RGB Digital Image Forgery Detection Using Singular Value Decomposition and One Dimensional Cellular Automata

Ahmad Pahlavan Tafti
Mohammad V. Malakooti

Department of Computer Engineering
IAU, UAE Branch
Dubai, United Arab Emirates.

NCM2012

IEEE
8th International Conference on
Computing and Advanced Information Management
April 24 ~ 26, 2012 Seoul, Korea

Titles:

- Abstract
- Introduction To Digital Image Forgery Detection Techniques
- Singular Value Decomposition
- Cellular Automata
- Proposed Algorithm
- Experimental Results and Discussion
- Conclusion

Abstract

- A reliable approach for RGB digital image forgery detection is presented in this paper.
- Our method is based on the Singular Value Decomposition (SVD) and Cellular Automata.
- We applied the algorithm on hundreds of 800 × 800, RGB digital images.
- The experimental results have illustrated the robustness, visual quality and reliability of our proposed algorithm.

Introduction To Digital Image Forgery Detection Techniques

Digital media like digital images and documents should be authenticated against the forgery due to availability of powerful tools in the field of editing and manipulating these media (H. Farid, March 2009) .

Forged Image

Original Image

Figure 1. Forged Images; Example

Introduction To Digital Image Forgery Detection Techniques

Forged Image

Original Image

Figure 2. Forged Images; Example

Forged Image

Original Image

Figure 3. Forged Images; Example

Introduction To Digital Image Forgery Detection Techniques

Digital Image Forgery Detection Techniques

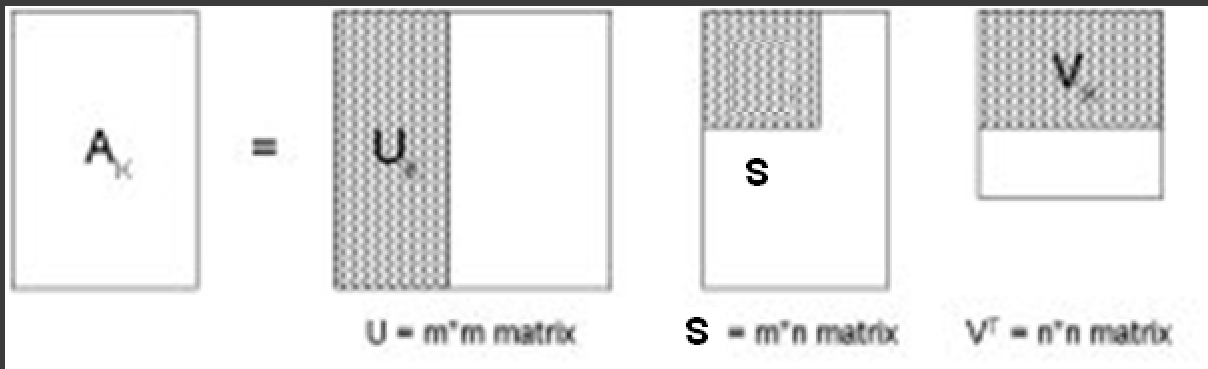
- **Active Approaches**
  - **Digital Data Embedding**
  - **Digital Signature**
- **Passive Approaches**
  - **Pixel-Based Methods** (Cloning, Resampling, Splicing, Statistical)
  - **Format-Based Methods** (JPEG Quantization, Double JPEG, JPEG Blocking)
  - **Physically-Based Methods** (Light Direction 2-D,3-D, Light Environment)
  - **Camera-Based Methods** (Chromatic Aberration, Color Filter Array, Sensor Noise, Camera Response)
  - **Geometric-Based Methods** (Principal Point, Metric Measurement)

Active Approach, with digital data embedding is used in our algorithm.

Singular Value Decomposition

SVD is used to decompose any square or non-square digital image matrix into three orthogonal matrices that contain the useful features of the image.

SVD can help us to select the dominant features in a digital image (K. Z. Mao, 2005)



Singular Value Decomposition

The SVD can decompose any real or complex n × p matrix into product of three matrices, an orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V as following:

$$A_{n\times p} = U_{n\times n}\, S_{n\times p}\, V^T_{pxp}. \quad (1)$$

where U and V are Orthogonal Matrices ;i.e.

$$U^TU = UU^T = I_{n\times n}. \quad (2)$$

and

$$V^TV = VV^T = I_{pxp} \quad (3)$$

Where the columns of U are called the Left Singular Vectors (Orthogonal Eigenvectors of AA^T), S (the same dimensions as A) a diagonal matrix that has the Singular Values (the Square roots of the Eigen values of AA^T or A^TA), and the columns of V called the Right Singular Vectors (Rows of V^T, Orthogonal Eigenvectors of A^TA) (Alter O, 2000).

Singular Value Decomposition

$$A=\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad (4)$$

$$B\,X = \lambda\,X \longrightarrow (B - \lambda\,I)\,X = 0 \qquad (5)$$

$$\text{We have } B = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad (6)$$

$$\lambda_1 = 3, \quad \mathbf{v}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad (7)$$

$$\lambda_2 = 1, \quad \mathbf{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad (8)$$

Singular Value Decomposition

The Singular values can be calculated as:

$$\sigma_1 = \|\mathbf{A}\mathbf{v}_1\|_2 = \sqrt{\lambda_1} = \sqrt{3}; \tag{9}$$

$$\sigma_2 = \|\mathbf{A}\mathbf{v}_2\|_2 = \sqrt{\lambda_2} = 1 \tag{10}$$

Thus we can immediately calculate u1, u2.

$$\mathbf{u}_1 = \frac{1}{\sigma_1}\mathbf{A}\mathbf{v}_1 = \frac{1}{\sqrt{6}}\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \tag{11}$$

$$\mathbf{u}_2 = \frac{1}{\sigma_2}\mathbf{A}\mathbf{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \tag{12}$$

u3 must be selected such that to be orthogonal to both u1, u2. Thus, it can be written as:

$$\mathbf{u}_3 = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \tag{13}$$

Therefore, A can be decomposed as products of three matrices:

$$\begin{bmatrix} 1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \\ 1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ 2/\sqrt{6} & 0 & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

Cellular Automata

A cellular automaton (CA) is a discrete model studied in computability theory, mathematics, physics, complexity science, theoretical biology and microstructure modeling. It consists of a regular grid of cells, each in one of a finite number of states, such as "On" and "Off ".The grid can be in any finite number of dimensions **(J.P. Renard, 2002).**

For each cell, a set of cells called its neighborhood is defined relative to the specified cell. For example, the neighborhood of a cell might be defined as the set of cells a distance of 2 or less from the cell. An initial state (time t=0) is selected by assigning a state for each cell. A new generation is created (advancing t by 1), according to some fixed rule (generally, a mathematical function) that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood **(J.P. Renard, 2002).**

Cellular Automata

Fig.4 shows a simple two state and one dimensional cellular automata with a line of cells. A specific cell can be either be on (value = 1) or off (value = 0). The closest cells to cell X are those to its immediate right and left. In this figure, we have a local neighborhood of three cells. The state of X at the time t + 1 will be determined by the states of the cells within its neighborhood at the time t.

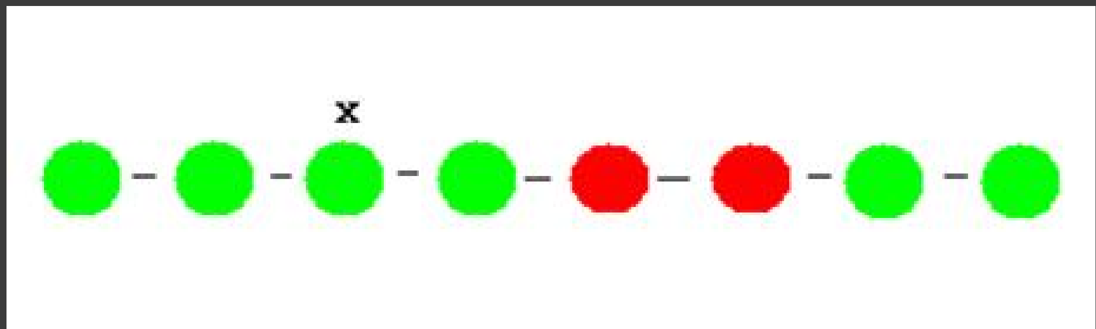


Figure 4. One Dimensional Cellular Automata

Cellular Automata

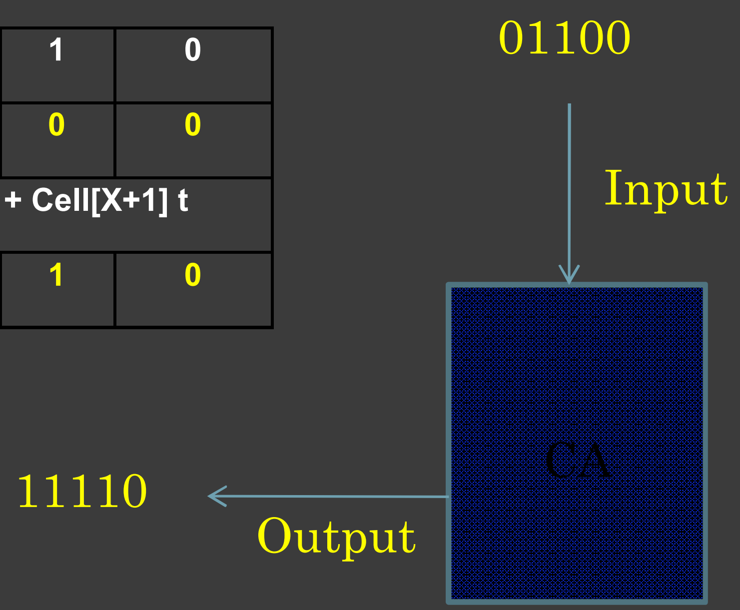
We can set a local rule for each cellular automata. For example, we can estimate the value of cell X in time t+1 with the following rule:

Cell[X] t+1 = Cell[X-1] t + Cell[X+1] t

Assume that the input sequence is 01100 and we want to use the above rule for our cellular automata, then the output sequence will be 11110. Table 2.1. shows the output of these cellular automata.

Cell Number	4	3	2	1	0
Input Sequence (time t)	0	1	1	0	0
Cellular Automata Rule	Cell[X] t+1 = Cell[X-1] t + Cell[X+1] t				
Output Sequence (time t+1)	1	1	1	1	0

TABLE 1: AN EXAMPLE OF CELLULAR AUTOMATA



Cellular Automata

Cellular Automata Usages:

- Multimedia Content Generation
- Artificial Life
- Modeling and Simulation
- Cryptography
- Random Number Generation, Cipher Key Generation

Proposed Algorithm

•The main idea of our proposed algorithm is to create a robust secret key and embed it in the LSB of a layer of the original image, to protect it against forgery.

•Our proposed method is based on active approaches in which some data or secret key is embedded in to the spatial domain of the original image for the authentication.

•We have implemented our algorithm on a set of one hundred images and calculated the Singular Values, Right and Left Singular Vectors of the original image and pushed the SVD features into one dimensional cellular automata to generate the secret key.

Proposed Algorithm

Cell Number	Input Value (time *t*)	Rule
0	Sum of all singular values of Original Image (Red Matrix).	$Cell[X]_{t+1} = Cell[X-1]_t$ XOR $Cell[X+1]_t$
1	Mean of all singular values of Original Image (Red Matrix)	
2	Sum of all right singular vectors of Original Image (Red Matrix).	
3	Mean of all right singular vectors of Original Image (Red Matrix).	
4	Sum of all singular values of Original Image (Green Matrix).	
5	Mean of all singular values of Original Image (Green Matrix).	
6	Sum of all right singular vectors of Original Image (Green Matrix).	
7	Mean of all right singular vectors of Original Image (Green Matrix).	

TABLE 2: Proposed Cellular Automata and Its Local Rule

Proposed Algorithm

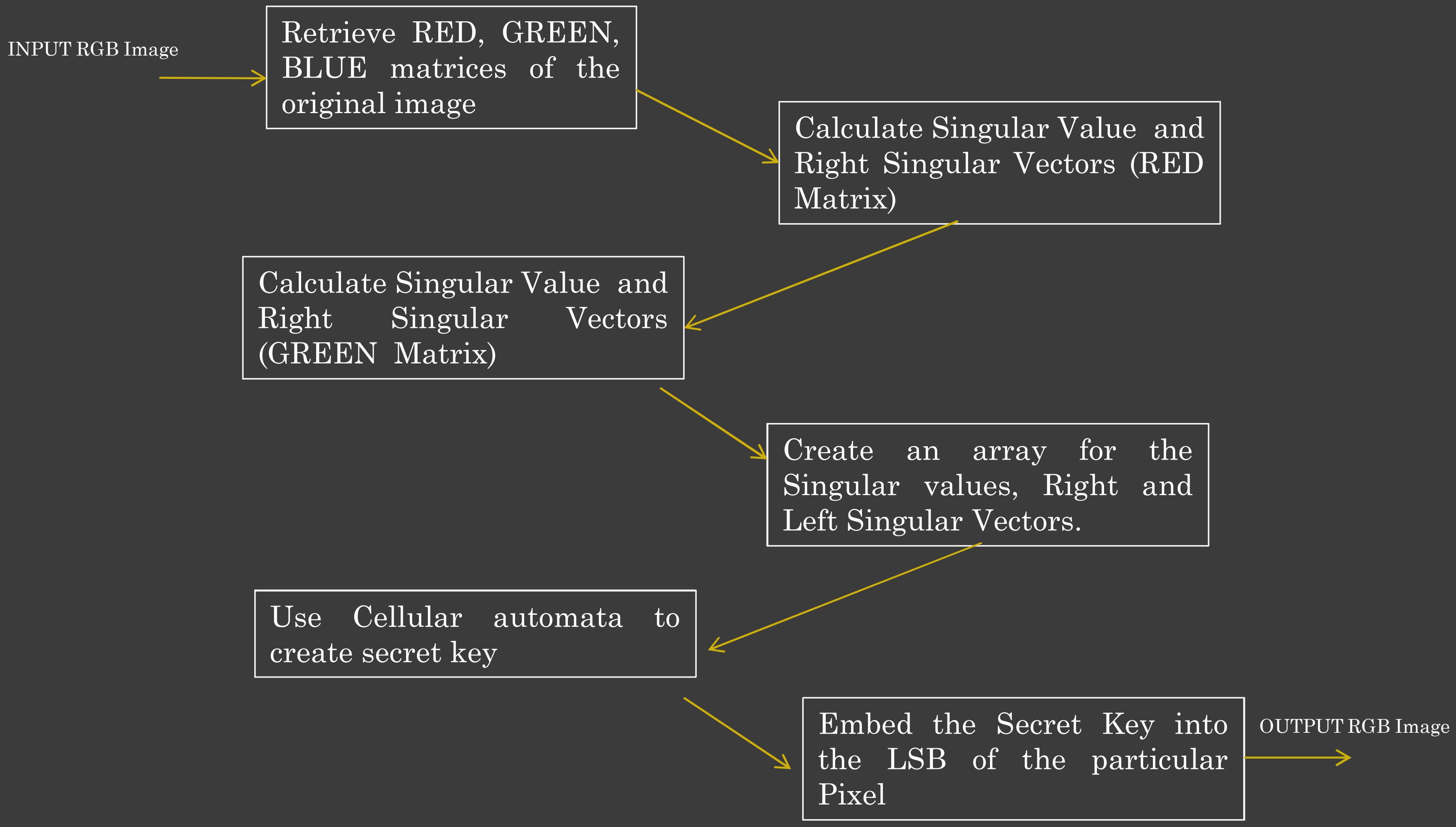


Figure 5: Block Diagram of Proposed Algorithm

Proposed Algorithm

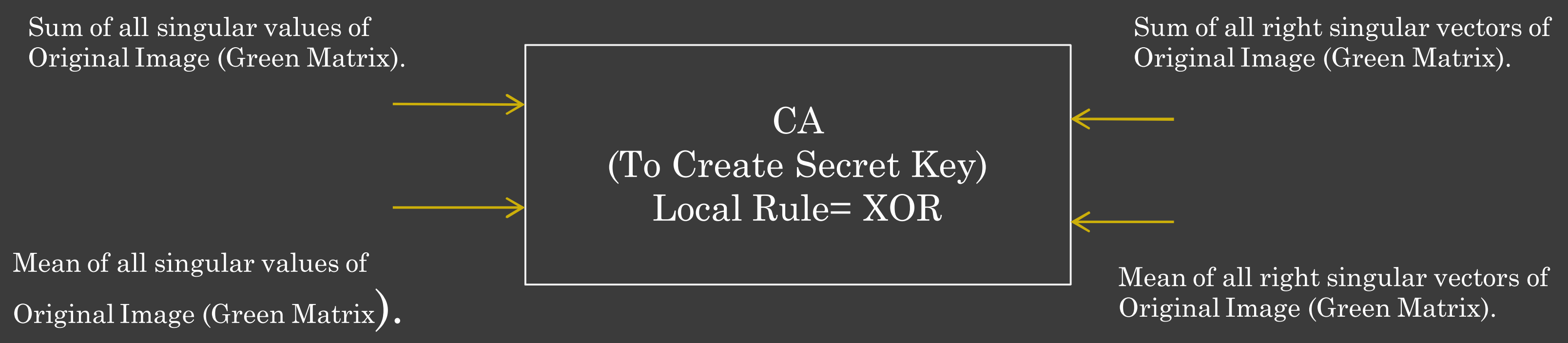


Figure 6: Block Diagram of Proposed Cellular Automata

Data Embedding Algorithm

Input: *RGB image to apply our proposed data embedding on it for the forgery protection.*
Output: *RGB image file.*

Step1: Open the original image and obtain the Red, Green and Blue matrices of the image.

Step2: Calculate the singular value *and* right singular vectors *of the Red Matrix.*

Step3: Calculate the singular value *and* right singular vectors *of the Green Matrix.*

Step4: Perform the cellular automata rule. This rule performs on the array list to create a Secret key.

Step5: Convert the Secret key to the binary representation.

Step6: Select the first eight pixels in the Blue Layer (Blue Matrix) and embed the binary sequences of Secret key into the LSB of each pixel for the authentication.

Forgery Detection Algorithm

Input: *image that contains a Secret key*
Output: *Digital image Forgery Detection Alarm*

Step1: Open the input image and make digital image matrix.
Step2: Initialize the integer variable SecretValue to zero.
Step3: Initialize the integer variable EigenVsArrayValue to zero.
Step4: Calculate the singular value *and the* right singular vectors *of the Red Matrix.*
Step5: Calculate the singular value *and the* right singular vectors *of the Green Matrix.*
Step6: Perform the cellular automata rule according to the Table II. This rule performs on the array list to create a Secret key.
Step7: Select the first eight pixels of Blue layer and extract the LSB binary value of each pixel.
Step8: Set SecretValue = Value of the Secret key that generated in Step 4 and 5 and 6.
Step9: Set EigenVsArrayValue = the extraction value in Step 7.
Step10: If EigenVsArrayValue = = SecretValue then print message "False Forgery Alarm"
Else Print message "True Forgery Alarm";

Experimental Results

To prove the performance of our proposed forgery detection method, a system has been built to implement the proposed algorithms. Six experiments will be presented in this section to show the implementation and the results of our proposed method. These are as follow:

- Visual Quality
- True and False Alert
- Time Complexity
- Diffusion
- Confusion
- PSNR

Visual Quality

Figure 7: Original Image

Figure 8: Data Embedded Image

True and False Alert Detection

True Alert (%)	False Alert (%)
94	6

TABLE 3: True and False Alert Detection.

Dataset: 100 Digital RGB Image

Time Complexity

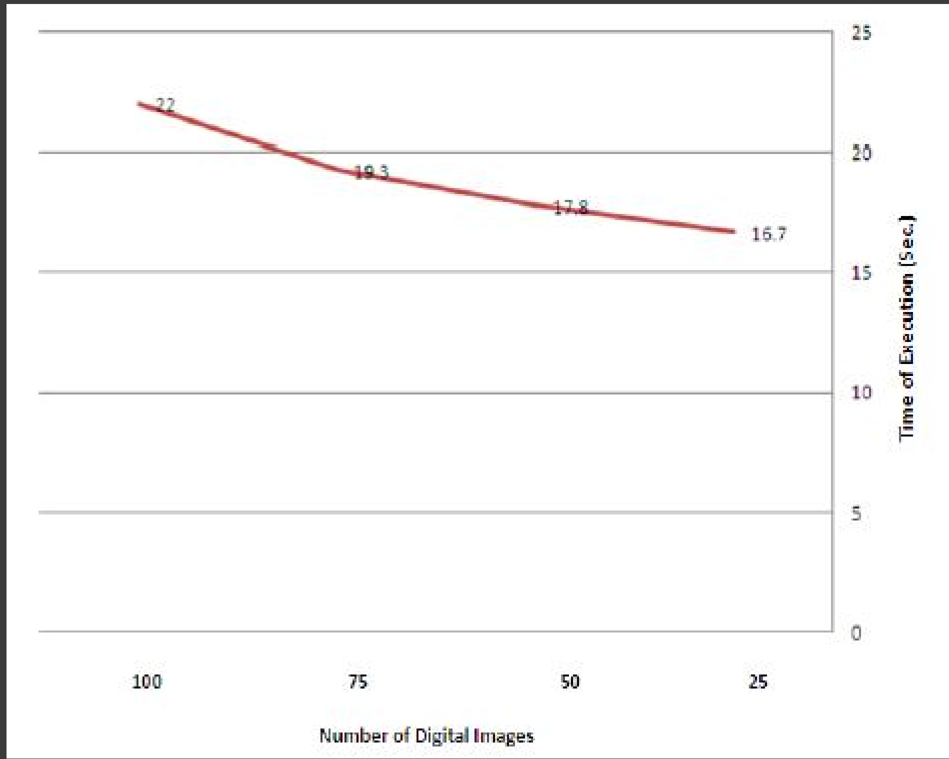


Figure 9: Time Complexity; Line Chart

We tested our code on a computer with CPU Pentium IV, 3.20 GHZ

Diffusion

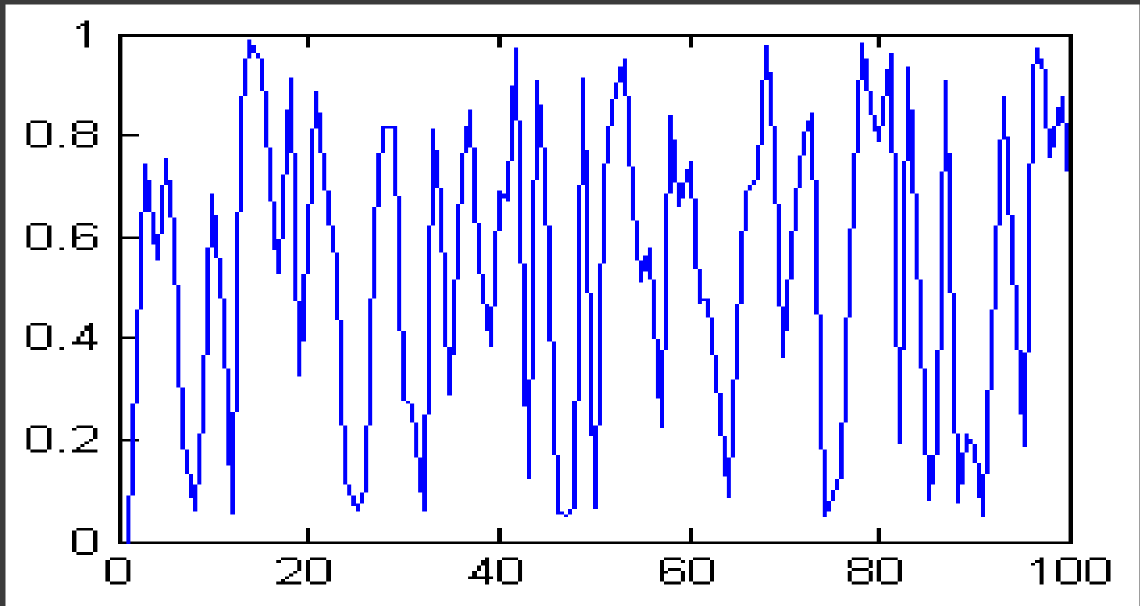


Figure 10: Diffusion Char for Proposed Secret Key. Row indicates the number of images and column indicates the random number which is between 0 and 1

Confusion

Sum of Eigen Values (Red Layer)	Mean of Eigen Values (Red Layer)	
47	11.75	Original Image
40	10	10 Pixel Altered
39	9.75	30 Pixel Altered

Table 4: CONFUSION OF OUR PROPOSED METHOD THE YELLOW COLOR INDICATES THE CHANGING MADE IN THE CORRESPONDING VALUES

PSNR

Measuring the PSNR (Peak Signal-to-Noise Ratio) of our proposed method is the next experiment. It indicates the maximum possible power of a signal and the power of corrupting noise that affects the output. We mentioned that all pixels in both of the input and output images in our proposed method are based on 8 bits. We implement PSNR function in MATLAB R2009a. When we run this function in MATLAB R2009a, it indicates that the PSNR for our proposed algorithm was 34.73dB. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better (Welstead, 1999) .

Therefore, our proposed algorithm has a well formed PSNR.

Conclusion

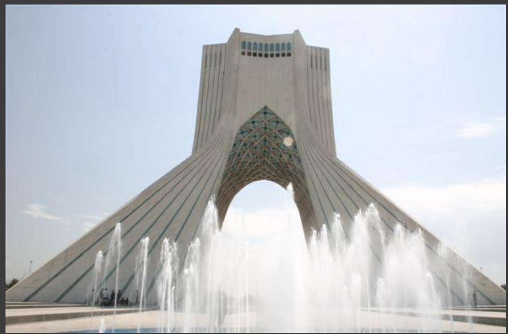
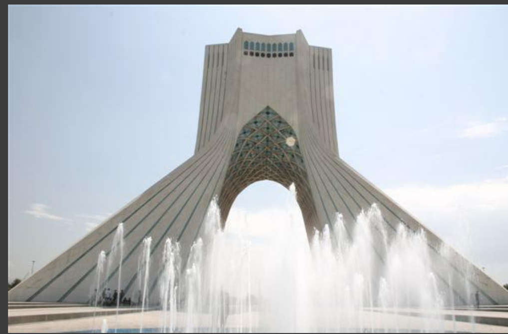
The proposed method has been applied completely and successfully for digital image forgery detection.

In this work we have presented a new method base on singular value decomposition and cellular automata which was done by calculating the invaluable dominant attributes of a RGB digital image.

The cellular automata rule generates a secret key that can be used to embed it into the original image for the forgery detection. This algorithm needs the original image for the forgery detection process.

The experimental results obtained from this method, specially the confusion, diffusion and true and false alert of our proposed model, clearly shown the robustness and reliability of our method. There are some aspects that can be improved in these methods. For instance, we can use the neural networks to improve the quality of the active digital image forgery methods.

Examples

Original Image	Sum of Singular Value (Original Image)	Data Embedded Image	Sum of Singular Value (Data Embedded Image)
	17		11
	53		59
	31		24

References

[1] H. Farid, "*Image Forgery Detection a Survey", IEEE Signal Processing Magazine, March 2009.*

[2] A. Pahlavan Tafti, M.V. Malakooti, M.Ashourian, S.Janosepah, "*Digital Image Forgery Detection Through Data Embedding in Spatial Domain and Cellular Automata", IEEE 7th International Conference on Digital Contents, Multimedia Technology and Its Application, Busan, Korea, Aug 2011.*

[3] I. J. Cox, M. L. Miller, and J. A. Bloom, "*Digital Watermarking", San Francisco, CA: Morgan Kaufmann, 2002.*

[4] A. Bovik, "*The Essential Guide to Image Processing", Academic Press is an imprint of Elsevier, 2009.*

[5] W. Puech, J. Rodrigues, "*Crypto-compression of Medical Image by Selective Encryption of DCT", 13th European Signal Processing Conference, Turkey, September 2005.*

[6] K.Z. Mao, "*Identifying Critical Variables of Principal Components for Unsupervised Feature Selection", IEEE Trans. Syst., Man, Cybern. B, vol. 35(2), pp. 339-344, 2005.*

[7] Alter O, Brown PO, Botstein D, "*Singular value decomposition for genome-wide expression data processing and modeling", Proc Natl Acad Sci U S A, 97, 10101-, 2000.*

[8] Golub, G.H., and Van Loan, C.F. "*Matrix Computations", 2nd ed. Baltimore: Johns Hopkins University Press, 1989.*

[9] M. Embree, "Numerical Analysis Lecture Notes", Rice University, October 2009.

[10] Shatten A., "*Cellular Automata", Institute of General Chemistry Vienna University of Technology, Austria, 1997.*

[11] O. Lafe, "*Data Compression and Encryption Using Cellular Automata Transforms", Artif. Intell, vol. 10, No. 6, pp. 581-591, 1997.*

[12] J. Urías, " *Cryptography Primitive Based On A Cellular Automata", An Interdisciplinary Journal Of Nonlinear Science, 8(4):819-822, 1998.*

[13] Welstead, Stephen T. "*Fractal and Wavelet Image Compression Techniques", SPIE Publication. pp. 155-156. ISBN 978-0819435033. 1999.*

[14] Wade Trappe, Lawrence C. Washington, "*Introduction to Cryptography with Coding Theory". Second edition. Pearson Prentice Hall, 2006.*

Thank you very much

Questions?