G and B, resulted and each handled separately but in the same way. Thus, the ciphering operation consists to alter pixels positions of the original image which is concretized by the fact of disrupt in parallel the elements positions of the three components R, G and B in order to have the corresponding ciphering image.

Step 2 : Generating the set of pixels to be processed
In order to accelerate the calculation process and to assure, in the same time, a slowly convergence to the solution, only a subset of the overall set of pixels forming the entire image will be processed on a certain iteration. The pixels forming the subset will be randomly selected from all image pixels. The cardinality of this subset will be determined experimentally and the details are explained in the section reserved for experimentation.

Step 3 : Choosing the type of neighborhood
In order to have a non-uniform behavior of the cellular automaton modeling the problem of encryption, we consider two possible types of neighborhood: Von Neumann neighborhood (4-neighborhood) and the Moore neighborhood (8-neighborhood). This aspect is very important to increase the confusion level of the proposed algorithm and thus to greatly complicate the task of the cryptanalysts (attackers). For each of the pixels chosen in the previous step, one of two types of neighborhood is randomly selected to be studied in the next step.

Step 4 : Study of the neighborhood
For each of the pixels selected in the step 2, the neighborhood type chosen in the previous step is considered in order to select the pixel whose position will be permuted with the treated pixel position. Among this neighborhood, the most different neighbor (pixel) from the treated pixel will be elected for pixels positions swapping. The difference will be measured by the following function: /$
\frac{F(P_j/P_{kl})}{(R_{ij} - R_{kl}) + (G_{ij} - G_{kl}) + (B_{ij} - B_{kl})}$.
where:
$P_j$ is the treated pixel,
$P_{ij}$ is one of the eight neighboring pixels $P_{ij}$,
$R_{ij}$, $G_{ij}$ and $B_{ij}$ are the components encoding the pixel $P_{ij}$ following the RGB coding space,
$R_{kl}$, $G_{kl}$ and $B_{kl}$ are the components encoding the pixel $P_{kl}$ following the RGB coding space.

So, the function quantifying the differences between the processed pixel and its neighbors (four or eight neighbors) will be calculated. The neighbor corresponding to the greatest difference value is selected for swapping positions. More explicitly, this can be formalized as follows:
$F = \text{Max}(F(P_j/P_{kl}))$.

Step 5 : Stop criterion
After a number of iterations experimentally determined, the algorithm converges to the best found solution. It is important to remember that no guarantee is given about the quality of this solution, so it may not be optimal, but it is the best solution found.

3.2.4. Deciphering
We consider, now, the decryption process which is the inverse process allowing to make the image again intelligible without any loss of information. Our first proposed algorithm S2CA is a symmetric algorithm, so the generated key must be kept secret. The latter is calculated at the same time with image ciphering over the iterations. This key is the permutation of the positions of the pixels forming the ciphered image to obtain the positions of the pixels forming the original image. Therefore, it varies from...
I. Souici, H. Seridi, H. Akdag

one image to another since it depends on the image. And it is only by introducing the appropriate key that the pixels of the encrypted image join their initial positions to regenerate the original image. Explicitly, to encrypt a given image, the size of the key generated by this algorithm represents the multiplication of image size in terms of pixels by the number of bits needed to encode the image size.

3.3. Description of S2CA-EX

The second algorithm is an extension or a safer alternative than the first algorithm S2CA. Note that only the scheme of this second algorithm is almost the same as the first algorithm. What make the difference between the two algorithms are the adopted encoding and the resulting operations.

3.3.1. Terminology

The correspondence between the concepts of cellular automata and those of the cryptographic process adopted by the second proposed algorithm, S2CA-EX, is presented through Table 2.

Table 2. Correspondence between cellular automata and cryptography according to S2CA-EX.

<table>
<thead>
<tr>
<th>Cellular automata</th>
<th>Cryptography</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>RGB image coding</td>
</tr>
<tr>
<td>cell</td>
<td>an R or G or B element of</td>
</tr>
<tr>
<td></td>
<td>an RGB image coding</td>
</tr>
<tr>
<td>neighbours</td>
<td>R or G or B neighbouring value</td>
</tr>
<tr>
<td>state</td>
<td>R or G or B value</td>
</tr>
<tr>
<td>transition rule</td>
<td>calculating the ciphering image</td>
</tr>
</tbody>
</table>

3.3.2. Ciphering

The encoding adopted by this second algorithm represents the concatenation of the elements of the three components R, G and B encoding the image pixels. The encryption operation involves maximum disrupting of the positions of the various elements forming the original image coding according to the model summarized through the following algorithm.

**Algorithm S2CA-EX**

begin

1) Coding of original image,
   *For i = 1 to Nbr of iterations Do*
   2) Generating the set of elements to be processed,
      *For j = 1 to Nbr of elements resulting from step 2 Do*
      3) Choosing the type of neighborhood,
      4) Study of the neighborhood of each of the elements,
      5) Permutation of elements locations (treated element, neighboring element),
end

End.

More explicitly, this second algorithm extends the work of S2CA on a more confusional coding space. Indeed, as will be demonstrated through the results presentation, with S2CA-EX, pixels change not only their positions but they also change their values. This offers the S2CA-EX a most important confusional power than S2CA confusional power. Similarly, the solution will be calculated over a set number of iterations fixed experimentally. However, no guarantee is given about the quality of this solution, but cryptographically speaking, it remains better than that calculated by S2CA.

3.3.3. Deciphering

The same S2CA deciphering mechanism will be used by this second proposed algorithm S2CA-EX. The only difference is that we handle here, the elements positions and not pixels positions.

4. S2CA Experimental results and discussion

4.1. S2CA Experimental results

Before presenting the results of S2CA applying on test images, it will be important to illustrate the adopted parametric choices (number of iterations and treated pixels rate among all image pixels). For more clarity, we present the case of application of S2CA on the test image shown in Figure 1. Table 3 and Figure 2 show the results of different tests in terms of efficiency and ciphering time according to different values of number of iterations and rate of processed pixels per iteration. The ciphering images corresponding to the different tests are given in Figure 4.

Fig. 1. Model image (size : 120 × 120).
Table 3. Evolution of S2CA efficiency and ciphering time according to the number of iterations and the processed pixels rate.

<table>
<thead>
<tr>
<th>Tests</th>
<th>Number of iterations</th>
<th>Treated pixels rate</th>
<th>Ciphering time (ms)</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>40 %</td>
<td>72</td>
<td>1102796</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>60 %</td>
<td>76</td>
<td>1198934</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>80 %</td>
<td>83</td>
<td>1281956</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>40 %</td>
<td>79</td>
<td>1298816</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>60 %</td>
<td>88</td>
<td>1383792</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>80 %</td>
<td>97</td>
<td>1475958</td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td>40 %</td>
<td>91</td>
<td>1399884</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>60 %</td>
<td>112</td>
<td>1488932</td>
</tr>
<tr>
<td>9</td>
<td>30</td>
<td>80 %</td>
<td>125</td>
<td>1576218</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>40 %</td>
<td>104</td>
<td>1467624</td>
</tr>
<tr>
<td>11</td>
<td>40</td>
<td>60 %</td>
<td>129</td>
<td>1564044</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
<td>80 %</td>
<td>164</td>
<td>1629872</td>
</tr>
<tr>
<td>13</td>
<td>50</td>
<td>40 %</td>
<td>119</td>
<td>1523646</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>60 %</td>
<td>153</td>
<td>1617478</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>80 %</td>
<td>219</td>
<td>1685548</td>
</tr>
<tr>
<td>16</td>
<td>60</td>
<td>40 %</td>
<td>127</td>
<td>1562804</td>
</tr>
<tr>
<td>17</td>
<td>60</td>
<td>60 %</td>
<td>185</td>
<td>1652636</td>
</tr>
<tr>
<td>18</td>
<td>60</td>
<td>80 %</td>
<td>290</td>
<td>1726268</td>
</tr>
<tr>
<td>19</td>
<td>70</td>
<td>40 %</td>
<td>145</td>
<td>1590866</td>
</tr>
<tr>
<td>20</td>
<td>70</td>
<td>60 %</td>
<td>248</td>
<td>1705580</td>
</tr>
<tr>
<td>21</td>
<td>70</td>
<td>80 %</td>
<td>340</td>
<td>1759844</td>
</tr>
<tr>
<td>22</td>
<td>80</td>
<td>40 %</td>
<td>158</td>
<td>1614856</td>
</tr>
<tr>
<td>23</td>
<td>80</td>
<td>60 %</td>
<td>275</td>
<td>1729782</td>
</tr>
<tr>
<td>24</td>
<td>80</td>
<td>80 %</td>
<td>425</td>
<td>1782832</td>
</tr>
<tr>
<td>25</td>
<td>90</td>
<td>40 %</td>
<td>178</td>
<td>1673454</td>
</tr>
<tr>
<td>26</td>
<td>90</td>
<td>60 %</td>
<td>332</td>
<td>1747002</td>
</tr>
<tr>
<td>27</td>
<td>90</td>
<td>80 %</td>
<td>460</td>
<td>1782064</td>
</tr>
<tr>
<td>28</td>
<td>100</td>
<td>40 %</td>
<td>213</td>
<td>1680494</td>
</tr>
<tr>
<td>29</td>
<td>100</td>
<td>60 %</td>
<td>420</td>
<td>1786838</td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>80 %</td>
<td>574</td>
<td>1820268</td>
</tr>
<tr>
<td>31</td>
<td>110</td>
<td>40 %</td>
<td>263</td>
<td>1794728</td>
</tr>
<tr>
<td>32</td>
<td>110</td>
<td>60 %</td>
<td>471</td>
<td>1765100</td>
</tr>
<tr>
<td>33</td>
<td>110</td>
<td>80 %</td>
<td>650</td>
<td>1830178</td>
</tr>
<tr>
<td>34</td>
<td>120</td>
<td>40 %</td>
<td>317</td>
<td>1716952</td>
</tr>
<tr>
<td>35</td>
<td>120</td>
<td>60 %</td>
<td>528</td>
<td>1815850</td>
</tr>
<tr>
<td>36</td>
<td>120</td>
<td>80 %</td>
<td>730</td>
<td>1857216</td>
</tr>
</tbody>
</table>

Fig. 2. (a) Variation of S2CA ciphering time through the performed tests, (b) Variation of efficiency through the performed tests.
Fig. 3. (a) Variation of ciphering time per iteration according to treated pixels rate, (b) variation of efficiency per iteration according to treated pixels rate.

Fig. 4. Ciphering images obtained by S2CA for different parameter values.
As our goal is to achieve a compromise between efficiency and ciphering time, so, according to these tests value, it is clear that a number of 90 iterations and a pixels treated rate of 80% are sufficient to obtain good efficiency with an acceptable ciphering time (460 ms). These parameter values are valid whatever the image because the efficiency doesn’t depend on the image size.

We have conducted several experiments using images of different sizes and nature (black and white, with grey level, color) to illustrate the performance of S2CA. Figures 5, 6 and 7 present a sample of our experiments. They show the ciphering versions of three original images.

The following table (Table 4) presents the results in terms of key sizes, ciphering and deciphering time of the three models images presented above obtained by S2CA.

<table>
<thead>
<tr>
<th>Image</th>
<th>Image Size (pixels)</th>
<th>Key size (bits)</th>
<th>Ciphering time (ms)</th>
<th>Deciphering time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>300×187</td>
<td>897600</td>
<td>3160</td>
<td>316</td>
</tr>
<tr>
<td>I2</td>
<td>175×200</td>
<td>560000</td>
<td>1574</td>
<td>207</td>
</tr>
<tr>
<td>I3</td>
<td>50×50</td>
<td>30000</td>
<td>318</td>
<td>46</td>
</tr>
</tbody>
</table>

**4.2. S2CA Discussion**

A good crypto-system must satisfy the six key points of Kerckhoffs [33] and those of Shannon [34]. More explicitly, the crypto-system quality is measured primarily by its resistibility face to different types of attacks.

Let’s start by analyzing the resistibility of S2CA against a differential attack. Such attack attempts to reach conclusions on the ciphering algorithm operating by comparing ciphered versions of several original images and in best case ciphered versions of several blocks forming a same image. Thus, in the case of block ciphering algorithms (DES [35, 36, 37], 3DES [35, 38], AES [39], etc.), when the image contains homogeneous areas, all identical blocks are also identical after encryption, so that the ciphering image will contain textured areas. But since S2CA processes on the image in one pass, which means that it operates on the whole image and not on image blocks; then the differential cryptanalysis will be shelved. Therefore also, S2CA presents robustness to noise versus block cipher algorithms, which is equivalent to the fact that an error in a coded bit will not propagate any significant errors in the current block and by following in the whole image during the blocks recombination.

The other frightening cryptanalysis is the brute force attack; the short length key permits it. From the results summarized in Table 4, it is clear that the sizes keys are dependent on images sizes. In the case of the image I3 which is a small image (50×50), we note that the size of the key generated by S2CA is largely secured (3000 bits). And as in reality it is almost rare to process images of very small sizes, S2CA is relatively secure...
face an exhaustive attack. It depends on the size of the processed image.

5. S2CA-EX Experimental results and discussion

5.1. S2CA-EX Experimental results

As in section 4.1, we fix the parameter values by applying several times S2CA-EX on the image model presented on figure 1. Then, we present the results of application of S2CA-EX on the same models images previously used.

Table 5. Evolution of S2CA-EX efficiency and ciphering time according to the number of iterations and the processed pixels rate.

<table>
<thead>
<tr>
<th>Tests</th>
<th>Number of iterations</th>
<th>Treated pixels rate</th>
<th>Ciphering time (ms)</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>20 %</td>
<td>1746</td>
<td>1951096</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>40 %</td>
<td>1938</td>
<td>2497608</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>60 %</td>
<td>2601</td>
<td>2722146</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>20 %</td>
<td>2179</td>
<td>2514624</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>40 %</td>
<td>2749</td>
<td>2792296</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>60 %</td>
<td>3460</td>
<td>2854822</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>20 %</td>
<td>2706</td>
<td>2784176</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>40 %</td>
<td>3487</td>
<td>2855576</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>60 %</td>
<td>3182</td>
<td>2847828</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>20 %</td>
<td>3144</td>
<td>2844636</td>
</tr>
<tr>
<td>11</td>
<td>30</td>
<td>40 %</td>
<td>3457</td>
<td>2854278</td>
</tr>
<tr>
<td>12</td>
<td>30</td>
<td>60 %</td>
<td>3007</td>
<td>2839062</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>20 %</td>
<td>3514</td>
<td>2863960</td>
</tr>
<tr>
<td>14</td>
<td>40</td>
<td>40 %</td>
<td>3372</td>
<td>2849100</td>
</tr>
<tr>
<td>15</td>
<td>40</td>
<td>60 %</td>
<td>3415</td>
<td>2852518</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>20 %</td>
<td>3506</td>
<td>2860318</td>
</tr>
<tr>
<td>17</td>
<td>50</td>
<td>40 %</td>
<td>3520</td>
<td>2854278</td>
</tr>
<tr>
<td>18</td>
<td>50</td>
<td>60 %</td>
<td>3108</td>
<td>2842054</td>
</tr>
</tbody>
</table>

Fig. 8. (a) Variation of S2CA-EX ciphering time through the performed tests, (b) Variation of efficiency through the performed tests.

Fig. 9. (a) Variation of ciphering time per iteration according to treated pixels rate, (b) variation of efficiency per iteration according to treated pixels rate.
We present below the results of application of this second algorithm, S2CA-EX, on the same models images previously used to test the first proposed algorithm S2CA. Thus, Figures 11, 12 and 13 recall these models images and show the corresponding ciphered images.

Table 6. Results obtained by S2CA-EX.

<table>
<thead>
<tr>
<th>Image</th>
<th>Image Size (pixels)</th>
<th>Key size (bits)</th>
<th>Ciphering time (ms)</th>
<th>Deciphering time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>300×187</td>
<td>302940</td>
<td>8710</td>
<td>586</td>
</tr>
<tr>
<td>I2</td>
<td>175×200</td>
<td>178500</td>
<td>5487</td>
<td>324</td>
</tr>
<tr>
<td>I3</td>
<td>50×50</td>
<td>97500</td>
<td>1039</td>
<td>137</td>
</tr>
</tbody>
</table>

The following table shows the results obtained by S2CA-EX applied to the models images I1, I2 and I3.
5.2. S2CA-EX Discussion

Starting by the comparison of the computation time of S2CA and S2CA-EX with that of the main cryptosystems (AES [39] on the side of symmetrical cryptosystems, RSA [1] on the side of asymmetrical cryptosystems and EEA [40] and SEC [41] which are ciphering algorithms exploiting genetic algorithms), Table 7 and Figure 14 show a summary of ciphering and deciphering time of the image I2 by each of the mentioned algorithms.

Table 7. Ciphering and deciphering time of S2CA and S2CA-EX compared to the main ciphering standards.

<table>
<thead>
<tr>
<th>Ciphering Time (ms)</th>
<th>S2CA</th>
<th>S2CA-EX</th>
<th>AES</th>
<th>RSA</th>
<th>EEA</th>
<th>SEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ciphering Time (ms)</td>
<td>1574</td>
<td>5487</td>
<td>500</td>
<td>42800</td>
<td>25300</td>
<td>1800</td>
</tr>
<tr>
<td>Deciphering Time (ms)</td>
<td>207</td>
<td>324</td>
<td>80</td>
<td>42500</td>
<td>260</td>
<td>60</td>
</tr>
</tbody>
</table>

Fig. 14. Ciphering and deciphering time of S2CA and S2CA-EX compared to the main ciphering standards.

However, in cryptography, computing time isn’t the unique evaluation criterion of cryptosystems, but the confusion is one of the key concepts outlined by Shannon on which must be based a ciphering algorithm. It is used to hide the relationship between the clear data (original) and the corresponding encrypted data. Therefore, it aims to make the data readable as little as possible. In our case, the two algorithms developed S2CA and S2CA-EX are of different confusion levels.

Indeed, the confusion level of S2CA-EX is better than that of S2CA because S2CA-EX uses the positions elements encoding the image pixels, each considered as separate unit (cell) of the final image coding (network). Thus, the ciphered image will have a good chance for containing new pixels not included in the original image. This thing complicates considerably, even penalizes, any possible cryptanalysis. However, S2CA confusion level is strictly dependent on the size of the image to be encrypted because it operates on the pixels image positions. The other important point participating on the increase of the confusion power of the proposed approach is the use of non-uniform cellular automata, where the randomly change of the neighborhood type tested for each chosen element for treatment over the different iterations of the converging process, doesn’t allow a cryptanalyst to follow or guess the intermediate states leading to the final results.

Testing now the sensitivity of the proposed cryptographic process to the used key. In our case, the generated key is calculated from the original image and the corresponding ciphered image, so it changes from a problem instantiation to another. Thus, the encryption on different instances of the same image gives rise to a set of different ciphered images, each time, and generates different session key for the ciphering of the same original image. This will provide our ciphering proposed approach a very high sensitivity to key since all intercepted key by an illegal manner will be used to decrypt only a single encrypted version of the same image, so it will not be useful later.

The problem encountered in the case of symmetrical cryptosystems, and therefore in the case of our developed algorithms which are symmetrical algorithms, is the communication of the secret key to the recipient in order to be used for the extraction of the original information (deciphering). To do this, some designers have proposed to combine the functionality of the symmetrical and asymmetrical cryptography. This is the case, for example, of PGP. It creates randomly an IDEA secret key and encrypts the data with this key; then it encrypts the IDEA secret key and sends it using the recipient RSA public key. Thus, the same principle can be adopted to secure the transfer of our key by encrypting them using an asymmetrical cryptosystem.
In this case a hybrid scheme of a secure transmission can be seen as shown in the following figure:

However, for S2CA and S2CA-EX, it is clear that the key size is strictly dependent on the image size. The problem which arises when encrypting an image of large size is that the size of generated keys becomes also large which consumes a great key ciphering time because public cryptosystems are already slow. To remedy this defect, we propose to encode (or compress) the generated key by a lossless coding system such as Huffman coding [42], for example, to reduce its size, then it will be ciphered by an asymmetric cryptosystem. The package will be, thereafter, sent to the recipient including the ciphered image and the coded and ciphered key. Upon receipt, we proceed vice versa. We decrypt the coded key using the recipient's public key, then we decode it to get the key generated by S2CA or S2CA-EX, which will allow the reproduction of the original image. Thus, the scheme of security processes encompassing this last feature is as follows:

Co-published by Atlantis Press and Taylor & Francis
Copyright: the authors
A second solution for a secure transmission of the generated session key may be envisaged. It is the watermarking technique. After having coded the key generated by S2CA or S2CA-EX in the same way proposed in the first key secure transmission solution (using Huffman coding), we propose here to mark the ciphered image by the coded session key. Thus, the security process scheme encompassing the second key transmission solution can be summarized as in figure 17. However, this second transmission solution is conditioned by the coded key size to satisfy the compromise Robustness-capacity-Invisibility during the watermarking [43], [44].

Fig. 17. General scheme of the ciphering process and generated key secure transmission by watermarking.

6. Conclusion

In this article, we are interested by images ciphering problem. We propose two new encryption algorithms named S2CA and S2CA-EX based on non-uniform cellular automata optimization algorithm where the solutions are calculated dynamically iteration after iteration. We have benefited from the randomness - which is the cryptanalysts enemy - to choose, both pixels processed on each iteration and the type of neighborhood considered to address each of these pixels. It is almost impossible to reach the initial information (images) or less to establish a relationship between the encrypted ones. Indeed, we always get different encrypted images for each application of the algorithm. Similarly for the generated encryption key that changes with each problem instantiation even for the encryption of the same image. Comparing S2CA and S2CA-EX with those registering under the same encryption symmetric mode operating by block, it will be useful to note that differential cryptanalysis is easily applied in parallel on different blocks. However, and since our algorithms operate on the whole image, this will make them so difficult breakable, even brought out of this category of attacks. Thus, according to the results of the comparison presented above, it is clear that S2CA and S2CA-EX are the most resistible to exhaustive attacks. Similarly, our ciphering approach has shown a very good confusion level because of the proposed cellular automata heterogeneity characteristic and the randomness aspect exploited on some S2CA or S2CA-EX steps.

References


33. A. Kerckhoffs. La cryptographie militaire, in Journal des sciences militaires (Janvier 1883).


40. I. Souici, H. Seridi, and H. Akdag, Images Encrypton by the Use of Evolutionary algorithms, in Analog

