Abstract – A multi-secret sharing scheme, which is based on one-dimensional reversible cellular automata with memory, allowing any member of a set of participants to share a secret color image with the rest of participants is presented. Moreover, it is also explained that with this scheme, any participant is able to recover all the secret color images if and only if the whole set of participants pools their shadows. Finally, the security of the multi-secret sharing scheme is studied and it is proved that the protocol withstands the most important attacks. In particular, it is secure against any statistical attack.

Keywords: Graphic cryptography, Multi-secret sharing schemes, One-dimensional cellular automata.

1. Introduction and Preliminaries

It is well known that secret sharing schemes are cryptographic protocols that permit to share a secret among a set of participants. In the protocol, specific subsets of participants can recover the secret. These schemes were proposed by Shamir ([18]) and Blakley ([4]). In the origin, their purpose was to safeguard secret data from loss. Nevertheless, secret sharing schemes are used today in many applications, specially in access control, opening safety devices, etc.

An extension of the previous protocol occurs when the set of participants wants to share several secrets and not only one. In this case, the scheme is called a multi-secret sharing scheme.

The most extended protocols of (multi)secret sharing schemes are the \((k,n)\)-threshold schemes, where \(k,n \in \mathbb{Z}\), and \(1 \leq k \leq n\). In these protocols, a Dealer computes \(n\) secret shadows from the original secrets and distributes them, in a secure way, to \(n\) participants. Then, any \(k\), or more, participants can recover the original secrets by pooling their shadows; whereas any group of \(k-1\), or fewer, participants cannot recover the original secrets.

A \((k,n)\)-threshold scheme is called ideal if the size of each shadow is the same as the size of the original secret, and it is said perfect if the knowledge of any \(k-1\), or fewer, shadows provides no information about the original secret (see [16], [19]).

When the secrets are images, photos or figures, these schemes are called graphic secret sharing schemes.

The first proposal for a graphic secret sharing scheme is known as visual cryptography ([17]). This scheme is a \((k,n)\)-threshold scheme and it is perfect but not ideal. Since then, many modifications and improvements have been proposed in order to solve some problems related to the contrast of the recovered secret images, the number of colors of the original image, etc. (see [2], [6], [7], [14], [15], [20], [23]).

In the last years several (multi)secret sharing schemes have been proposed but mostly aimed at sharing texts. Only a few of them targeted sharing images. Some of the former ones are based on hash functions ([9], [11], [12]), Lagrange interpolation ([10], [26]), coding theory ([8]), and on the RSA cryptosystem ([13]). Among the latter, the schemes presented in [5], [22] were based on visual cryptography, and the proposed in [25] is defined for multi-users with the purpose of being used in watermarking schemes.

In this work, we present a multi-secret sharing scheme based on one-dimensional reversible cellular automata that allows the sharing of secret color images. In fact, the proposed scheme is an \((n,n)\)-threshold scheme. The shadows computed are again color images.
and they are obtained as configurations of the cellular automata considered.

More specifically, our proposal assumes that a Dealer and \( n \geq 1 \) participants, \( P_1, \ldots, P_n \), exist such that each participant has a secret image defined by any number of colors, \( I_1, \ldots, I_n \). Each participant wants to share his secret image on condition that he could access to the rest of images if all participants agree to do so.

In [3], a multisecret sharing scheme based also on cellular automata was proposed. That scheme is ideal, perfect and secure. Nevertheless, it has some restrictions which have been improved in the present proposal, as it will be explained below.

Our proposal can be used in other applications. For example, if several secrets must be protected by using the same information as if only one secret must be protected, and a secret key can be used to encrypt such secrets. Another application case occurs if the size of the secret to be protected is so big that it becomes convenient or necessary to split it into several parts.

The rest of this paper is organized as follows: In Section II the basic concepts related to one-dimensional reversible cellular automata are recalled. In Section III the proposed multi-secret sharing scheme is presented. Some experimental examples are shown in Section IV and the security of the scheme is analyzed in Section V. Finally, the conclusions are shown in Section VI.

2. One-dimensional reversible cellular automata

Finite cellular automata (CA) are discrete dynamical systems formed by a finite number of identical objects called cells ([24]). These cells have a state which changes at every discrete time step according to a deterministic rule.

A one-dimensional finite cellular automata (CA) is a set \( \mathcal{A} = (C, S, V, f) \), where \( C \) is the cellular space formed by a linear array of \( m \) cells. Each cell is denoted by \( \langle j \rangle, 0 \leq j \leq m - 1 \). \( S \) is the state set, i.e., the set of all possible values of the cells. In general, if the CA considered is finite with \( p \) states, then \( S = \mathbb{Z}_p \). The ordered finite set \( V = \{ \alpha_1, \alpha_2, \ldots, \alpha_d \} \subset \mathbb{Z} \), with \( |V| = d \), is the set of indices of \( C \) so that the neighborhood of a cell \( \langle j \rangle \) is the ordered set of \( d \) cells around the given cell: \( \tilde{V} = \{ a_{j+0\alpha_1}, \ldots, a_{j+0\alpha_d} \} \), where \( a_j \) stands for the state of the cell \( \langle j \rangle \). In general, we denote by \( a_j^{(t)} \in S \) the state of the cell \( \langle j \rangle \) at time \( t \) and by \( \tilde{V}^{(t)} = \{ a_{j+0\alpha_1}^{(t)}, \ldots, a_{j+0\alpha_d}^{(t)} \} \), its neighborhood at the same time.

Finally, the local transition function \( f : S^d \rightarrow S \) is the function determining the evolution of the CA throughout the time, i.e., \( f \) permits to determine the changes of the states of every cell taking the states of its neighbors into account. In this way, the state of the cell \( a_j \) in the time \( t + 1 \) is given by the formula:

\[
a_j^{(t+1)} = f \left( \tilde{V}^{(t)} \right) = f \left( a_{j+0\alpha_1}^{(t)}, \ldots, a_{j+0\alpha_d}^{(t)} \right).
\]

In general, the evolution of a CA considers that the state of every cell at time \( t + 1 \) depends on the state of its neighborhood at time \( t \), \( \tilde{V}^{(t)} \). Nevertheless, it is possible to consider that this evolution also depends on the states of other cells at times \( t - 1, t - 2, \ldots, t - l \), for example. In this case, it is said that the CA is a CA with memory of order \( l + 1 \) (the CA needs to remember \( l + 1 \) configurations to determine the next one), and its transition function (see (1)) can be represented as:

\[
a_j^{(t+1)} = \sum_{h=0}^{l} f^{(t-h)} \left( \tilde{V}^{(t-h)} \right),
\]

where each \( f^{(t-h)} \) is a specific local transition function.

As the cellular space is finite with \( m \) cells, boundary conditions must be established in order to ensure that the evolution of the cellular automata is well-defined. Here, periodic boundary conditions are taken into account. That is, if \( i \equiv j \pmod{m} \) then \( a_i^{(t)} = a_j^{(t)} \).

The set of states of all cells in a time \( t \) is called the configuration at time \( t \) and it is represented by the following vector:

\[
C^{(t)} = (a_0^{(t)}, a_1^{(t)}, \ldots, a_{m-1}^{(t)}) \in S \times \cdots \times S.
\]

In particular, \( C^{(0)} \) is called the initial configuration and the evolution of \( C \) is the sequence

\[
C_\text{evolve} = (C^{(0)}, C^{(1)}, C^{(2)}, \ldots).
\]

If \( \mathcal{C} \) is the set of all possible configurations of the CA, the global function of the CA is defined as a linear transformation,

\[
\Phi: \mathcal{C} \rightarrow \mathcal{C}, \quad \Phi(C^{(t)}) = C^{(t+1)}
\]

that gives the configuration at the next time step during the evolution of the CA. If \( \Phi \) is bijective then the CA is called reversible and there exists another cellular automaton, called its inverse, with global function \( \Phi^{-1} \), such that the backward evolution is possible ([11], [21]).

3. A graphic multi-secret sharing scheme

In this section, a graphic multi-secret sharing scheme is proposed. Our proposal is a \((n,n)\)-threshold scheme such that each participant \( P_1, \ldots, P_n \) shares a secret color image, \( I_1, \ldots, I_n \) with the rest of participants. In the scheme, the whole set of participants can
recover all secret images only when all of them agree to share their shadows.

The scheme proposed in [3] is based on two-dimensional cellular automata whereas the one proposed in this work uses only one-dimensional ones. This fact permits to define more efficient algorithms to compute the shadows than in the previous case. With our proposal, each participant can use their own secret key instead of using the value generated by the Dealer. In previous proposals, the bitlength of that value was exactly 9 bits. This value was fixed because it must be equal to the number of elements of the neighborhood of each cell, which was Moore neighborhood. In our proposal, the bitlength of the secret key chosen for each participant can be different and, hence, the security of the system is improved.

The scheme has been divided into three phases: Definition, Sharing and Recovering.

3.1. Definition phase

In this phase, the Dealer defines the reversible one-dimensional CA with memory of order \( n + 1 \) to be used in the scheme, \( \mathcal{A} = (C, S, V, f) \). In order to do this, he receives the \( n \) secret images, \( I_i \), \( 1 \leq i \leq n \), and the secret keys of each participant for the scheme, \( k_i \). If the keys are similar to those used in symmetric cryptography, and if we denote their bitlength as \( l_i \), then \( l_i \) will be an element of the set \( \{32, 64, 128, 192, 256\} \). If a participant does not send their secret key to the Dealer, the Dealer will generate a secret value for that participant and he will send it to the participant at the end of the sharing phase.

The next step to be done by the Dealer is the normalization of the secret images. In fact, it is very likely that the secret images have different sizes and color palettes. This normalization converts all the secret images into images of the same size (for example, by padding them with white pixels), and saves all the images in files with the largest color palette used.

In this way if the image \( I_i \), \( 1 \leq i \leq n \), is a black and white, grey or color image, defined by \( m_i = r_i \times c_i \) pixels, it can be represented by a linear array of \( m_i p_i \) bits, where \( p_i \) is the number of bits needed to define each pixel, that is, \( p_i \in \{1, 8, 24\} \). Hence, all the images can be considered as images defined by \( m \) bits, where \( m = \max(m_i) \cdot \max(p_i) \). Hence, the cellular space of the CA, \( C \), will be a linear array of bits of size \( m \) and the state set is given by \( S = \mathbb{Z}_2 \).

There are two more elements needed to complete the CA: the set of indices, \( V \), and the transition function, \( f \). In general, \( V \) and \( f \) are unique for each CA; nevertheless, since in our proposal, \( n \) participants will take part in the process and each of them with their own key, \( k_i \), \( n \) sets of indices and \( n \) transition functions will be considered, one for each participant. Both, \( V \) and \( f \), will depend on the value of \( k_i \).

If the secret key of the participant \( P_i \) is \( k_i \), with bitlength \( l_i \) (which must be an even number), the set of indices considered, for a cell \( \langle j \rangle \) given, is

\[
V_i = \{ (j - l_i/2), \ldots, (j - 1), (j), (j + 1), \ldots, (j + l_i/2) \},
\]

so that the neighborhood of a cell \( a_j \) at time \( t \) is

\[
\bar{V}_i(t) = \{ a_{j-l_i/2}, \ldots, a_{j-l_i/2}^l, a_{j-l_i/2}^l+1, \ldots, a_{j+l_i/2}^l \}.
\]

Then, \( V(t) = \{ V_1(t), \ldots, V_n(t) \} \), where \( |V(t)| = l_i + 1 \).

In order to define the transition function \( f \), one can use the expression given in (2) modifying it a suitable way. To do this, we define a transition function for each participant \( P_i \), \( 1 \leq i \leq n \), as follows:

\[
f_i(t)(\bar{V}_i(t)) = f_{i_k}(\bar{V}_i(t)) = f_{i_k}(\bar{V}_i(t)) = \sum_{h=1}^{n} f_{h_k}(V_h(t-h)) \pmod{2},
\]

where \( f_{i_k} \) is the identity: \( f_{i_k}(V_0(t)) = f_{i_k}(a_j^{t_l}) = a_j^{t_l} \).

This way of defining the transition function of \( \mathcal{A} \), \( f \), permits to define its inverse CA by considering the transition function \( g \) of \( \mathcal{A}^{-1} = (C, S, V, g) \) given by

\[
a_j^{t_l+1} = g(V) = f_{i_k}(V_0(t-h)) - \sum_{h=1}^{n} f_{i_k}(V_h(t-h-n)) \pmod{2}.
\]

3.2. Sharing phase

Once the CA is defined, the Dealer must compute the shadows to be shared in a secure way to the \( n \) participants. To do this, he generates a random bit sequence of length \( m \), \( I_0 \), and considers that this is the initial configuration of the CA defined above, i.e.,
\(C(0) = I_0\). The other secret images are the next \(n\) configurations of the reversible one-dimensional CA with memory of order \(n + 1\). So, \(C(i) = I_i, 0 \leq i \leq n\).

Then, the Dealer generates an integer number, \(e\), in order to determine the number of iterations to be applied to the CA, that is, the order of its evolution. This number must be greater than \(n + 1\) in order to avoid that any of the secret images might be considered as a shadow; hence, \(e > n + 1\). So, the Dealer computes the evolution of the CA of order \(e - 1\) from the last configuration known, applying the transition function defined in (3), and he obtains:

\[
\mathcal{e}_{n+e} = \left(C(0), \ldots, C(n), C(n+1), \ldots, C(e), \ldots, C(n+e-1)\right).
\]

Since \(C(0)\) is a random bit sequence, so are the configurations \(C(j), n + 1 \leq j \leq n + e - 1\).

After computing the evolution \(\mathcal{e}_{n+e}\), the Dealer sends to each participant one of the last \(n\) configurations computed, and the number of order of that participant through a secure channel. If the participant did not send to the Dealer any secret key, the Dealer, will send that secret to him.

In this way, each participant \(P_i, 1 \leq i \leq n\), receives the configuration \(C(i+e-1)\), which is the shadow \(S(n-i)\). Finally, the Dealer publishes the configuration \(C(e-1) = S(n)\) so that the following phase can be carried out.

### 3.3. Recovering phase

If the whole set of \(n\) participants agrees to recover all the secret images, they can proceed as follows. Recall that if only one of the participants does not agree, no secret image will be recovered since the proposed scheme is an \((n,n)\)-threshold scheme.

All participants know the public configuration \(S(n) = C(e-1)^{-1}\); so to proceed, they share their shadows and their secret keys: \(S(n-i)\) and \(k_i\), \(1 \leq i \leq n\), where \(S(n-i) = C(i+e-1)\). Next, they use a CA which is the inverse of the one used by the Dealer in the Definition phase, \(C^{-1} = (C,S,V,g)\), and iterate it \((e-2)\) times, computing the evolution of order \(n + e - 2\) by using the transition function given in (4). Note that in this phase, the participants do not need to compute the original initial configuration \(C(0)\) because it is not a secret image but it is the random binary sequence generated by the Dealer in the Definition phase. Hence, they obtain

\[
\hat{e}_{n+e-1} = \left(S(0), \ldots, S(n), S(n+1), \ldots, S(e), \ldots, S(n+e-2)\right)
\]

thus recovering the \(n\) secret images as \(C(j) = S(n+e-j-1), 1 \leq j \leq n\).

### 4. Experimental results

In this section we will present the results obtained with an implementation of the scheme described in the previous section. To do this, we will use three different images defined by a different number of pixels and different color palettes.

The images of the \(n = 3\) participants are the following. The first participant has the logo of WorldComp 2009, \(l_1\), which is an image defined by only two colors and with \(262 \times 96\) pixels (see Fig. 1) and he has not any secret key. The second participant, with a key, \(k_2\), of \(l_2 = 64\) bits, chooses a map of Madrid in the year 1562, \(I_2\). This engraving is a grey level image (actually 253 different grey tones) and it has \(700 \times 400\) pixels (see Fig. 2). Finally, the third participant has a photo of Las Vegas (see Fig. 3) with \(71391\) colors and \(413 \times 400\) pixels, \(I_3\); his secret key, \(k_3\), has also \(l_3 = 64\) bits.

---

**Figure 1:** Logo of WorldComp 2009.

**Figure 2:** Map of Madrid in 1562.

**Figure 3:** Photo of Las Vegas.

Once the Dealer has received the secret images, he normalizes them so that the secret images are defined by \(700 \times 400\) pixels and each pixel is defined by 24 bits. Then, he generates a secret key for the first participant: \(k_1\). From the three keys, the Dealer defines the reversible one-dimensional CA following the steps given in §3.1. The value of the keys used in this example, in hexadecimal base, are: \(k_1 = \ldots\)
6c344d36c3bf1672, k_2 = cdcbeb3edaaa20c9, and k_3 = 8cf6a3730301b941. Hence, the transition function of the CA is
\[ f(\bar{V}) = f_{k_1}(V_1^{(t)}) + f_{k_2}(V_2^{(t)}) + f_{k_3}(V_3^{(t)}) + f_{k_0}(V_0^{(t)}) \]
where \( f_{k_1} \), for example, is given by
\[ f_{k_2}(\bar{V}_2^{(t)}) = 1 \cdot a_{j-32} + 1 \cdot a_{j-31} + 0 \cdot a_{j-30} + 0 \cdot a_{j-29} + \ldots + 1 \cdot a_{j-1} + a_j + 1 \cdot a_{j+1} + \ldots + 1 \cdot a_{j+29} + 0 \cdot a_{j+30} + 0 \cdot a_{j+31} + 1 \cdot a_{j+32} \]

In the sharing phase, the Dealer generates a random bit sequence, \( I_0 = C^{(0)} \), of \( m = 700 \cdot 400 \cdot 24 = 6720000 \) bits; chooses \( e = 6 > n + 1 = 4 \); and determines the evolution or order \( n + e - 1 = 9 \) of the CA. So, he only has to compute 5 configurations, \( C^{(4)}, C^{(5)}, C^{(6)}, C^{(7)}, \) and \( C^{(8)} \), since he knows the \( n = 3 \) first images and the random initial configuration: \( I_1 = C^{(j)}, 0 \leq j \leq 3 \).

Then, the shadow \( C^{(5)} = S^{(3)} \) is published, whereas the shadows to be distributed to the participant \( P_1 \) is \( C^{(6)} = S^{(2)} \), to the participant \( P_2 \) is \( C^{(7)} = S^{(1)} \), and to the participant \( P_3 \) is \( C^{(8)} = S^{(0)} \) (v.g.r. see Fig. 4).

To recover the original images, if the three participants agree, they only have to share their shadows and keys, and to follow the recovering phase.

5. Security

In this section we study the behavior of the proposed scheme with respect to its security.

If all the secret images are of the same size and defined with the same palette, it is clear that the proposed scheme is ideal since all the shadows are of the same size and defined by the same palette. Otherwise, the scheme is not ideal. Moreover, the scheme is perfect since if, at least, one shadow belonging to a participant is unknown, there is no information about the whole configuration of the CA and then it is not possible to determine the evolution of the CA.

Next, we analyze the statistical security of the scheme. This analysis is performed by three tests. The first test is based on the number of colors of each shadow and on a comparison between the histograms of the secret images and the shadows; the second one has been designed in order to test the influence of one modified pixel on the shadow; and the third test consists of computing the correlations of adjacent pixels (horizontal, vertical and diagonal) of the original images and the shadows. Table 1 shows the number of colors of each image.

<table>
<thead>
<tr>
<th>Image</th>
<th>( S^{(5)} )</th>
<th>( S^{(4)} )</th>
<th>( S^{(3)} )</th>
<th>( S^{(2)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colors</td>
<td>277602</td>
<td>277600</td>
<td>277682</td>
<td>277714</td>
</tr>
</tbody>
</table>

The histograms of the secret images and of the shadows \( S^{(3)} \) and \( S^{(2)} \) are given in Figs. 5–6. They show that the histograms of the shadows are fairly uniform and significantly different from the histograms of the original images. The histograms represent the number of pixels (y-axis) for each pixel (x-axis) given by its RGB value (0–255).

Figure 4: Public shadow \( S^{(3)} \).

Figure 5: Histograms of Logo, Madrid and Las Vegas.

Figure 6: Histograms of shadows \( S^{(3)} \) and \( S^{(2)} \).

The next test has been designed in order to test the influence on a shadow by the change of one pixel. The test consists of two values: the Number of
Pixels Change Rate, NPCR and the Unified Average Changing Intensity, UACI. The former measures the percentage of different pixel numbers between two images, whereas the latter gives a measure about the average intensity of differences between two images.

The values of NPCR and UACI are computed as follows: If \( S = (s_{ij}) \) and \( \tilde{S} = (\tilde{s}_{ij}) \) are two shadows obtained from the original secrets that differ in only one pixel, the bipolar array of size \( r \times c \), \( D = (d_{ij}) \), is defined by considering that \( d_{ij} = 0 \) if \( s_{ij} = \tilde{s}_{ij} \), and \( d_{ij} = 1 \) otherwise. The NPCR and UACI are:

\[
NPCR = \frac{\sum_{i=1}^{r} \sum_{j=1}^{c} d_{ij}}{r \cdot c} \times 100\% ,
\]

\[
UACI = \frac{1}{r \cdot c} \left( \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{| s_{ij} - \tilde{s}_{ij} |}{\text{number of colors}} \right) \times 100\% ,
\]

It is clear that a high value of NPCR (near 100%) and a high value of UACI (bigger than 25%) represents that the change of a pixel in the secret image influences significantly in the shadows. As in these experimental results, the values are bigger than 99% and 33%, respectively, we can state that the proposed scheme is very sensitive with respect to small changes in the secret images.

Given that the initial configuration \( C^{(0)} = I_0 \) is generated at random, these tests are not particularly relevant for the current proposal. However, for the sake of completeness we have executed these tests. The results for the NPCR test and for the UACI test comparing the corresponding pairs of shadows are shown in Table 2 and Table 3, respectively.

### Table 2: NPCR test comparing pairs of shadows

<table>
<thead>
<tr>
<th>Shaddows</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S^{(1)}-S^{(2)} )</td>
<td>99.6071</td>
<td>99.6246</td>
<td>99.6186</td>
</tr>
<tr>
<td>( S^{(2)}-S^{(3)} )</td>
<td>99.6100</td>
<td>99.5936</td>
<td>99.6029</td>
</tr>
<tr>
<td>( S^{(3)}-S^{(4)} )</td>
<td>99.6046</td>
<td>99.6089</td>
<td>99.6029</td>
</tr>
<tr>
<td>( S^{(5)}-S^{(6)} )</td>
<td>99.6245</td>
<td>99.6036</td>
<td>99.6132</td>
</tr>
</tbody>
</table>

### Table 3: UACI test comparing pairs of shadows

<table>
<thead>
<tr>
<th>Shaddows</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S^{(1)}-S^{(2)} )</td>
<td>33.5395</td>
<td>33.3947</td>
<td>33.4954</td>
</tr>
<tr>
<td>( S^{(2)}-S^{(3)} )</td>
<td>33.5212</td>
<td>33.3933</td>
<td>33.5243</td>
</tr>
<tr>
<td>( S^{(3)}-S^{(4)} )</td>
<td>33.4910</td>
<td>33.4382</td>
<td>33.3890</td>
</tr>
<tr>
<td>( S^{(5)}-S^{(6)} )</td>
<td>33.4702</td>
<td>33.4764</td>
<td>33.4759</td>
</tr>
</tbody>
</table>

To test the correlation between a set of adjacent pixels, 1000 pairs of two horizontally adjacent pixels, 1000 pairs of two horizontally adjacent pixels, and 1000 pairs of two diagonally adjacent pixels, for the original image as well as for its shadows, have been randomly selected. For each case, the correlation coefficient of each pair has been computed and the results are shown in Tables 4 and 5.

The results state that the correlation coefficients are very different. For example, in the secret images the correlation coefficients for two horizontally adjacent pixels are 0.4114, 0.7364, and 0.9405 (note that these coefficients are bigger if the image has more colors, as expected). Nevertheless, in the four shadows, these coefficients are \(-0.0383, 0.0291, -0.0046, \) and 0.0151, respectively. Similar results are obtained for vertically and diagonally adjacent pixels.

### Table 4: Correlation coefficients of adjacent pixels for the secret color images

<table>
<thead>
<tr>
<th>Image</th>
<th>Logo</th>
<th>Madrid</th>
<th>Las Vegas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>0.4114</td>
<td>0.7364</td>
<td>0.9405</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.4774</td>
<td>0.6051</td>
<td>0.9612</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.3709</td>
<td>0.5344</td>
<td>0.9033</td>
</tr>
</tbody>
</table>

### Table 5: Correlation coefficients of adjacent pixels for the shadows

<table>
<thead>
<tr>
<th>Shadows</th>
<th>( S^{(1)} )</th>
<th>( S^{(2)} )</th>
<th>( S^{(3)} )</th>
<th>( S^{(4)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>(-0.0383)</td>
<td>0.0291</td>
<td>(-0.0046)</td>
<td>0.0151</td>
</tr>
<tr>
<td>Vertical</td>
<td>(-0.0620)</td>
<td>0.0600</td>
<td>0.0341</td>
<td>0.0355</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.0327</td>
<td>0.0051</td>
<td>(-0.0629)</td>
<td>(-0.0217)</td>
</tr>
</tbody>
</table>

As an example, Fig. 7 and Fig. 8 show the correlation distribution of 1000 pairs of horizontally adjacent pixels of secret images and shadows, respectively. One can observe that in the first case (Fig. 7), the distributions follow, approximately, the main diagonal, which gives an idea of the strong correlation among the pixels of the secret images; whereas for the shadows (Fig. 8), the clouds of points seem to distribute in a random way, which indicates a weak correlation among the pixels of the shadows.

6. Conclusions

The most important conclusions of this work are:
1) A graphic multi-secret scheme based on reversible one-dimensional cellular automata has been proposed.

2) This scheme is an \((n,n)\)-threshold scheme and it is ideal-like and perfect.

3) The statistical properties of this scheme have been studied and it has been proved that the multi-secret sharing scheme is secure against statistical attacks.

4) Moreover, in the proposed scheme, each participant can use their own secret key with chosen bitlength, instead of using the value generated by the Dealer, with a fixed bitlength 9. So, the security is improved.

Some future works are:

- Study how to extend this scheme to general \((k,n)\)-threshold schemes with \(k < n\).

- Improve the implementation of the algorithms used.

Acknowledgements. This work has been partially supported by Ministerio de Industria, Turismo y Comercio (Spain), in collaboration with CDTI and Telefónica I+D under the project SEGUR@ (CENIT-2007 2004).

References


