Neapolis University

HEPHAESTUS Repository **http://hephaestus.nup.ac.cy**

School of Information Sciences Articles

2014

Image Encryption using the Recursive Attributes of the eXclusive-OR Filter

Chatzichristofis, Savvas A.

Springer

http://hdl.handle.net/11728/10148

Downloaded from HEPHAESTUS Repository, Neapolis University institutional repository

Image Encryption using the Recursive Attributes of the eXclusive-OR Filter

S. A. CHATZICHRISTOFIS[1], L. BAMPIS[1], O. MARQUES[2], M. LUX[3] AND Y. BOUTALIS[1]

[1]*Democritus University of Thrace, Xanthi, Greece.*
[2]*Florida Atlantic University, Boca Raton, Florida, USA.*
[3]*Klagenfurt University, Klagenfurt, Austria.*

This paper describes a visual multimedia content encryption approach based on cellular automata (CA), expanding the work proposed in [4]. The presented algorithm relies on an attribute of the eXclusive-OR (XOR) filter, according to which, the original content of a cellular neighborhood can be reconstructed following a predefined number of applications of the filter. During the interim time marks, the cellular neighborhood is greatly distorted, making it impossible to be recognized. The application of this attribute to the field of image processing, results to a strong visual multimedia content encryption approach. Additionally, this paper proposes a new approach for the acceleration of the application of the XOR-filter, taking advantage of the Summed Area Tables (SAT) approach.

1 INTRODUCTION

In recent literature, there are numerous image encryption algorithms which may be classified according to their ability to lossless reconstruct the encrypted image or lead to loss of information after the decryption. Additionally, these methods can be classified based on the approach used to achieve the encryption, which may be divided into four categories: *SCAN*-based techniques [2] [6] [20], *Chaos*-based techniques [22] [17], *Structure*-based information [8], and finally algorithms where the encryption technique combines

elements from the other techniques or use elements borrowed from different scientific disciplines. However, every approach is characterized by advantages and disadvantages in terms of security level and effectiveness [6].

Cellular Automata (CA) have attracted researchers from several different disciplines (e.g. from the field of robotics [3, 13], image processing [24] and environmental modelling [11]) and a large number of research papers are published every year. Several image encryption methods based on cellular automata (CA) are already reported in literature. In [19] a family of basic functions, generated from the evolving states of CA, is used to encrypt multimedia information. In [23], CA were used to produce the bit stream of the key in a Vernam cipher cryptography. An image encryption method based on the permutation of the pixels of an image and the replacement of the pixel values is proposed in [6], where the permutation is done by SCAN patterns. The pixel values are replaced using a progressive CA substitution with a sequence of CA data that is generated from the CA evolution rules. Other CA image security methods are reported in [12] and [15].

A lossless visual multimedia content encryption method based on CA is presented in this paper, expanding the work presented in [4]. This algorithm is based on an attribute of the eXclusive-OR (XOR) filter, according to which, its application to a square sized CA has the ability to reconstruct the original content of the CA after a predefined number of repetitions. This attribute is presented in details in Section 2.

The presented method is a symmetric-key based one. Detailed description of the key construction as well as its use in encrypting and decrypting images, is presented in Section 3. Section 4 describes the encryption characteristics and evaluates several security issues regarding the application of the presented method. Additionally, Section 5 proposes a new method, suitable for simplifying and accelerating the filtering process. Finally, the conclusions are drawn in Section 6.

2 THE RECURSIVE ATTRIBUTE OF THE XOR FILTER

A single channel image (i.e. a grayscale or a binary image) can be considered as a CA which is comprised of a linear two-dimensional (2D) table of identical cells. Every pixel of the image corresponds to a cell of the CA and each one of these cells can be found in k different states. The local state of the cell x, y during time step t is given by the formula:

$$S_t^{x,y} \in \sum = \{0, 1, \ldots, k-1\} \tag{1}$$

where $k = 256$ for grayscale images and $k = 2$ for binary ones.

The total state of a cell during time t, designated as s_t, is the configuration of the whole table:

$$S_t = (S_t^{0,0}, S_t^{0,1}, S_t^{0,2}, \ldots, S_t^{W-1,H-1}) \in \sum^{W \times H} \tag{2}$$

where W, H designate the width and height of the 2D CA respectively.

At every time step all the cells of the CA recalculate their state concurrently according to the following rule: The values of the cells in the Moore neighborhood with radius $r = 1$, participate in the logical operation (XOR) and their result is placed in the central cell of the neighborhood. It has been reported in the literature [7] [10] [5] [14] that after applying the filter t times, the CA returns to its original state, i.e. $S_t^{x,y} = S_0^{x,y}$. In other words, the original content of a cellular neighborhood can be reconstructed after a predefined number of applications of the filter. The reconstruction periodicity is based strictly on the size of the CA and on the **boundary conditions**. The presented encryption method is based on the reconstruction periodicity which occurs under two different conditions:

Condition 1 *The application of an XOR filter on a CA of $N \times N$ dimensions and* periodical *boundaries has the ability to lossless reconstruct the CA after $t = N/2$ repetitions when $N = 2^p$ and $p \in Z$.*

Periodical boundaries may be visualized as taping the left and right edges of the rectangle to form a tube, then taping the top and bottom edges of the tube to form a torus (donut shape).

Condition 2 *The application of an XOR filter on a CA of $N' \times N'$ dimensions and* pseudo-cells *boundaries has the ability to lossless reconstruct the CA after $t = 2^p \times 2$ iterations when $N' = 2^p - 1$ and $p \in Z$. In other words, the application of an XOR filter on a CA can reconstruct the CA after $t = (N' + 1) \times 2$ steps.*

In this case of Pseudo-Cells boundaries, the data forming the boundaries belong to hypothesized neighboring cells. In other words, the $N' \times N'$ CA is regarded as a part of another $N' + 1 \times N' + 1$ CA, to the boundary conditions of which the filter is not applied. An application of a CLF (Coordinate Logic Filter)-XOR filter to a 2D CA of 3×3 ($p = 2$). It is crucial to emphasize that under both conditions, during interim time steps, the CA content and consequently the single channel image is greatly distorted in an almost unpredictable manner.

In case of color images, each channel is processed individually. More specifically, each channel is considered as a 2D CA. Hence, for a typical RGB image, 3 CAs of identical dimensions are generated. The cell values are integer numbers in $[0, 255]$, which are subsequently converted into a binary form. The XOR filter is applied on every cell of every CA, which is actually an application of the filter to every pixel of each channel of the image. Essentially, this filter executes the XOR operation to the binary form values of the 9 cells that belong to the Moore neighborhood on which the filter is applied. The application of the filter to the nine 8-bit values results in an 8-bit number, and the operation outcome is placed in the central cell of the neighborhood. This procedure is repeated for every pixel/cell of every channel.

The exact same procedure should be followed when the filter is applied more than once, with new values being generated for all the CA cells after each iteration. These new values constitute the basis for the next iteration and so on. It is worth noting that, after few repetitions, the image is greatly distorted, making it impossible to be recognized. Also, it is important to note that, the fewer the regularities a picture depicts, the less iterations are needed in order to distort the content of the image. In case of real-world images, less than 5 iterations are enough to completely disfigure the visual content. On the other hand, in case of single-color images, the presented method cannot be applied.

3 THE PRESENTED IMAGE ENCRYPTION SCHEME

Let the plaintext image be of $N \times N$ dimensions, with $N = 2^p, \quad p \in \mathcal{Z}$. If this is not the case, the image should be modified by adding extra random generated pixels as surrounding frame, in order to get the desired dimensions. Initially, each channel of the plaintext image is regarded as an independent CA, to each of which the XOR filter is applied $K[i]$ times, with $i \in [0, 2]$. Each value of i (i.e. $i = 0$, $i = 1$, and $i = 2$) corresponds to a channel of the plaintext image (i.e. red, green, and blue channel, respectively). The $K[i]$ values are generated via the random number generator proposed in [18] and belongs to $K[i] \in [0, N/2]$.

In the sequence, an integer pseudo random number Q, where $Q \in [4, Y]$, $2^Y \leq N/2$, is generated, defining the number of R sub-images that the $K[i]$ times filtered plaintext image will be divided to. Every sub-image is of $N' \times N'$ dimensions, with: $N' = 2^Q - 1$.

Given that MOD $\left(\frac{N}{N'}\right) > 0$, necessarily, some sub-images will overlap. The manner via which the plaintext image is divided may be seen in Figure

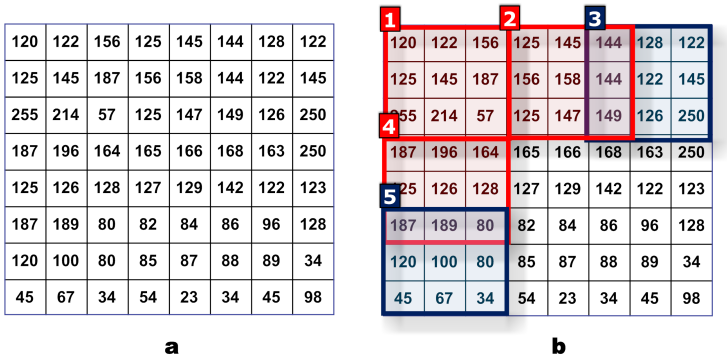

FIGURE 1
Separating the plaintext Image into Sub Images

1. Note that although $Q > 4$, so that the smallest sub-image possible being of dimensions 15×15, in the given example the values $N = 8$ and $Q = 2$ are selected for ease of understanding. Hence, the dimensions of each sub-image are $N' \times N' = 3 \times 3$. As may be seen in Figure 1, the top-right and bottom-left sub-images (designated as '3' and '5'), are borrowing pixels from the neighboring sub-images

CAs can be categorized into two categories: Linear CAs and non-linear ones. If a rule of a CA employs XOR filter only then it is labelled a linear rule and the corresponding CA is called a linear one. Adopting a linear rule in a cryptographic method is generally concerned as a potential weakness [9] [16]. In order to enhance the strengtheners of the cryptographic primitive and reduce the potential linearity drawbacks, the presented method adopts the following strategy: Each R sub-image is regarded as an 'independent' CA, with its boundaries being assumed as pseudo-cells. When applying the XOR filter to the sub-images' CA, the values of the cells of the neighboring CA participate to the boundary conditions. The random number generator produces R values in $[0, 2\times(N'+1)]$ which are stored in the array $RA[i], i \in [0, R-1]$, as well as R values in $[0, N/2]$ which are stored in array $RB[i], i \in [0, R-1]$. In every sub-image i, the XOR filter will be applied $RA[i]$ times. Furthermore, the strengtheners of the presented method is reinforced by randomly changing the order in which the sub-images will be filtered. For example, assume that a grayscale plaintext image C is divided into 4 sub-images $C[i], i \in [0, 3]$ of

dimensions $N' \times N'$. Assume also that every sub-image is about to be filtered $RA[i]$ times. Even if $RA[0] = RA[1] = RA[2] = RA[3]$, the result depends on the order in which the sub-images are filtered, because each filtering of every sub-image utilizes information of the current state of the neighboring sub-images. Furthermore, it is worth noting that in order to reconstruct every sub-image, it has to be filtered $(2 \times (N' - 1)) - RA[i]$ times, following the opposite order of the initial $RA[i]$ filter applications. For example, when an XOR filter is applied $RA[0]$ times to sub-image $C[0]$, then $RA[1]$ times to $C[1]$, then $RA[2]$ times to $C[2]$ and finally $RA[3]$ times to $C[3]$, in order to reconstruct the sub-images' content, the XOR filter must be applied $(2\times(N'-1))-RA[3]$ times to $C[3]$, then $(2\times(N'-1))-RA[2]$ times to $C[2]$, then $(2 \times (N'-1)) - RA[1]$ times to $C[1]$, and finally $(2 \times (N'-1)) - RA[0]$ times to $C[0]$.

A random generated number defines which space ordering method will be used. Some common examples of space ordering methods are (a) the raw order, (b) the row prime, (c) the Morton order, (d) the Peano Hilbert order, (e) the Cantor diagonal order, (f) the spiral order, (g) the Gray order, (h) the double gray order, (i) the U order, (j) the Z-Order, and so on [21]. Of course, any other ordering method could be utilized, given that it is known to both the sender and the receiver side. Finally, after the $RA[i]$ filter applications on the i sub-image, the whole image is filtered $RB[i]$ times. This way, the neighboring cells states are even more altered. The procedure is completed when all the sub-images are filtered. By adopting this approach, the state of a filtered CA does not only relies on its previous state, but also takes into account the state of its neighbouring CAs, the RA values as well as the order in which the sub-images are filtered.

The key for encrypting a single channel image is divided into the following parts. The first part is an integer number $K \in [0, N/2]$ which describes the number of XOR filter applications to the plaintext image. The second part describes the size N' of every sub-image, whereas the third part is an integer F designating the utilized space ordering method.

The final part of the key includes R values $RA[i], i \in [0, R-1]$, describing the number of the XOR filter applications to every sub-image, as well as R values $RB[i], i \in [0, R-1]$ for the number of filter applications to the plaintext image, after every sub-image filtering. In case of color images, 3 such keys are required, one for every channel of the image. Note that to increase the encryption security, different values for every part of each key may be selected.

Given that the length of the key alone, may reveal crucial part of the in-

formation that an attacker would need to decrypt the image, the key may be modified as follows. Although the number of sub-images equals R, the key may include $\frac{N}{15} + 1$ values to the arrays RA and RB (the value 15 corresponds to the smallest sub-image size possible). From all the values in the arrays RA and RB, only the first R correspond to the actual times the XOR filter was applied to the sub-images and the plaintext image, whereas the rest of the values are random numbers generated. Thus, for same sized images, the key length is the same irrespectively of the size of the sub-images. The procedure of exchanging the key is not a feature of the presented method and it is omitted.

For decrypting the image, the receiver follows the exact opposite procedure. The first retrieved element from the key is the N' value, so that the size of the sub-images is recognized, and subsequently identify which of the RA and RB values are included in the key are useful. Then the XOR filter is applied $(N/2 - RB[i])$, $i \in [R, 0]$, times to the encrypted image, and then $(2 \times (N' + 1) - RA[i])$, $i \in [R, 0]$, times to the sub-image that is defined by the ordering method – which is known via the F value of the key. It is crucial to note that during decryption, the inverse ordering of the selected space ordering method is followed. Finally, it should be reminded that the presented method reconstructs the plaintext image in a lossless manner.

4 STATISTICAL ANALYSIS

Initially, the resistance of the method against a brute force attack is evaluated. Brute force attack is a trial and error method in which every possible combination of characters against the encrypted data is tested in an attempt to retrieve the key. Suppose an adversary possesses a cipher image with $N \times N$ dimensions, with $N = 512$. To begin with, a random value $Q \in [4, 8]$, $2^8 \leq 256$ is chosen by the adversary, say $Q = 4$. Thus the adversary assumes that the image is divided into 4 sub-images of dimensions $512/(2^Q - 1) \times 512/(2^Q - 1)$. There are 1225 sub-images in total. Then, for every sub-image $16 \times 2 = 32$ combinations are tested. For the sake of simplicity, suppose that the plaintext image is NOT filtered by K times and that after filtering each sub-image, the entire image is NOT filter RB times. At the beginning, the adversary would apply the filter up to 32 times to every sub-image. In one of those trials the sub-image content would be reconstructed (given that Q is the correct one), and thus the 1/1225 of the image would have been revealed. If no sub-image would have been reconstructed, the adversary would choose a different Q. After revealing the first sub-image, the adversary has to unlock the 8 neigh-

boring sub-images identifying the utilized space ordering method as well - with finite complexity. Then, following the inverse ordering, and after only up to 32 iterations per sub-image, the whole image is reconstructed.

By initially filtering the plaintext image K times, it is impossible for the adversary to realize that a sub-image is reconstructed, because its original state is the K times filtered state of the plaintext. Hence the only possible way to reconstruct the content is the following: select randomly a space ordering method and locate the last sub-image and its neighboring 8 sub-images. Adversary's goal would be to test all the possible RA combinations to this neighborhood, and then for each one of these combinations to test the $N/2 = 256$ filtering iterations to the whole image. Once more, in this scenario it is supposed that after filtering a sub-image, the whole image is NOT RB times filtered. The complexity in this case is defined as: $256 \times 9^{32} = 8.8 \times 10^{32}$. Of course, the correct space ordering method should have been chosen at first place. Furthermore, note that there is no guaranty that a part of the plain text image is truly revealed even if the RA values of the neighborhood are identified, in case the rest of the sub-images are not reconstructed as well.

The application of the filter RA times to every sub-image, is on the one hand, a time consuming procedure, but on the other hand, intensively increases the complexity of the approach. Based on this attribute, it is impossible to isolate a neighborhood of sub-images, due to the fact that the population of the neighborhood changes. Furthermore, given that the whole procedure should be followed for every channel of the image, it is strongly believed that a brute force attack would have no result at reconstructing the plaintext image.

To distinguish the difference between the plaintext image and the encrypted one, initially, the function of peak signal to noise ratio (PSNR) is adopted. Although PSNR is commonly used as a measure of quality of reconstruction of non-lossless encryption methods, it is a very strong tool for describing image distortion as well (estimate the distortion level of an image submitted to noise). This approach has been adopted from several image encryption approaches so as to demonstrate that the PSNR results of the encrypted image (in relation with the plaintext image) are similar to the PSNR values of random generated images. This result demonstrates that the encrypted image can be regarded as a randomly generated one, reinforcing the observation that, visually, no information about the plaintext image can be revealed. PSNR is usually expressed in logarithmic decibel scale. When using 8 bit images the greatest difference between images is calculated at 0 dB, whereas for matching images the value tends to infinity. To assess the method, the

experiment proposed in [5] is conducted for 3 artificially constructed images. The method is applied to 3 real-world images which are well known in the literature (known as 'Lena', 'Lady' and 'Fruits'). Then, the PSNR between the plaintext and the encrypted images is calculated as well as the PSNR between the plaintext and the randomly generated images. The results per channel are presented in Table 1.

TABLE 1
PSNR Values (in dB) for the Encrypted and Random Generated Images (R/G/B)

	Lena	Lady	Fruits
Random Image 1	6.87/6.98/6.54	7.08/6.41/6.91	7.24/7.30/6.94
Random Image 2	7.54/7.01/6.75	6.24/6.80/6.37	6.44/6.11/6.71
Random Image 3	6.42/6.10/6.26	6.96/6.37/6.92	7.42/7.36/6.81
Encrypted Lena	7.78/7.97/7.90	7.21/7.45/7.12	8.11/8.43/8.52
Encrypted Lady	8.89/8.68/8.65	8.69/8.33/8.12	8.06/8.70/8.27
Encrypted Fruits	7.04/7.69/7.58	6.91/7.16/6.83	7.51/7.22/7.67

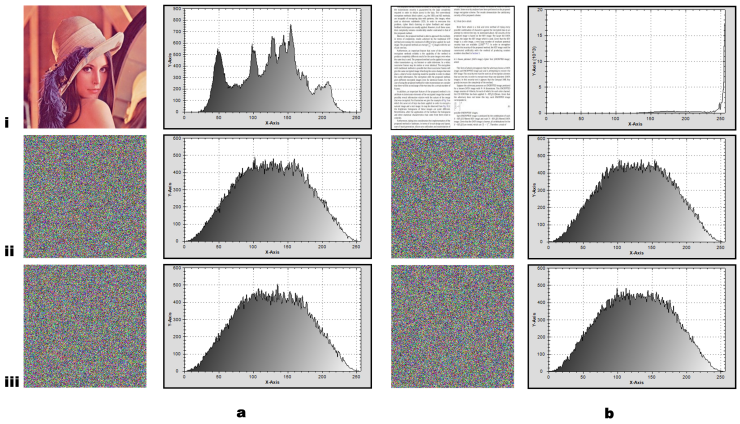

FIGURE 2
Encrypting (a) an image with natural content and (b) an image depicting a document using the same key.

As may be seen, the PSNR results for the encrypted images and the random generated images are similar. This fact suggests that the encrypted images are

as distorted as the artificially generated ones, and do not reveal information about the plaintext image content.

TABLE 2
Histograms Statistical Characteristics

	Mean	Median	Std. Dev.	Max. Prob.
Lena	123.56	128	47.678	154
Encrypted Lena 1	127.02	127	49.444	102
Encrypted Lena 2	127.22	127	49.367	135
Document	228.16	250	35.414	255
Encrypted Document 1	127.15	127	49.241	134
Encrypted Document 2	127.21	127	49.347	112

An important feature of the presented method is its attribute to deteriorate elements of the encrypted image that would possibly reveal information relative with the nature of the image that was encrypted. The example of Figure 2 is provided for illustration of this fact, where the same key has been applied in order to encrypt a natural image (Lena) Figure 2(a) and a grayscale image depicting a document Figure 2(b). As may be observed, the brightness histograms of these images are quite different. Nevertheless, after the application of the method, the resulting histograms tend to coincide. The experiment was repeated twice for every image whereas some characteristics of the image histograms are presented in Table 2. Evaluating the results in Table 2, as well as the histograms of the encrypted images, it is easily seen that in both cases the encrypted images tend to coincide. Consequently, it is impossible for an attacker to identify the nature of the image that was encrypted.

5 ACCELERATING THE XOR FILTER APPLICATION

In this section we propose the use of the Summed Area Tables (SAT) [1] approach for the manipulation of the CLF-XOR image processing part of the algorithm. The CLF-XOR filters are applied multiple times on 3×3 pixels image areas, several times during the presented method. Thus, accelerating the CLF-XOR filter application introduces great overall performance improvement.

A SAT is a two dimensional array of partial sums of a given two dimensional initial array. When the SAT is generated, it can be used to quickly

calculate the summation of values of any rectangular subset of the initial array.

The efficient approach to create the SAT $S_{i,j}$ of an initial array $Y_{i,j}$ is described by the following formula

$$S_{i,j} = Y_{i,j} + S_{i-1,j} + S_{i,j-1} - S_{i-1,j-1} \tag{3}$$

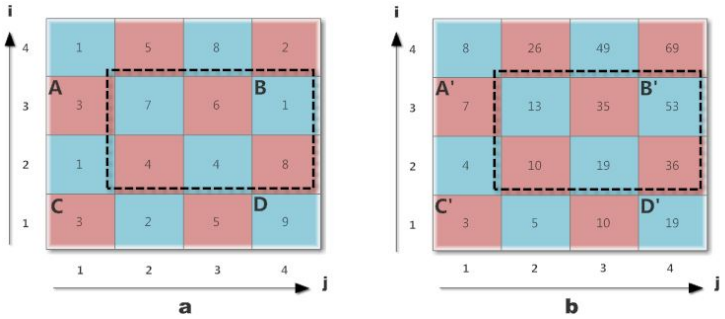

FIGURE 3
(a) Initial Table, (b) Summed Area Table.

For the generation of the SAT, the method calculates consecutive cell values row by row. Starting from the bottom left cell of both arrays the $S_{1,1}$ equals $Y_{1,1}$. The rest of the SAT values can be calculated using the equation 3 replacing with zero the values that fall out of boundaries. A visual example of a SAT is depicted in Figure 3 where (a) is the initial array and (b) the corresponding SAT. With the Summed Area Table calculated, any rectangle area of values can be evaluated with just four array references. The noted rectangle area of values in Figure 3(a) is summed by:

$$\text{sum}_{\text{Noted\_Area}} = S(B') - S(A') - S(D') + S(C') \tag{4}$$

It is worth noting that, S values that fall out of SAT boundaries are considered zero valued. The SAT method has been widely used for the purposes of image processing in several methods presenting computational advantages, especially for cases of summing pixel values of overlapping regions. In our case, even though the method does not contain any summation process, the SAT structure can be employed for the XOR filter application. This application is a two-step procedure. The first step includes the creation of the XOR

Area Table (XOR-AT). As before, the new table (XOR-AT) serves as a quick reference not for summed area values but for XOR filtered area values.

The formulas that govern this table are presented below. The summation and the subtraction are replaced with the XOR operation and the equation for the evaluation of the values can be modified to:

$$X_{i,j} = Y_{i,j} \oplus X_{i-1,j} \oplus X_{i,j-1} \oplus X_{i-1,j-1} \tag{5}$$

After obtaining the XOR Area Table, filtering any rectangular area with the XOR operator can be executed using only four array references. In our method, these areasare constituted of nine values (Moore) and the corresponding formula can be adjusted as follows:

$$\text{XOR}_{i,j} = X_{i+1,j+1} \oplus X_{i+1,j-2} \oplus X_{i-2,j+1} \oplus X_{i-2,j-2} \tag{6}$$

As before, X values that fall out of XOR-AT boundaries are considered zero valued. In conclusion, while the straightforward approach requires 8 XOR operations per cell (Moore), the proposed two-stepped method reduces the needed XOR operations to a total of six (3 in the first and 3 in the second step). Thus, employing the SAT-inspired method to perform the XOR filtering leads to a 25% acceleration, compared to the straightforward XOR filtering approach.

6 CONCLUSIONS

This paper presents a visual multimedia content encryption method employing the recursive attributes of the XOR filter on cellular automata. The decryption result is a lossless representation of the original encrypted image. Experimental results have shown that the resulted encrypted images do not contain statistical information able to reveal the source from which they originate (i.e. the plaintext image). Moreover, the presented method appears to be able to withstand brute force attacks.

REFERENCES

[1] Bay, H., Ess, A., Tuytelaars, T., Gool, L.J.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding 110(3), 346–359 (2008)

[2] Bourbakis, N., Alexopoulos, C.: Picture data encryption using scan patters. Pattern Recognition 25 (6), 567–581 (1992)

[3] Charalampous, K., Amanatiadis, A., Gasteratos, A.: Efficient robot path planning in the presence of dynamically expanding obstacles. In: ACRI. Lecture Notes in Computer Science, vol. 7495, pp. 330–339. Springer (2012)

[4] Chatzichristofis, S.A., Marques, O., Lux, M., Boutalis, Y.S.: Image encryption using the recursive attributes of the exclusive-or filter on cellular automata. In: ACRI. vol. 7495, pp. 340–350. Springer (2012)

[5] Chatzichristofis, S.A., Mitzias, D.A., Sirakoulis, G.C., Boutalis, Y.S.: A novel cellular automata based technique for visual multimedia content encryption. Optics Communications 283(21), 4250 – 4260 (2010)

[6] Chen, R.J., Lu, W.K., Lai, J.L.: Image encryption using progressive cellular automata substitution and scan. In: IEEE International Symposium on Circuits and Systems ISCAS 2005. pp. 1690–1693 (2005)

[7] Chen, R., Lai, J.: Image security system using recursive cellular automata substitution. Pattern Recognition 40(5), 1621–1631 (2007)

[8] Chung, K., Chang, L.: Large encrypting binary images with higher security. Pattern Recognition Letters 19(5-6), 461–468 (1998)

[9] Das, S., Chowdhury, D.R.: Generating cryptographically suitable non-linear maximum length cellular automata. In: ACRI. pp. 241–250 (2010)

[10] Dasgupta, P., Chattopadhyay, S., Chaudhuri, P., Sengupta, I.: Cellular automata-based recursive pseudoexhaustive test patterngenerator. IEEE Transactions on Computers 50(2), 177–185 (2001)

[11] Georgoudas, I.G., Sirakoulis, G.C., Scordilis, E.M., Andreadis, I.: A cellular automaton simulation tool for modelling seismicity in the region of xanthi. Environmental Modelling and Software 22(10), 1455–1464 (2007)

[12] Guan, P.: Cellular automaton public-key cryptosystem. Complex Systems 1, 51–56 (1987)

[13] Ioannidis, K., Sirakoulis, G.C., Andreadis, I.: Cellular automata-based architecture for cooperative miniature robots. Journal of Cellular Automata 8(1-2), 91–111 (2013)

[14] Jin, J., hong Wu, Z.: A secret image sharing based on neighborhood configurations of 2-d cellular automata. Optics & Laser Technology 44(3), 538 – 548 (2012)

[15] Kari, J.: Cryptosystems based on reversible cellular automata. Personal communication (1992)

[16] Karmakar, S., Chowdhury, D.R.: Nocas : A nonlinear cellular automata based stream cipher. In: Automata. pp. 135–146 (2011)

[17] Koduru, S., Chandrasekaran, V.: Integrated confusion-diffusion mechanisms for chaos based image encryption pp. 260–263 (2008)

[18] Kotoulas, L., Tsarouchis, D., Sirakoulis, G., Andreadis, I.: 1-d cellular automaton for pseudorandom number generation and its reconfigurable hardware implementation p. 4 (2006)

[19] Lafe, O.: Data compression and encryption using cellular automata transforms. Engineering Applications of Artificial Intelligence 10(6), 581–592 (1997)

[20] Maniccam, S.S., Bourbakis, N.G.N.: Image and video encryption using scan patterns. Pattern Recognition 37(4), 725–737 (2004)

[21] Samet, H.: Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005)

[22] Scharinger, J.: Fast encryption of image data using chaotic kolmogorov flows. Electronic Imaging 17 (2), 318–325 (1998)

[23] Seredynski, F., Bouvry, P., Zomaya, A.: Cellular automata computations and secret key cryptography. Parallel Computing 30(5-6), 753–766 (2004)

[24] Zagoris, K., Pratikakis, I.: Scene text detection on images using cellular automata. In: ACRI. pp. 514–523 (2012)