Robust Image Encryption Based on Balanced Cellular Automaton and Pixel Separation

Dijana TRALIC, Sonja GRGIC

Dept. of Wireless Communications, Faculty of Electrical Engineering and Computing,
University of Zagreb, Unska 3, HR1000 Zagreb, Croatia

dijana.tralic@fer.hr

Manuscript received March 10, 2016

Abstract. *The purpose of image encryption is to protect content from unauthorized access. Image encryption is usually done by pixel scrambling and confusion, so process is possible to reverse only by knowing secret information. In this paper we introduce a new method for digital image encryption, based on a 2D cellular automaton and pixel separation. Novelty in the proposed method lies in the application of the balanced 2D cellular automata with extended Moore neighborhood separately on each level of pseudorandom key-image. This process extends key space several times when compared to the previous methods. Furthermore, pixel separation is introduced to define operation for each pixel of the source image. Thanks to pixel separation, decryption process is more difficult to conduct without knowing secret information. Moreover, encryption is robust against different statistical attacks and analysis, does not affect image quality and can cope with loss of encrypted image content.*

Keywords

Image security, encryption, decryption, cellular automaton, statistical tests

1. Introduction

Nowadays images are widely used in various applications, such as communication, multimedia systems, medical imaging, telemedicine, monitoring, and military communication, so their security is becoming increasingly important. Generally, encryption is used to effectively protect images transmitted through public channels, because it makes images unrecognizable, unless decrypted by a secret key. There are several conventional encryption methods, but these algorithms have limitation in encrypting images such as low efficiency, bulky data, and high correlation among pixels [1].

Image encryption algorithm is considered reliable if: the scheme is secure against different statistical and plaintext attacks; there is no visual degradation of decrypted image; corruption of encrypted image part does not propagate on the whole decrypted image; key space is large enough to make it impossible to guess correct secret information; encrypted images show random properties.

Good candidates for image encryption are cellular automata (CA), thanks to their properties like parallelism, homogeneity and unpredictability [2–4]. Cellular automata are distributed systems with a large number of rules that can simulate complicated and pseudorandom behaviors [5]. In recent years, CA has been used for image cryptography in following areas: watermarking [6], [7], secret image sharing [8–11], and image encryption [12–18]. The first image encryption system based on cellular automata was designed with one-dimensional elementary CA [16]. This approach uses only two neighbors to define CA rules so its key space is quite small. Later, 2D cellular automata are introduced for image encryption. Most of approaches use Von Neumann neighborhood [12], [13], [19], [20], or incomplete Moore neighborhood [14], which also influences the capacity of key space. Later, image encryption method based on Moore and extended Moore neighborhoods is introduced [21], but it reduces possible key space volume due to Moore neighborhood size. Chaos theory is applied in [22], however it is quite complex to be used for image encryption and it results in a very small key space volume.

In this paper we introduce a new method for digital image encryption, based on a 2D cellular automaton and pixel separation. First, sender and receiver exchange secret information: seed and separation values. Then 2D cellular automaton with extended Moore neighborhood is used to generate pseudorandom key-image based on selected seed. Encryption is then performed by combining source image with key-image. In this process pixel separation is applied to define operation for each pixel based on secret separation values. Operation for each pixel is selected among ten possible operations making it more difficult to decrypt the encrypted image without secret information.

The rest of the paper is organized as follows. In Sec. 2 the proposed method is presented. Section 3 brings simulation results, and Section 4 contains comparison to state of the art methods. Conclusion is given in Sec. 5.

DOI: 10.13164/re.2016.0548

2. Proposed Method

The proposed method is based on the idea of combining a source image (the image that should be encrypted) with a key-image. Pseudorandom key-image is generated using balanced cellular automaton rules, which are applied on binary levels. Also, rules are formed on extended Moore neighborhood which extends key space. In encryption process, pixel separation is introduced to make decryption more robust and complicated for unauthorized party.

2.1 Cellular Automata

Cellular automaton (CA) [24] is a discrete system that contains a regular grid of cells, each in one finite state. A CA can be presented by a quadruple [24]:

$$\langle C, S, N, f \rangle \tag{1}$$

where:

- C is a d-dimensional cellular space that consists of c cells, i.e. 2-dimensional binary image that consists of $M \times N$ values,
- S is a s-value state space, i.e. s = 2 for binary images (0 or 1),
- N is an n-cell CA neighborhood, i.e. an arbitrarily selected group of n pixels $N(p_o)$ spatially close to the observed pixel p_o, Extended Moore neighborhood includes 24 pixels around the central pixel.
- $f: S^n \rightarrow S$ is a cell-state transition function defined as a set of rules that represents the connection of the selected neighborhood $N(p_o)$ and the value of the observed pixel p_o.

Therefore, the value of pixel p_o in the next time step depends only on values of pixels in the selected neighborhood $N(p_o)$. By considering extended Moore neighborhood (n = 25) and binary images (s = 2), there are total of $s^n = 2^{25}$ = 33554432 possible combinations of values in neighborhood. Also, there are $2^{33554432}$ possible rules. For example, rule 0 means that all combinations of neighboring elements result in the value of the observed pixel p_o equal to 0 (2). Note that a new state of the observed pixel p_o for all combinations of neighborhood presents CA rule's value in the binary form.

$$\begin{aligned}
&\text{if } N(p_o(t))=\{000000000\} \text{ then } p_o(t+1)=0,\\
&\text{if } N(p_o(t))=\{000000001\} \text{ then } p_o(t+1)=0,\\
&\ldots\\
&\text{if } N(p_o(t))=\{111111111\} \text{ then } p_o(t+1)=0.
\end{aligned} \tag{2}$$

CA rules that contain equal number of zeros and ones, meaning that half of neighborhood combinations result in zero and other half in one, are called balanced rules. It can be proved that the image generated by the balanced CA rules has higher entropy than the image generated by the unbalanced CA rules [5].

2.2 Encryption

Encryption of image is done in three steps. First, secret information is generated: seed S is chosen as a random number in the range [0, 2^{32}], and separation values S_v are generated as an arbitrary sequence of numbers from 0 to 9 (corresponding to the number of used operations). Secret information $\{S, S_v\}$ is transmitted using secured channel. Then pseudorandom key-image is created using cellular automata and secret seed S (see Sec. 2.2.1). Finally, encryption is done by combining the key-image with the source image using pixel separation algorithm and separation values S_v (see Sec. 2.2.2).

2.2.1 Pseudorandom Key-Image Generation

The pseudorandom key-image is generated by applying CA rules on pseudorandom binary levels and combining it in a new image. Generation of the pseudorandom key-image is done using the following steps:

- Generation of pseudorandom numbers: Seed S is used to generate pseudorandom numbers K_i, i = {1, 2, …, 8}, in range [0, 33554432]. Each pseudorandom number K_i is dedicated to one binary level of a resulting key-image (index i marks dedicated binary level).
- Generation of binary levels: Each number K_i is first used to generate $M \times N$ pseudo random binary matrix $\mathbf{B}_i$ ($M \times N$ correspond to the size of the source image).
- Selection of CA rules: Number K_i is used to choose CA rule R_i for binary level i. CA rule R_i is chosen as the balanced rule nearest to the number K_i, and defined as:

  $$\begin{aligned}&R_i = 2^{K_{ib}},\\ &K_{ib} = \mathrm{nn}(K_i) \,|\, 2^{K_{ib}} = \text{balanced rule}\end{aligned} \tag{3}$$

  where nn represents the nearest neighbors function.

  In chosen CA configuration with 25 neighbors, there are 16777216 balanced rules.
- Application of CA rules: CA rule R_i is applied on $M \times N$ binary matrix $\mathbf{B}_i$ to get a new $M \times N$ binary matrix $\mathbf{B}_{ic}$. For each element in matrix $\mathbf{B}_i$, extended Moore neighborhood of 25 elements (24 neighbors around the central element) is considered to define a new value in matrix $\mathbf{B}_{ic}$ according to selected CA rule R_i.

  $$\begin{aligned}&\text{if } (\mathbf{B}_i(N(p_o)) == \text{de2bi}(R_i(l)))\\ &\qquad \text{then } \mathbf{B}_{ic}(p_o) = R_i(l)\end{aligned} \tag{4}$$

  where de2bi represents decimal to binary function.
- Key-image forming: binary images $\mathbf{B}_{ic}$, i = {1, 2, …, 8} are formed into a pseudorandom key-image K by using matrix $\mathbf{B}_{ic}$ as the i-th binary level of key-image K.

2.2.2 Pixel Separation

After generation of pseudorandom key-image *K*, encryption of image is done using pixel separation:

- Separation values S_v are used to make a new order of operations for combining source image and key-image. This is done by changing index of each operation according to secret separation values S_v.

$$
\begin{aligned}
O_{Sv(0)} &= \mathrm{xor}(I(x,y),K(x,y)),\\
O_{Sv(1)} &= \overline{\mathrm{xor}(I(x,y),K(x,y))},\\
O_{Sv(2)} &= \mathrm{xor}(\mathrm{shift}(I(x,y),2),K(x,y)),\\
O_{Sv(3)} &= \overline{\mathrm{xor}(\mathrm{shift}(I(x,y),2),K(x,y))},\\
O_{Sv(4)} &= \mathrm{xor}(\mathrm{shift}(I(x,y),4),K(x,y)),\\
O_{Sv(5)} &= \overline{\mathrm{xor}(\mathrm{shift}(I(x,y),4),K(x,y))},\\
O_{Sv(6)} &= \mathrm{xor}(\mathrm{shift}(I(x,y),6),K(x,y)),\\
O_{Sv(7)} &= \overline{\mathrm{xor}(\mathrm{shift}(I(x,y),6),K(x,y))},\\
O_{Sv(8)} &= \mathrm{xor}(\tilde{I}(x,y),K(x,y)),\\
O_{Sv(9)} &= \overline{\mathrm{xor}(\tilde{I}(x,y),K(x,y))}.
\end{aligned}
\tag{5}
$$

Note that last 2 operations involve changing of the most significant bit of pixel *I*(*x*, *y*). We introduce 10 different operations because smaller sets do not introduce enough confusion in pixel values.

- Pseudorandom key-image *K* is used to determine operation type index (*T*) for each pixel in the source image using the last digit at each location.

$$T = \mathrm{mod}(K,10)\,. \tag{6}$$

In this step, pixels are separated in ten different groups according to operation type, and we call this process pixel separation.

- Each pixel in source image *I*(*x*, *y*) is then encrypted by applying operation $O_{T(x,y)}$ on image *I*(*x*, *y*) and key-image *K*(*x*, *y*). The result of encryption is encrypted image *E*.

Advantage of this process lies in the fact that encrypted image is more difficult to decrypt, because decryption requires correct guessing of the used operation for each pixel in the source image.

2.3 Decryption

Decryption of an encrypted image is done using the following steps:

- Receiver receives secret information (seed *S* and separation values S_v) through a protected channel.
- Receiver uses seed *S* to generate pseudorandom key-image.
- Receiver uses separation values S_v to make a new order of operations for combining encrypted image *E* and key-image *K*, as described in Sec. 2.2.2. Note that the same operation set is applied in decryption process, but encrypted image *E* is used instead of source image *I* in (5).

The result of decryption is the decrypted image, which is identical to the source image.

3. Simulation Results and Analysis

Different analysis and simulations are conducted to demonstrate performance of the proposed method. To demonstrate encryption property, results are presented for the following five images: Lena.jpg, Cameraman.png, Baboon.jpg, Circle.bmp and plain image of size 512 × 512. Algorithm is developed in Matlab 2013b, and run on computer with Intel Core i7 processor under Windows 10 operation system. Simulation is applied for a random seed and separation values (as highlighted for each presented example).

3.1 Key Space Analysis

Secret information used for encryption of a source image is noted as a key. The first part of the key is seed *S*, selected randomly in range [0, 2^{32}]. Note that upper limit is set according to Matlab limit for pseudo number generator input. Therefore, probability to select the right seed is equal to:

$$p_{\mathrm{s}} = \frac{1}{2^{32}} \approx 2.32 \cdot 10^{10}\,. \tag{7}$$

Having in mind that we use 25 CA neighbors, there are $2^{25} = 33554432$ possible binary combinations of binary values 0 and 1, and $2^{33554432}$ possible rules. Probability to select the right CA rule for one binary level is:

$$p_{\mathrm{r}} = \frac{1}{2^{33554432}}\,. \tag{8}$$

The second part of the key contains separation values. There are 10! = 3628800 possible combinations to use. Possibility of using the right combination is:

$$p_{\mathrm{sv}} = \frac{1}{10!}\,. \tag{9}$$

The volume of the proposed secret key is very large due to the fact that different CA rule is selected for each binary level of the key, and can be defined as:

$$V = 10!\cdot\left[\binom{33554432}{16777216}\right]^{8} \approx 6.7301 \cdot 10^{80807099}\,. \tag{10}$$

3.2 Key Sensitivity Test

To test key sensitivity, a 512 × 512 image (Lena.jpg) is encrypted by using the seed $S_1 = 72635290$. Then, the least significant bit of the seed is changed, so that the original seed becomes $S_2 = 72635291$, and used to encrypt the same image. Finally, the two encrypted images, encrypted by the two slightly different seeds, are compared. Results are presented in Fig. 1. The difference between two images encrypted with seeds that differ in only 1 bit is quite large. This illustrates that the proposed method generates uncorrelated key-images that cannot be predicted.

We also calculated the number of pixels change rate (*NPCR*) between encrypted images E_1 and E_2 as:

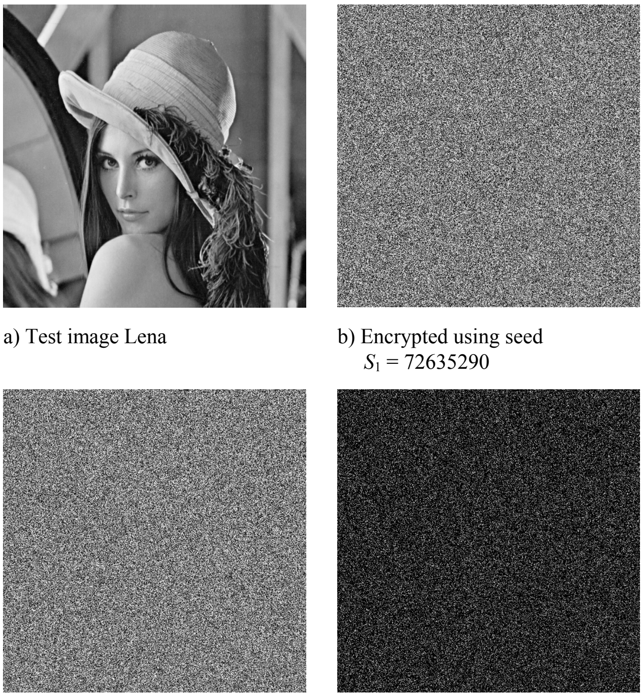
$$NPCR(E_1, E_2) = \frac{\sum_{i,j} D(i,j)}{NM} \cdot 100, \qquad (11)$$
$$D(i,j) = \begin{cases} 1 & E_1(i,j) == E_2(i,j) \\ 0 & \text{otherwise} \end{cases}$$

where N and M is the width and height of E_1 and E_2.

For two independent random images, the expected value of *NPCR* is: $E(NPCR) = (1 - 2^{L}) \cdot 100$, where L is the number of bits used to represent one pixel of the image [21]. For 8-bit per pixel gray-scale random images, $E(NPCR) = (1 - 2^{8}) \cdot 100 = 99.6094\%$. Value of NPCR in our example is equal to 99.38%, which proves that the method generates unpredictable key-images.

To measure the average intensity of the differences between the two encrypted images, the unified average changing intensity (*UACI*) is defined as:

$$UACI(E_1, E_2) = \frac{1}{NM}\left[\sum_{i,j} \frac{|E_1(i,j) - E_2(i,j)|}{255}\right] \cdot 100 \qquad (12)$$


a) Test image Lena

b) Encrypted using seed $S_1 = 72635290$

c) Encrypted using seed $S_2 = 72635291$

d) Difference between b) and c)

Fig. 1. Examples of encryption of Lena using two very similar seeds.

where N and M is the width and height of E_1 and E_2. For two 8-bit per pixel random gray-scale images, the expected value of UACI is $E(UACI) = 33.4635\%$.

Table 1 presents *NPCR* and *UACI* values for all tested images with different pair of seeds (S_1, S_2). Values of *NPCR* and *UACI* indicate that encrypted images have good randomness.

Image	Seed	$NPCR(E_1, E_2)$	$UACI(E_1, E_2)$
Lena	$S_1 = 25$, $S_2 = 24$	99.34 %	33.52 %
Cameraman	$S_1 = 3458$, $S_2 = 3459$	99.38 %	33.59 %
Baboon	$S_1 = 514887$, $S_2 = 514886$	99.31 %	33.63 %
Circle	$S_1 = 154$, $S_2 = 155$	99.40 %	33.44 %
Plain	$S_1 = 674854$, $S_2 = 674855$	99.38 %	33.37 %

Tab. 1. Pixel change rate and average intensity of the differences between two encrypted images.

3.3 Image Histogram

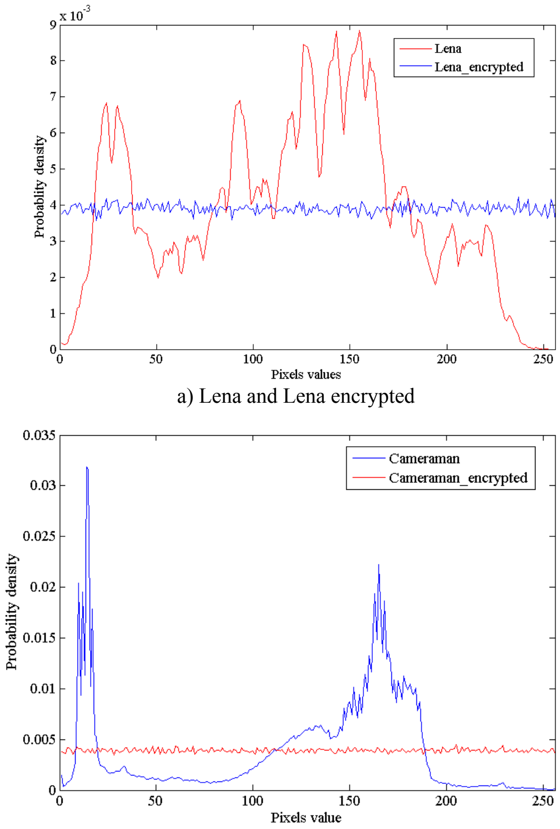
To demonstrate robustness of the proposed method, we calculated histogram and probability density function (pdf) of two original and encrypted images. In general, more uniformed histogram results in less statistical attacks. Figure 2 shows pdfs of Lena and Cameraman and their encrypted versions. Encrypted images have almost uniform pdfs regardless of the original images, so statistical attacks are very difficult to conduct.

Correlation between pixels is also a good indicator of security. The smaller correlation means the better encryption effect. To demonstrate encryption effect, we calculated the correlation coefficient as:

$$\rho(I,E) = \frac{\frac{1}{N}\sum_{i=1}^{N}(I_i - \bar{I})(E_i - \bar{E})}{\sqrt{D(I)}\sqrt{D(E)}}, \qquad (13)$$
$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

where I represents the source image, E the encrypted version of the source image, and N is the total number of pixels. The value of the correlation coefficient can vary between −1 and 1. If $|\rho|$ is near 1, images are correlated and if $|\rho|$ is near 0, there is a trivial correlation between images. Correlation coefficient for Lena and its encrypted version ($S = 72635290$), shown in Fig. 1.b), is equal to –0.0003, leading to conclusion that two images are not correlated. Furthermore, correlation coefficient of Lena and its encrypted version (Fig. 1.c) with different seed ($S = 72635291$) is –0.0053.

To better demonstrate correlation between images, we calculated correlation of pixels in vertical, horizontal and diagonal directions by randomly selecting 2500 pairs of adjacent pixels both from source image and encrypted image, and calculate the correlation coefficients. Results



a) Lena and Lena encrypted

b) Cameraman and Cameraman encrypted

Fig. 2. Probability density functions.

Image	Horizontal	Vertical	Diagonal
Lena	0.9863	0.9694	0.9643
Lena encrypted $S = 72635290$	0.0074	–0.0044	0.0053
Cameraman	0.9404	0.9448	0.9376
Cameraman encrypted $S = 1234567$	–0.0088	$-5.7\cdot10^{-4}$	$1.4\cdot10^{-5}$
Baboon	0.9801	0.9770	0.9680
Baboon encrypted $S = 24574$	–0.0072	0.0088	0.0021
Circle	0.9769	0.9826	0.9765
Circle encrypted $S = 62$	0.0042	–0.0023	0.0099
Plain	1	1	1
Plain encrypted $S = 99999$	0.0093	–0.0078	–0.0053

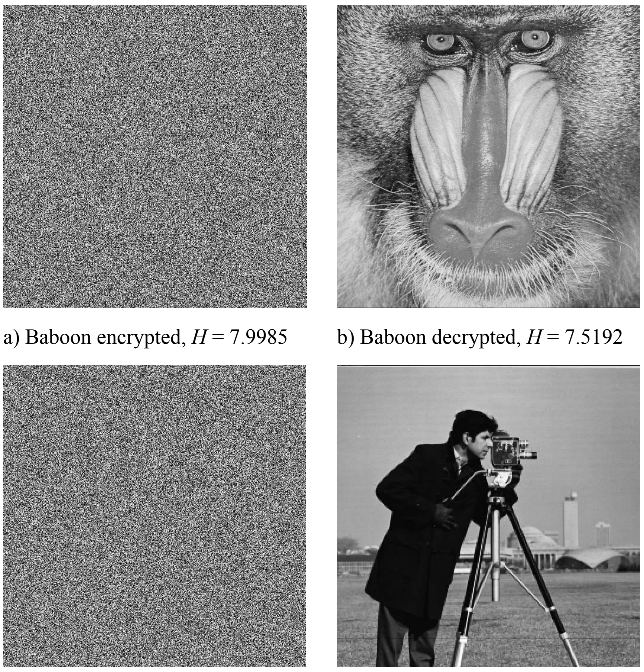
Tab. 2. Correlation between image pixels.

for five images and their encrypted version are presented in Tab. 2. The proposed algorithm effectively reduces the correlation between adjacent pixels for all tested examples. Note that correlation is especially reduced in the case of plain image.

3.4 Information Entropy

Entropy of message m is one of the important features for randomness, defined using probability of symbol $p(m_i)$:

$$H(m) = -\sum_{i} p(m_i)\log_2(p(m_i)) \,. \tag{14}$$



a) Baboon encrypted, $H = 7.9985$

b) Baboon decrypted, $H = 7.5192$

c) Cameraman encrypted, $H = 7.9980$

d) Cameraman decrypted, $H = 7.0843$

Fig. 3. Entropy of encrypted (S = 72635291) and decrypted images.

Image	Source Entropy	Encrypted Entropy
Lena	7.6962	7.9993
Cameraman	7.0843	7.9983
Baboon	7.5192	7.9992
Circle	1.7808	7.9729
Plain	0	7.3726

Tab. 3. Comparison of entropy of source and encrypted images.

For images with random pixels which are encoded by 8 bit, entropy should be equal to 8. However, entropy is usually smaller than 8, but a value closer to eight means that the possibility of predictability is less and the security level is higher.

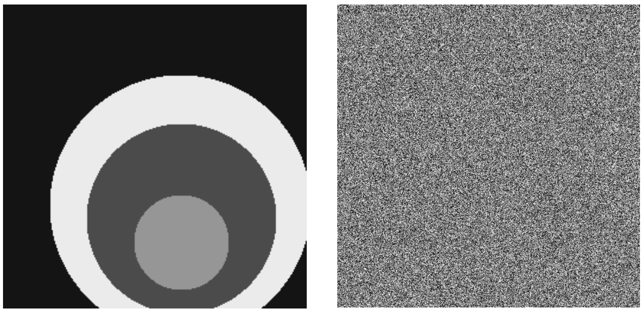
Figure 3 shows examples of image entropy for two encrypted and decrypted images. Decrypted images are identical to source images, which means that the proposed method does not influence image quality. Also entropy of encrypted images is very close to 8, which indicates their randomness.

Table 3 shows entropy of source images and their encrypted versions (seed $S = 75129$ for all examples). Encrypted images have entropy close to 8, which proves that the predictability is very low. Note that entropy is close to 8 even in the case when plain image is used as a source image.

3.5 Confusion Analysis

Confusion refers to making the relationship between the encrypted image and the private information as complex and uncorrelated as possible. We previously demonstrated the difference between two encrypted images using seeds that differ in only 1 bit (Fig. 1). An excellent encryption algorithm should be sensitive to small changes in seed. In other words, a slight change in the key should cause very different results. Figure 4 demonstrates results of decryption of encrypted Circle image with different seeds.

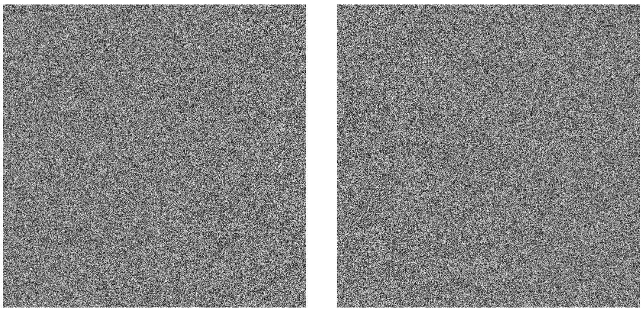
Furthermore, we simulated encryption of plain image using seed $S = 298375$. Then, encrypted image is decrypted using seed that differs in only one bit ($S = 298374$). Results are shown in Fig. 5, where it is possible to notice that the proposed method does not generate relationship between plain image and encrypted plain image. Also, since there is only 1 bit difference between the two seeds, the results show the high key sensitivity of the proposed CA encryption scheme.



a) Circle decrypted using correct seed $S = 72635290$

b) Circle decrypted using seed $S = 72635291$

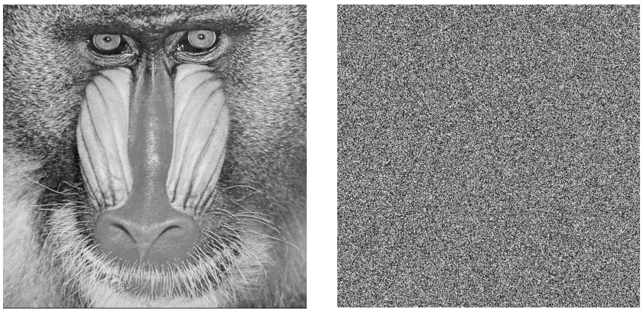
Fig. 4. Decryption of circle using seeds that differ in 1 bit.



a) Encrypted using $S = 298375$

a) Decrypted using $S = 298374$

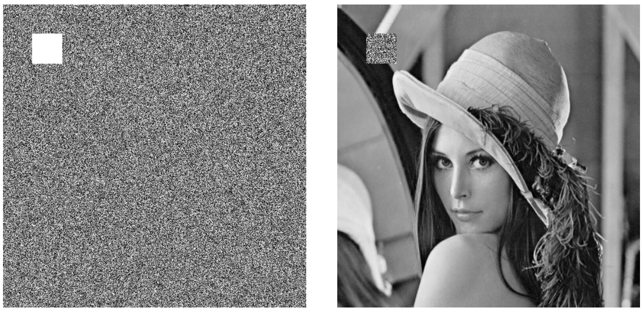
Fig. 5. Decryption of plain image using seeds that differ in 1 bit.



a) Decrypted using separation values S_v = 3 5 1 7 2 4 9 8 6 0

b) Decrypted using separation values S_v = 5 3 1 7 2 4 9 8 0 6

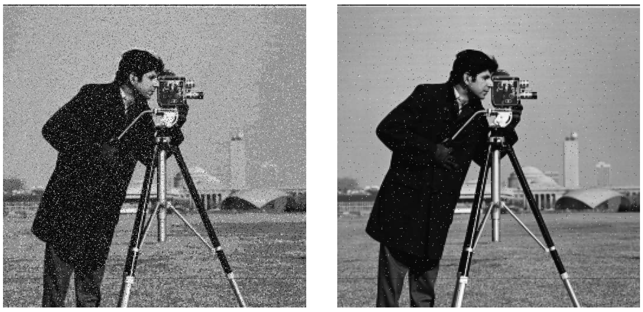
Fig. 6. Decryption of Baboon using different separation values.



a) Loss of 50 × 50 pixels

b) Decryption of a)

Fig. 7. Survival properties of the proposed method in the case of loss of the encrypted content.



a) Gaussian noise (zero mean and $\sigma^2 = 0.01$)

b) Salt and paper noise ($d = 0.01$)

Fig. 8. Survival properties of the proposed method in the case of corruption of the encrypted image.

Moreover, we tested decryption in the case when correct seed S is used, but incorrect separation values S_v are applied. Results are presented in Fig. 6, showing that small difference in separation values' order leads to inability to decrypt image even if correct seed S is used. Analysis also showed that equal percentage of operation type is applied to image pixels and that operations are distributed randomly on image.

3.6 Survival Properties

The proposed method is also robust to loss or corruption of encrypted image parts in a way that only the corresponding fraction is affected and the error does not propagate to other parts of the decrypted image. Good survival properties assure that a large area of the image survives so the method is reliable in transmissions with high error rate. Figure 7 demonstrates that loss of image's part did not affect other parts of the decrypted image.

Figure 8 shows the influence of noise on the decrypted image in the case when Gaussian noise and salt and pepper noise is applied on the encrypted image. Again, corruption of some pixels did not corrupt the whole image.

3.7 Time of Calculation

Time of calculation is an important factor for application of any method in the real time systems. Table 4

Method	**Encryption**	**Decryption**
Lena	22 ± 0.12 ms	22 ± 0.52 ms
Cameraman	21 ± 0.35 ms	21 ± 0.24 ms
Baboon	22 ± 0.47 ms	22 ± 0.38 ms
Circle	21 ± 0.61 ms	21 ± 0.19 ms
Plain	22 ± 0.03 ms	21 ± 0.83 ms

Tab. 4. Time of calculation for encryption and decryption.

contains a calculation time of the proposed encryption and decryption method for 5 test images. Note that all images have the same size and testing is done in the same conditions. Seed used for testing was set to $S = 72635290$ and separation values were set to $S_v = 3\ 5\ 1\ 7\ 2\ 4\ 9\ 8\ 6\ 0$. Values in Tab. 4 are determined as average values between 100 encryption/decryption attempts for each image. It is possible to see that calculation time is quite low and it does not depend on image content.

4. Comparison to Other Methods

The volume of the security key is much larger than the volume of the key introduced in [14], which is equal to 256. It is also much larger than 10^{9536}, $10^{2194 \times i}$, and 10^{14775} which are the lower bounds of the volume of the security keys introduced in [12], [19], and [20], respectively. Moreover, it is several times larger than the security key introduced in [21], which is defined as $H!\mathrm{e} \times 10^{10101039}$, where H is the height of the image. Key space is also much larger than chaos based key space proposed in [22], which is equal to 2^{280}. Table 5 contains comparison of key space.

By comparing the entropy with other methods, it is possible to notice that the proposed method assures higher entropy than [19], where entropy is 7.9971 and 7.9974 for Lena image and two different key-images. The value of entropy for Lena image is also higher than 7.9886 as accomplished with the method proposed in [21]. Table 6 contains comparison of image entropy values for Lena image.

Method	**Key space**
Von Neumann [12]	10^{9536}
Incomplete Moore [14]	256
Von Neumann [19]	$10^{2194 \times i}$
Von Neumann [20]	10^{14775}
Extended Moore [21]	$H!\mathrm{e} \times 10^{10101039}$
Chaos theory [22]	2^{280}
The proposed	$6.7301 \times 10^{80807099}$

Tab. 5. Comparison of key space for different methods.

Method	**Entropy**
Von Neumann [19]	7.9974
Von Neumann [20]	7.9971
Extended Moore [21]	7.9886
Chaos theory [22]	7.9992
The proposed	7.9993

Tab. 6. Comparison of image entropy for different methods.

Method	***NPCR***	***UACI***
Von Neumann [20]	99.47 %	31.82 %
Extended Moore [21]	99.77 %	33.49 %
Chaos theory [22]	99.76 %	33.35 %
The proposed	99.34 %	33.52 %

Tab. 7. Comparison of *NPCR* and *UACI* for different methods.

Values of *NPCR* and *UACI* are given in Tab. 7. The proposed method accomplished a bit lower value of *NPCR*, but it is still above the limit defined by (12). The value of *UACI* for the proposed method is higher than for other methods.

5. Conclusion

Robust and reliable image encryption is a key for effective protection in transmission through public channels. The main idea is to make images unrecognizable and assure that decryption is possible only using a secret key. The proposed method combines cellular automata and pixel separation to provide robust image encryption. It relies on secret information in the form of seed and separation values. Encryption of source image is accomplished using pseudorandom key-image, created by 2D cellular automaton. Cellular automata are good candidates for generation of pseudorandom numbers thanks to their properties like parallelism, homogeneity and unpredictability. Different CA rule is chosen for each binary level of pseudorandom key-image based on selected seed. Selection is limited to balanced rules, which results in higher entropy of key-image. CA rules are applied on extended Moore neighborhood, which results in larger key space. Finally, encryption is performed by combining source image and key-image. Pixel separation is applied to select right operation for each pixel based on secret separation values. Ten operations are defined to make decryption of an image without secret information more difficult.

Key space analysis shows that the proposed method has very large key space, which is important to avoid guessing of secret information. Key-images generated by the proposed method have good randomness property as demonstrated through sensitivity test. Randomness is proven also by calculating the entropy of encrypted images, which was very close to 8. Furthermore, image histograms are used to show almost uniform pdfs of encrypted images making statistical attacks very difficult to conduct. By calculating correlation coefficient, it is demonstrated that the algorithm effectively reduces the correlation between adjacent pixels. Survival properties in the case of data loss and noise, as well as computational time, are satisfactory for real time systems.

Comparison to the state of the art methods shows that the proposed method accomplished the largest key space volume while keeping very good randomness.

References

[1] WOLFGANG, R. B., DELL, E. J. Overview of image security techniques with applications in multimedia systems. In *Proceedings of the SPIE International Conference on Multimedia Networks: Security, Displays, Terminals, and Gateways*. Dallas (USA), 1997, p. 297–308. DOI: 10.1117/12.300900

[2] LAFE, O. Data compression and encryption using Cellular Automata Transforms. In *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*. Rockville (USA), 1996, p. 234–241. DOI: 10.1109/IJSIS.1996.565074

[3] TOMASSINI, M., SIPPER, M., PERRENOUD, M. On the generation of high-quality random numbers by two-dimensional cellular automata. *IEEE Transactions on Computers*, 2000, vol. 49, no. 10, p. 1146–1151. DOI: 10.1109/12.888056

[4] SEREDYNSKI, F., BOUVRY, P., ZOMAYA, A. Y. Cellular automata computations and secret key cryptography. *Parallel Computing*, 2004, vol. 30, no. 5–6, p. 753–766. DOI: 10.1016/j.parco.2003.12.014

[5] SARKAR, P. A brief history of cellular automata. *ACM Computing Surveys (CSUR)*, 2000, vol. 32, no. 1, p. 80–107. DOI: 10.1145/349194.349202

[6] WU, H. L., ZHOU, J. L., GONG, X. G. A novel image watermarking algorithm based on two-dimensional cellular automata transform. In *Proceedings of the 6th IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC '11)*. Chongqing (China), 2011, vol. 2, p. 206–210. DOI: 10.1109/ITAIC.2011.6030311

[7] MANKAR, V. H., DAS, T. S., SARKAR, S. K. Cellular automata based robust watermarking architecture towards the VLSI realization. *World Academy of Science, Engineering and Technology,* 2007, vol. 31, no. 8, p. 20–29.

[8] ALVAREZ, G., HERNANDEZ ENCINAS, L., MARTIN DEL REY, A. A multisecret sharing scheme for color images based on cellular automata. *Information Sciences*, 2008, vol. 178, no. 22, p. 4382–4395. DOI: 10.1016/j.ins.2008.07.010

[9] ESLAMI, Z., RAZZAGHI, S. H., ZAREPOUR AHMADABADI, J. Z. Secret image sharing based on cellular automata and steganography. *Pattern Recognition*, 2010, vol. 43, no. 1, p. 397–404. DOI: 10.1016/j.patcog.2009.06.007

[10] ESLAMI, Z., ZAREPOUR AHMADABADI, J. A verifiable multisecret sharing scheme based on cellular automata. *Information Sciences*, 2010, vol. 180, no. 15, p. 2889–2894. DOI: DOI:10.1016/j.ins.2010.04.015

[11] JIN, J., WU, Z. A secret image sharing based on neighbourhood configurations of 2-D cellular automata. *Optics and Laser Technology*, 2012, vol. 44, no. 3, p. 538–548. DOI: 10.1016/j.optlastec.2011.08.023

[12] CHEN, R. J., LU, W. K., LAI, J. L. Image encryption using progressive cellular automata substitution and SCAN. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*. 2005, vol. 2, p. 1690–1693. DOI: 10.1109/ISCAS.2005.1464931

[13] CHEN, R. J., CHEN, Y. H., CHEN, C. S., LAI, J. L. Image encryption/decryption system using 2-D cellular automata. In *Proceedings of the IEEE 10th International Symposium on Consumer Electronics (ISCE '06)*. St. Petersburg, 2006, p. 651 to 656. DOI: 10.1109/ISCE.2006.1689421

[14] HERNANDEZ ENCINAS, L., MARTIN DEL REY, A., HERNANDEZ ENCINAS, A. Encryption of images with 2-dimensional cellular automata. In *Proceedings of the 8th International Conference on Information Systems Analysis and Synthesis (ISAS '02)*. 2002, p. 471–476.

[15] HABIBIPOUR, M., MAAREFDOUST, R., YAGHOBI, M., RAHATI, S. An image encryption system by 2D memorized cellular automata and chaos mapping. In *Proceedings of the 6th International Conference on Digital Content, Multimedia Technology and Its Applications (IDC '10)*. 2010, p. 331–336.

[16] JIN, J. An image encryption based on elementary cellular automata. *Optics and Lasers in Engineering*, 2012, vol. 50, no. 12, p. 1836–1843. DOI: 10.1016/j.optlaseng.2012.06.002

[17] YU, L., LI, Y. X., XIA, X. W. Image encryption algorithm based on self-adaptive symmetrical-coupled toggle cellular automata. In *Proceedings of the 1st International Congress on Image and Signal Processing (CISP '08*). Sanya (China), 2008, p. 32–36. DOI: 10.1109/CISP.2008.48

[18] MALEKI, F., MOHADES, A., MEHDI HASHEMI, S., SHIRI, M. E. An image encryption system by cellular automata with memory. In *Proceedings of the 3rd International Conference on Availability, Security, and Reliability (ARES '08)*. Barcelona (Spain), 2008, p. 1266–1271. DOI: 10.1109/ARES.2008.121

[19] CHEN, R. J., LAI, J. L. Image security system using recursive cellular automata substitution. *Pattern Recognition*, 2007, vol. 40, no. 5, p. 1621–1631. DOI: 10.1016/j.patcog.2006.11.011

[20] CHEN, R. J., HORNG, S. J. Novel SCAN-CA-based image security system using SCAN and 2-D von Neumann cellular automata. *Signal Processing*, 2010, vol. 25, no. 6, p. 413–426. DOI: 10.1016/j.image.2010.03.002

[21] ZHANG, X., WANG, C., ZHONG, S., YAO, Q. Image encryption scheme based on balanced two-dimensional cellular automata. *Mathematical Problems in Engineering*, 2013, vol. 2013, 10 p. DOI: 10.1155/2013/562768

[22] WANG, X., LUAN, D. A novel image encryption algorithm using chaos and reversible cellular automata. *Communications in Nonlinear Science and Numerical Simulation*, 2013, vol. 18, no. 11, p. 3075–3085. DOI: 10.1016/j.cnsns.2013.04.008

[23] ROSIN, P. L. Training cellular automata for image processing. *IEEE Transaction on Image Processing*, 2006, vol. 15, no. 7, p. 2076–2087. DOI: 10.1109/TIP.2006.877040

[24] SUN, X., ROSIN, P. L., MARTIN, R. R. Fast rule identification and neighborhood selection for cellular automata. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 2011, vol. 41, no. 3, p. 749–760. DOI: 10.1109/TSMCB.2010.2091271

About the Authors ...

Dijana TRALIĆ received her PhD degree in Electrical Engineering from the University of Zagreb in 2015. Her research interests include signal and image processing, image forensics and broadcasting technologies. She is the author of 15 conference papers, one journal paper, one book chapter, and editor of 3 conference proceedings.

Sonja GRGIĆ received her PhD degree in Electrical Engineering from the University of Zagreb in 1996, where she is currently a Professor. Her research interests include image quality evaluation methods, television signal transmission, video communication technologies and image forensics. She is the author of 5 book chapters, 20 scientific journal papers, more than 140 papers published on international scientific conferences as well as of 15 reviewed studies and expert works. She was the editor of 8 international conference proceedings.