A novel application for transmission of Orthogonal Embedded Images by using Morphological Transform Domain & cryptographic methods

M.Venkata Kishore M.Tech (SE).,

CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, A.P., India.

JayaVani.V

Asst.Professor, CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, A.P., India.

SATYA P KUMAR SOMAYAJULA (Member *109101*, IAENG)

Asst.Professor, CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, A.P., India.

ABSTRACT

This paper proposes a data-hiding technique for water marked images and secured transmission of those watermarked images to the destination . today's lot of companies are stored their data in the images, they store data with their ownership. To achieve blind watermark extraction, it is difficult to use the detail coefficients directly as a location map to determine the data-hiding locations. Hence, we view flipping an edge pixel in binary images as shifting the edge location one pixel horizontally and vertically. Based on this we propose an morphological binary wavelet transform to track the shifted edges, which thus facilitates blind watermark extraction with support of cryptographic concepts.

In existing block-based approach, in which the block size is taken as 3*3 pixels or larger,and then we insert the text into that particular blocks and they are using morphological binary wavelet transform.now we propose an alternative approach to the above approach i.e we process an image in 2*2 pixel blocks. This gives high security when compared to existing approach.This allows flexibility in tracking the edges and also achieves low computational complexity. The two processing cases that flipping the candidates of one does not affect the flippability conditions of another are employed for orthogonal embedding, which means when ever we are embedded some text into the image the nieghbour pixels are not effected.A novel effective Backward-Forward Minimization method is proposed, which considers both back-wardly those neighboring processed embeddable candidates and forwardly those unprocessed flippable candidates that may be affected by flipping the current pixel. In this way, the total visual distortion can be minimized.

INTRODUCTION

WATERMARKING and data-hiding techniques have found wide applications in ownership identification, copy protection, fingerprinting, content authentication and annotation . The design requirements for a data-hiding or watermarking system are different catering for different applications. Recently authentication of digital documents has aroused great interest due to the wide applications in handwritten signatures, digital books, business documents, personal documents, maps, engineering drawings, and so on. On the other hand, editing an image becomes easier with the powerful image editing tools and digital cameras. Authentication to detect tampering and forgery is thus of primary concern. To ensure the authenticity and integrity of these digital documents has increased the confidence level from the user point of view.

The goal of authentication is to ensure that a given set of data comes from a legitimate sender and the content integrity is preserved. Hard authentication rejects any modification made to a multimedia signal, whereas soft authentication differentiates legitimate processing from malicious tampering. This system focuses on hard authenticator watermark-based authentication. Specifically, we investigate the problem of data hiding for binary images in morphological transform domain. Generally speaking, data hiding in real-valued transform domain does not work well for binary images due to the quantization errors introduced in the pre/post-processing. In addition, embedding data using real-valued coefficients requires more memory space. We observe that the morphological binary wavelet transform can be used to track the transitions in

Binary images by utilizing the detail coefficients. One rather intuitive idea in employing the morphological binary wavelet transform for data hiding is to use the detail coefficients as a location map to determine the data-hiding locations. However, this makes it difficult to achieve blind watermark extraction due to the fact that once a pixel is flipped, the horizontal, vertical and diagonal detail coefficients will change correspondingly.

The idea of designing an interlaced transform to identify the embeddable locations is motivated by the fact that some transition information is lost during the computation of a single transform and there is a need to keep track of transitions between two and three pixels for binary images data hiding. Specifically, we process the images based on 2*2 pixel blocks and combine two different processing cases that the flippability conditions of one are not affected by flipping the candidates of another for data embedding, namely “orthogonal embedding. Implementing the transforms by the “Exclusive OR (XOR)” operation addresses high security for trasimission.

The sequence of operations to be performed for image protection and secured image transimission are

1) Image as input
2) Watermark embedding
3) Authenticator Watermark
4) Swap Embedding
5) Watermarked Image

1)Image as input :

We give image as input ,process an image in 2x2 pixel blocks. This allows flexibility in tracking the edges and also achieves high computational complexity. The two processing cases that flipping the candidates of one does not affect the flippability conditions of another are employed for orthogonal embedding .

2)**Watermark embedding :**

watermarking is a technology for embedding various types of information in digital content. In general, information for protecting copyrights and proving the validity of data is embedded as a watermark. Watermarked content can prove its origin, thereby protecting the data.

3)Authenticator Watermark :

In this module we encrypt the data embedded image.The purpose of authenticator watermark of a block is invariant in the watermark embedding process, hence the watermark can be extracted without referring to the original image .The encryption and decryption technices used in this module.

4)Swap Embedding :

We flipp an edge pixel in binary images is equivalent to shifting the edge location horizontally one pixel and vertically one pixel. A horizontal edge exists if there is a transition between two neighboring pixels vertically and a vertical edge exists if there is a transition between two neighboring pixels horizontally. We swap an morphological images.

5)Watermarked image

The watermarked image is obtained by computing the inverse for the main processing block to reconstruct its candidate pixels.use this module we going to see the original watermarked image.

METHODOLOGY

Algorithm:

High-level description of the algorithm

- Key Expansion—round keys are derived from the cipher key using Rijndael's key schedule
- Initial Round

1. AddRoundKey—each byte of the state is combined with the round key using bitwise xor

- Rounds

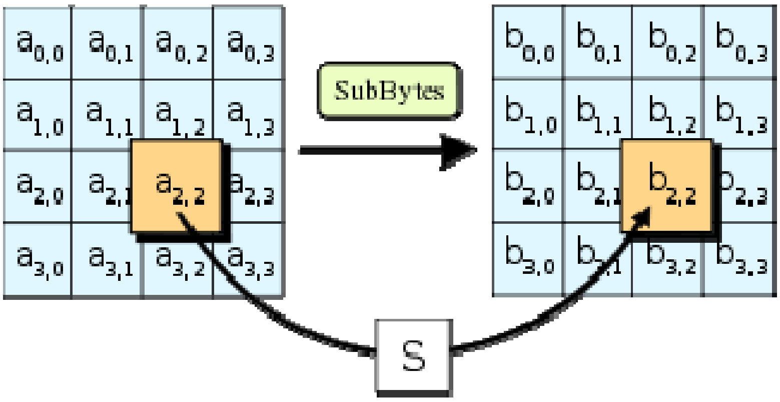
1. Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
2. Shift Rows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
3. Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column
4. AddRoundKey

- Final Round (no Mix Columns)

1. Sub Bytes
2. Shift Rows
3. AddRoundKey

The SubBytes step

In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S; b_{ij} = S(a_{ij}).In the SubBytes step, each byte in the array is updated using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over **GF**(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

The ShiftRows step

In the shift rows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.

The Shift Rows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the Shift Rows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively - this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

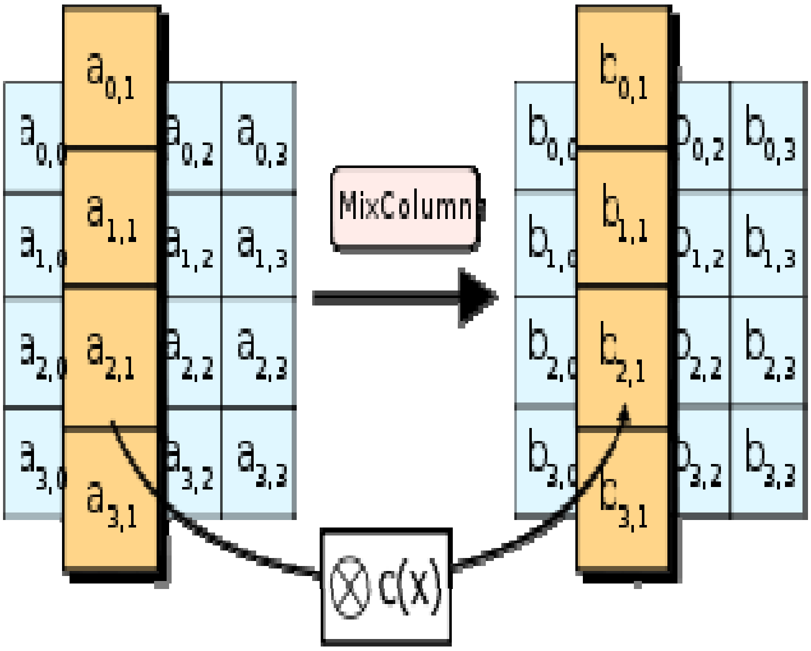
In the Mix Columns step, the four bytes of each column of the state are combined using an invertible linear transformation. The Mix Columns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with Shift Rows, Mix Columns provides diffusion in the cipher.

During this operation, each column is multiplied by the known matrix that for the 128 bit key is

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

The multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing xor with the initial unshifted value.

In more general sense, each column is treated as a polynomial over **GF**(2^8) and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from **GF**(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.



The AddRoundKey step

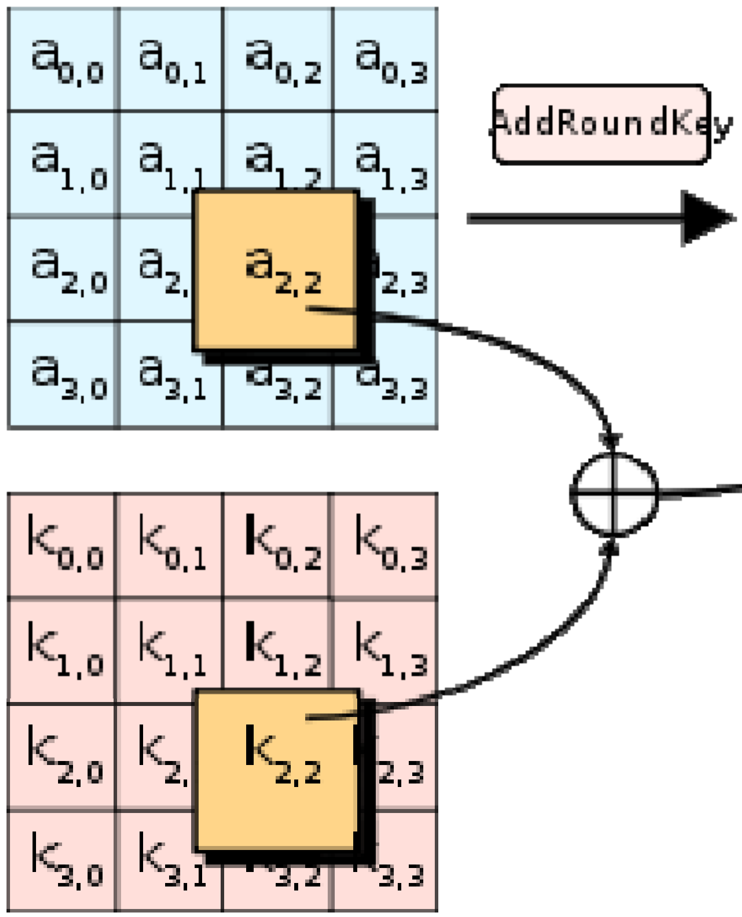
In the AddRoundKey step, the sub key is combined with the state. For each round, a sub key is derived from the main key using Rijndael's key schedule; each sub key is the same size as the state. The sub key is added by combining each byte of the state with the corresponding byte of the sub key using bitwise XOR.

Optimization of the cipher

On systems with 32-bit or larger words, it is possible to speed up execution of this cipher by combining Sub Bytes and Shift Rows with Mix Columns, and transforming them into a sequence of table lookups. This requires four 256-entry 32-bit tables, which utilizes a total of four kilobytes (4096 bytes) of memory—one kilobyte for each table. A round can now be done with 16 table lookups and 12 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the AddRoundKey step

If the resulting four kilobyte table size is too large for a given target platform, the table lookup operation can be performed with a single 256-entry 32-bit (i.e. 1 kilobyte) table by the use of circular rotates.

Using a byte-oriented approach, it is possible to combine the Sub Bytes, Shift Rows, and Mix Columns steps into a single round operation.



Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the Shift Rows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively - this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

Morphological binary wavelet transform:

In this image divided into 3*3 blocks.after we divide the image into 2*2 blocks as shon in below:

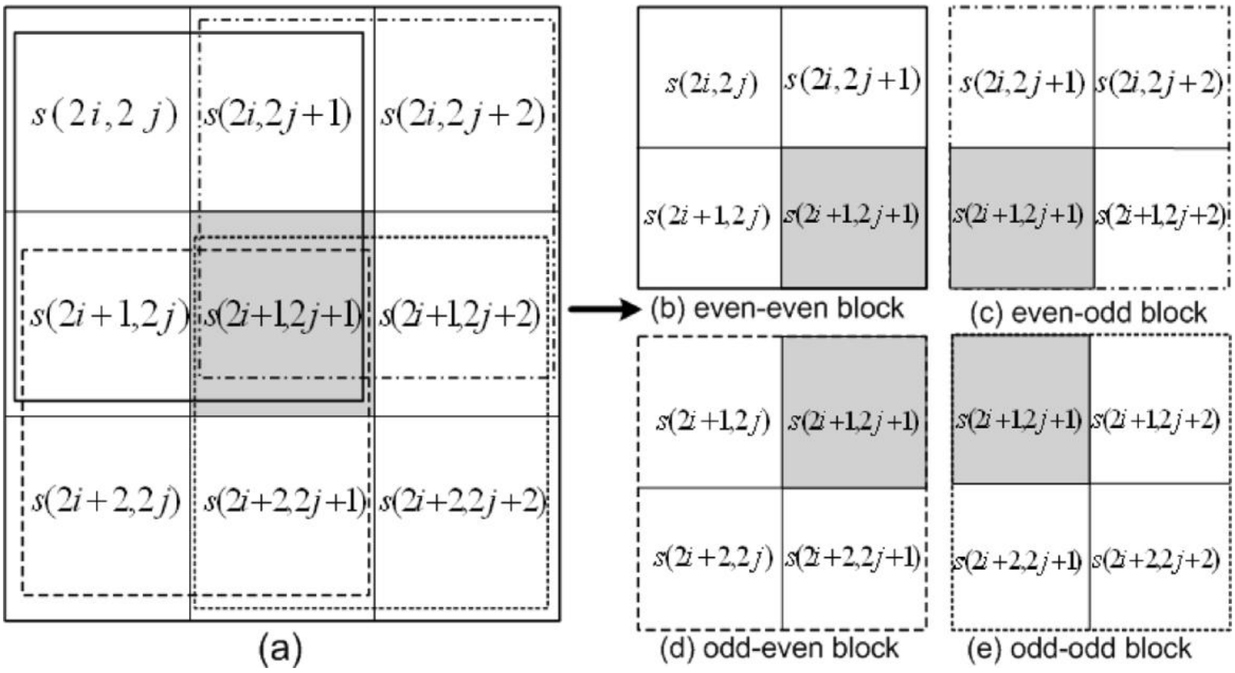
These blocks are given below:

1.even-even block

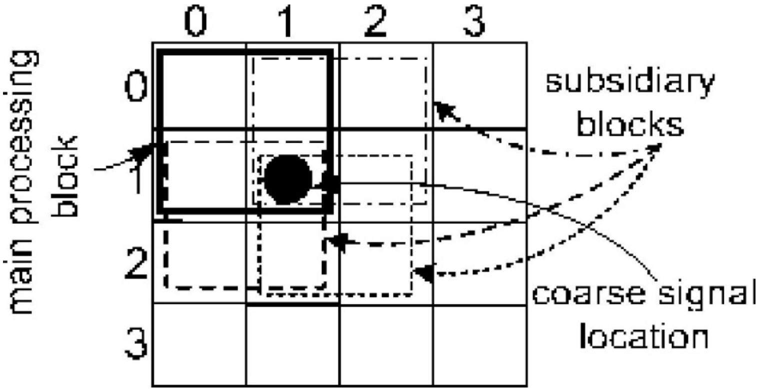
2.even-odd block

3.odd-even block

2.odd-odd block



Processing of each block as follows:



OPERATIONS

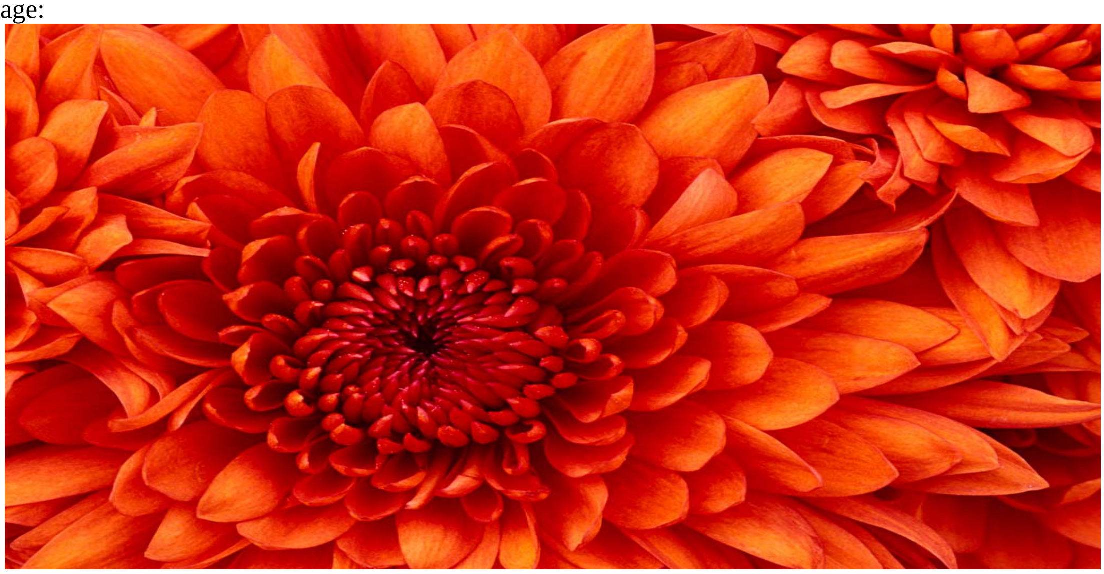
Sequence of operations to be performed for image security & secured image transimission

1) Image as input
2) Watermark embedding
3) Authenticator Watermark
4) Swap Embedding
5) Watermarked Image

1)Image as input :

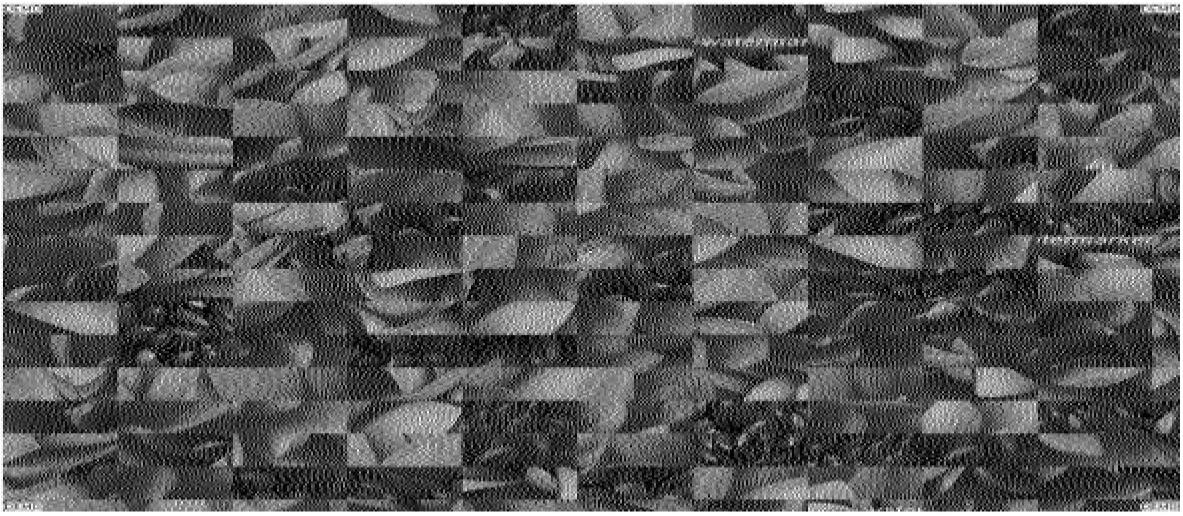
We give image as input ,process an image in 2x2 pixel blocks. This allows flexibility in tracking the edges and also achieves high computational complexity. The two processing cases that flipping the candidates of one does not affect the flip ability conditions of another are employed for orthogonal embedding .first we take the inage
And process the image as follows:
1) take an color image or black and white image
2) for color images convert into binary image
3) Perform the flip ability operations on image
Example here we are taking an image as follows:
1) Image as input

Color image:



Binary image:

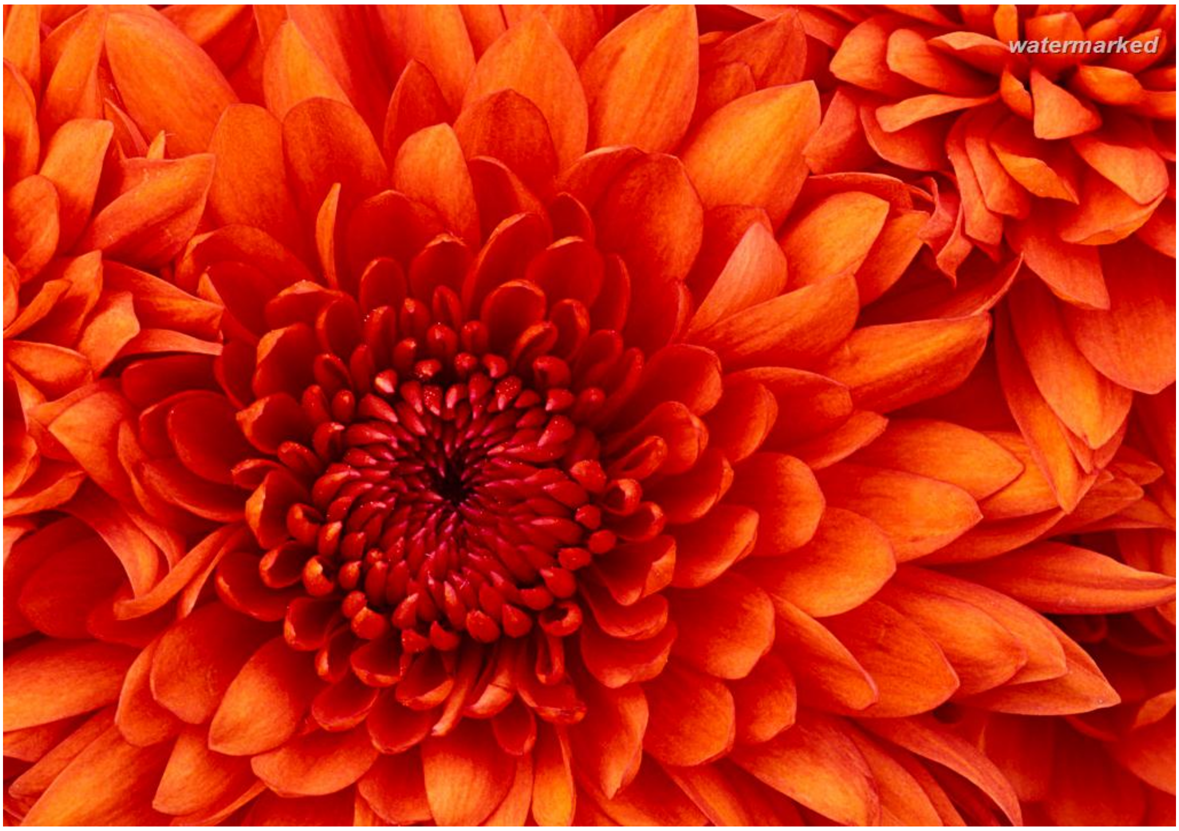
After flipping:



2) **Watermark embedding :**

watermarking is a technology for embedding various types of information in digital content. In general, information for protecting copyrights and proving the validity of data is embedded as a watermark. Watermarked content can prove its origin, thereby protecting the data.we are inserting the text into the image at anywhere we want.that insertion must be done without effecting the neighbor pixcels of the image.this is called orthogonal embedding

for example suppose we are inserting text “watermark” at rightside upper part of the image then we got image as follows:



3)Authenticator Watermark :

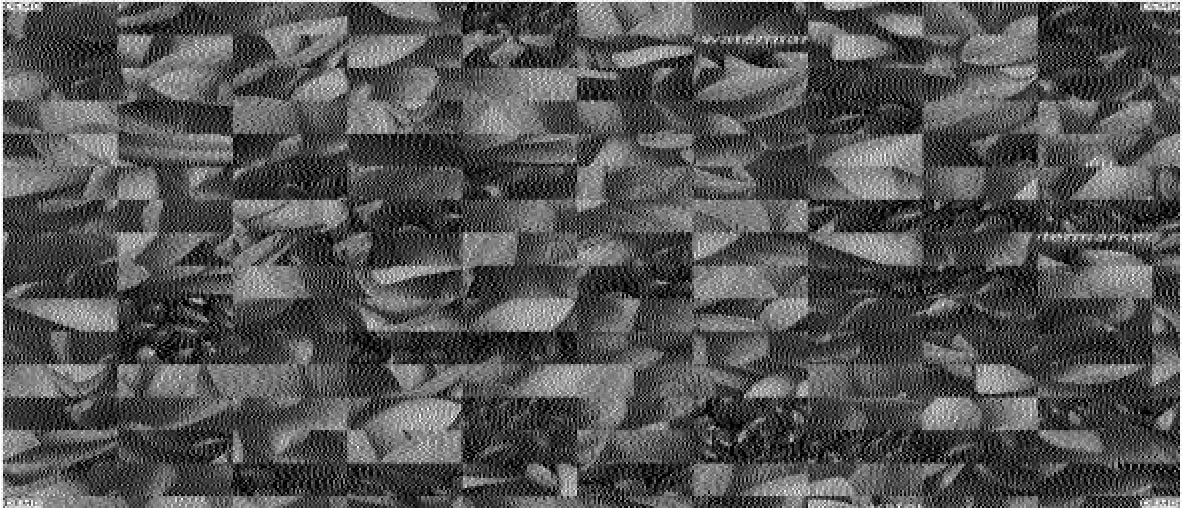
we encrypt the data embedded image.The purpose of authenticator watermark of a block is invariant in the watermark embedding process, hence the watermark can be extracted without referring to the original image .The encryption and decryption technices used in this .by using cryptographic concepts we encrypt the image and store where we want.for encryption we suggests advanced encryption algorithm(AES) or Rijndael's key schedule algorithm.

4)Swap Embedding :

We flipp an edge pixel in binary images is equivalent to shifting the edge location horizontally one pixel and vertically one pixel. A horizontal edge exists if there is a transition between two neighboring pixels vertically and a vertical edge exists if there is a transition between two neighboring pixels horizontally. We swap an morphological images.the swapping is performed among blocks in image is totally depends upon user choice.

After swapping we got an distorted image.this is image is never readable or understable by any third party or intruder.

The image after swapping is look like below:



5) Watermarked image

The watermarked image is obtained by computing the inverse for the main processing block to reconstruct its candidate pixels.use this module we going to see the original watermarked image.suppose user or receiver wants to get the original image then he/she must perform the inverse process of the above main process.then only he get the original image:

\

APPLICATIONS:

- Broadcast monitoring
- Proof of ownership
- Transaction Tracking
- Content authentication
- Modification and multiple watermark

CONCLUSION

- In this project , we present a high-capacity data-hiding scheme for binary images authentication based on the interlaced morphological binary wavelet transforms. The relationship between the coefficients obtained from different transforms is utilized to identify the suitable locations for watermark embedding such that blind watermark extraction can be achieved.

- Two processing cases that are not intersected with each other are employed for orthogonal embedding in such a way that not only can the capacity be significantly increased, but the visual distortion can also be minimized. Results of comparative experiments with other methods reinforce the present scheme's superiority in being able to attain larger capacity while maintaining acceptable visual distortion and low computational cost.

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, Digital Watermarking. San Mateo, CA: Morgan Kaufmann, 2001.

[2] B. Furht and D. Kirovski, Multimedia Security Handbook, B. Furht and D. Kirovski, Eds. Boca Raton, FL: CRC, 2005.

[3] Y. Liu, J. Mant, E. Wong, and S. H. Low, "Marking and detection of text documents using transform-domain techniques," in Proc. SPIE,San Jose, CA, 1999, vol. 3657, pp. 317–328.

[4] Q. Mei, E. K. Wong, and N. Memon, "Data hiding in binary text document,"in Proc. SPIE, 2001, vol. 4314, pp. 369–375.

[5] Y. C. Tseng and H.-K. Pan, "Data hiding in 2-color images," IEEE Trans. Comput., vol. 51, no. 7, pp. 873–878, Jul. 2002.

[6] K.-F. Hwang and C.-C. Chang, “A run-length mechanism for hiding data into binary images,” in Proc. Pacific Rim Workshop on Digital Steganography, Kitakyushu, Japan, Jul. 2002, pp. 71–74.

[7] H. Lu, X. Shi, Y. Q. Shi, A. C. Kot, and L. Chen, “Watermark embedding in DC components of DCT for binary images,” in Proc.,IEEE Workshop on Multimedia Signal Processing, Dec. 9–11, 2002,pp. 300–303.

[8] M. Wu and B. Liu, “Data hiding in binary images for authentication and annotation,” IEEE Trans. Multimedia, vol. 6, no. 4, pp. 528–538,Aug. 2004.

[9] H. Y. Kim and R. L. de Queiroz, “Alteration-locating authentication watermarking for binary images,” in Proc. Int. Workshop Digital Watermarking,2004, pp. 125–136.

[10] H. Lu, A. C. Kot, and Y. Q. Shi, “Distance-reciprocal distortion measure for binary document images,” IEEE Signal Process. Lett., vol. 11,no. 2, pp. 228–231, Feb. 2004.

[11] Orthogonal Data Embedding for Binary Images in Morphological Transform Domain-A High-Capacity Approach Huijuan Yang, Alex C. Kot, Fellow, IEEE, and Susanto Rahardja, Senior Member, IEEE

Authors Biography

M.Venkatakishore, Studying M.Tech in Software Engineering. in CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam,A.P., India.

JayaVani.V is working as Asst.Professor, in CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam,A.P., India. She has received his B.Tech from Gayatri vidya parisehad, Visakhapatnam and M.Tech (CST) from Andhra University, Visakhapatnam.

Somayajula Satya Pavan Kumar is working as Asst.Professor, in CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam,A.P., India. He has received his M.Sc(Physics) from Andhra University,Visakhapatnam and M.Tech (CST) from Gandhi Institute of Technology And Management University (GITAM), Visakhapatnam.,A.P., INDIA. His research areas include Software Engineering and network security