A Fragile Watermarking Scheme Based CRC Checksum and Public Key Cryptosystem for RGB Color Image Authentication

Nour El-Houda Golea

Department of Computer Science, University of BATNA, Algeria
golea.nour@gmail.com

Abstract. The increased use of multimedia applications pose more problems concerning the preservation of confidentiality and authenticity of the transmission of digital data. These data, in particular the images should be protected from tampering. The solution is the use of fragile watermarking. Fragile watermarking can be modeled as a problem of communication of a signal over a noisy and hostile channel, where the attack takes place. Indeed, the use of error checking algorithms appear natural. . Cyclic redundancy check (CRC) code provides a simple, yet powerful, method for the detection of burst errors during digital data transmission and storage. CRC is one of the most versatile error checking algorithm used in various digital communication systems. In this paper, we propose a novel fragile watermarking scheme based CRC checksum and public key cryptosystem for RGB color image authentication.

Keywords: Fragile watermarking, image authentication, Cyclic redundancy check (CRC), public key cryptosystem, RGB color image watermarking.

1 Introduction

Digital watermarking technology is the process of embedding information into digital data in such a way that it is imperceptible to a human observer but easily detected by computer algorithm. A digital watermark is a invisible information pattern that is embedded into a suitable component of the data source by using a specific computer algorithm. Digital watermarks are signals added to digital data (audio, video, or still images) that can be detected or extracted later to make an assertion about the data [1, 2, 3].

Digital watermarking schemes can be classified as either *robust* or *fragile* . Robust watermarking schemes can be used to authenticate ownership [4, 5], whereas fragile watermarking schemes are commonly used for image authentication to verify whether the received image was modified during transmission or not. One may hide the watermark imperceptibly in the image before transmission and detect it after receiving to make sure that the received image is original or corrupted. Many watermarking schemes for image authentication have been proposed [6, 7, 8].

The watermarks used for the authentication must contain information which determines the integrity of the image. The watermark must invisible and fragile

A. Elmoataz et al. (Eds.): ICISP 2012, LNCS 7340, pp. 316–325, 2012.

(so, any modification in the watermarked image also in the signature must be detected and it is very desirable that it can detect the corrupted region.

In general, fragile watermarking schemes divide an original image into non-overlapping blocks, embed a signature, and detect the modified location for every block. Memon and Wong [6] proposed a method in which an image is divided into blocks and each block contains the hash value calculated from the MSB[1]'s of the pixels forming that block. Fridrich [7] also proposed that the authentication watermark should exclusive-OR (XOR) the hash value of the block with more block information. Lin and al. [8] proposed a fragile block-wise, and content-based watermarking for image authentication and recovery. In this scheme, the watermark of each block is an encrypted form of its signature, which includes the block location, a content-feature of another block, and a CRC checksum. While the CRC checksum is to authenticating the signature. With block-based fragile watermarking we cannot detect exactly the modified pixels, but we can detect the corrupted block.

Fragile watermarking schemes are classified into those using the public key cryptosystem [7,8] and those using the private key cryptosystem as the tool for making the signature [6].

In this paper, we propose fragile watermarking based pixel detection approach using CRC and a public key cryptosystem. This approach is proposed to authenticate the RGB color image using the CRC checksum to detect the corrupted pixels.

The proposed method is decomposed from four functions: the first one generate a generator polynomial P_X with same size as the host image $n \times m$, each element $P_X(i,j)$ is used to create the watermark $W(i,j)$. The second function, use P_X to generate the watermark W of size $n \times m$, each element $W(i,j)$ is a binary sequence of six bits. The watermark $W(i,j)$ is the remainder of the division of the 18 MSB bits of the three colored pixels $R(i,j)$, $G(i,j)$ and $B(i,j)$ by $P_X\ (i,j)$. After this step, the P_X is encrypted using a secret key K_S , and performing the RSA encryption algorithm. The fired function embed each two bits of $W(i,j)$ in the two LSB[2] of the corresponding three colored pixels. At the reception, the encrypted P_X is decrypted using a public key K_P. The last function, extract the watermark (CRC checksum) from the two LSB of the three colored pixels. The extracted watermark is appended at the end of the 18 MSB of the colored pixels. Then, divide this new sequence by $P_X\ (i,j)$, if the remainder is zero then the pixel $f(i,j)$ is authentic else it is corrupted.

The remainder of this paper is organized as follows: Section 2 gives a brief description of the CRC principle. Our proposed scheme is presented in Section 3. In Section 4, the experimental results are described and analyzed. Finally, we draw the conclusions of our work in Section 5.

2 Cyclic Redundancy Check

Normally, for the error detection in digital communication systems, a certain number of check bits, often called a *checksum,* is computed on the message that needs to be transmitted. The computed checksum is then appended at the end of the message

[1] Most Significant Bit.

[2] Least Significant Bit.

stream and is transmitted. At the receiving end, the message stream's checksum is computed and compared with the transmitted checksum. If both are equal, then the message received is treated as error free. Cyclic Redundancy Code Check, or CRC works in a similar way, but it has greater capabilities for error detection than the conventional forms. CRC is one of the most versatile error checking technique used in various digital communication systems. Different CRC polynomials are employed for error detection. The size of CRC depends upon the polynomial chosen.

The message to be transmitted is treated as a polynomial and divided by an irreducible polynomial known as the *generator polynomial*. The degree of the generator polynomial should be less than that of the message polynomial. For a $n + 1$ bits generator polynomial, the remainder will not be greater than n bits. The CRC checksum of the data is the binary equivalent of the remainder after the division.

In general, an n-bit CRC is calculated by representing the data stream as a polynomial $M(x)$, multiplying $M(x)$ by x^n (where n is the degree of the polynomial P_X), and dividing the result by P_X. The rest of the division is the CRC checksum which is appended to the polynomial $M(x)$ and transmitted. The complete transmitted polynomial is then divided by the same P_X at the receiver end. If the result of this division has no remainder, there are no transmission errors [9].

3 Proposed Method

The proposed method is decomposed from four algorithms: P_X (generator polynomial) generation algorithm, W (watermark) generation algorithm, embedding and extraction algorithms. This method is modeled in Fig. 1.

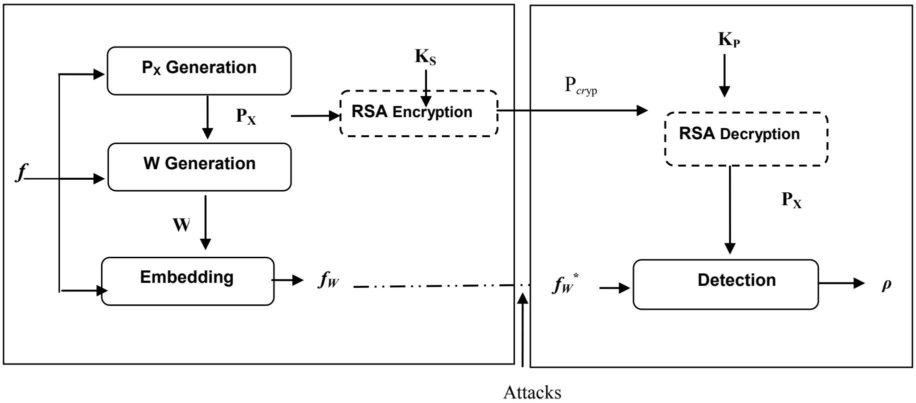


Fig. 1. Proposed model

The P_X generation algorithm create the matrix P_X of degree d taking the size of the original host image f and d. This function is described as $Generator_P$:

$$P_X = Generator_P(f,d). \tag{1}$$

The watermark generator algorithm generates a signature that contain the watermarking information, by taking the original host image *f* and a generator polynomial P_X, that is described as a function $Generator_W$:

$$W = Generator_W(f, P_X). \tag{2}$$

The embedding algorithm takes the signature and the host image, and generates the watermarked image f_w, that is described as a function *E*:

$$f_{w=} E(f, W). \tag{3}$$

The detection algorithm loads the watermarked, normal or corrupted image f_w^* and P_X, and calculate the measure ρ. The process can be described as function *D* :

$$\rho = D(f_w^*, \ P_X) \ . \tag{4}$$

if ρ=0 then the pixel is not corrupted, else it is corrupted.

The generator polynomial P_X must be encrypted using a secret key K_S and performing the asymmetric key *Encryption* algorithm. At the reception, the receptor must decrypt the generator polynomial P_{cryp} using the public key K_P and performing the *Decrypted* algorithm.

3.1 Generator Polynomial Generation Algorithm

This algorithm allows to create a matrix P_X of the same size of the host image *f* , each element $P_X(i, j)$ of this matrix is a generator polynomial used to calculate the CRC checksum corresponding to the pixel $f(i,j)$.

Input:

- $n \times m$: size of the host image *f*;
- *d*: the maximal degree of $G(x)$, i.e., the maximal number of bits used to insert the watermark (in this case $d=6$).

Output:

P_X: the generator polynomial matrix of size $n \times m$.

Steps:

```
for $i$=1 to $n$ do
    for $j$=1 to $m$ do
        - Randomly generate a binary sequence $g$ of size $d+1$;
        - $P_X(i,j)$ is calculated as:
```

$$P_X(i,j) = g_1 X^6 + g_2 X^5 + g_3 X^4 + g_4 X^3 + g_5 X^2 + g_6 X^1 + g_7 X^0. \tag{6}$$

To encrypt and decrypt the P_X, we propose to use the RSA algorithm (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman). The RSA cryptosystem is the most widely-used public key cryptography algorithm. It can be used to encrypt a message without the need to exchange a secret key separately. The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers [10, 11].

3.2 Watermark Generation Algorithm

This algorithm generate a watermark *W* of size *n* × *m*, where each element is presented by 6 bits. The watermark is generated depending on 18 MSB bits of the colored pixels (R, G and B). The following algorithm summarizes the way the watermark *W* is generated using the host image *f* and the generator polynomial P_X.

Input:

- *f*: RGB color host image of size *n* × *m;*
- P_X: generator polynomial matrix of size *n* × *m.*

Output:

- W : watermark of size *n* × *m,* each element *W(i,j)* is binary sequence of 6 bits W={ $W_1, W_2, \ldots, W_6$}.

Steps:

- For each colored pixels *R(i,j), G(i,j)* and *B(i,j)* do:

1. Construct the message *M* by concatenating the 6 MSB bits of each pixel.
2. Perform the CRC encoding to calculate the checksum:
   - Calculate $M' = M(x) \times x^d$;
   - The watermark *W(i,j)* is the remainder of division of *M'* by P_X *(i,j).*

Fig. 2 illustrates the block diagram of the watermark generation algorithm.

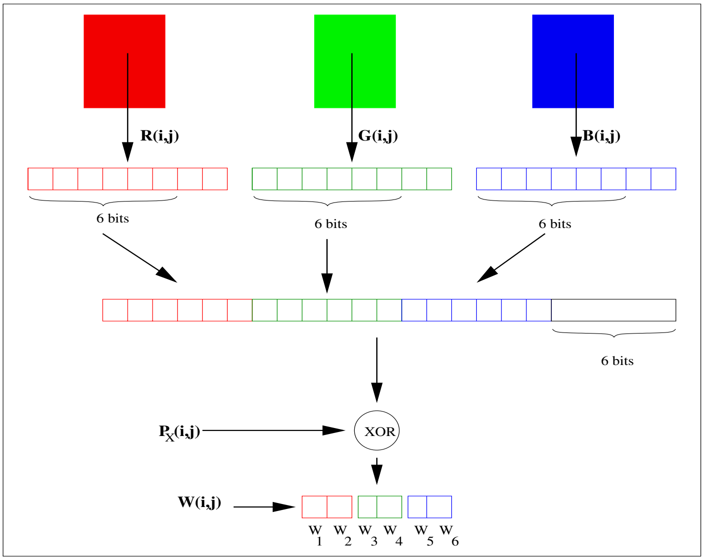


Fig. 2. Block diagram of the watermark generation algorithm

3.3 Embedding Algorithm

The following algorithm describes the way the watermark *W* is inserted in the host image *f*.

Input:

- *f*: RGB color host image of size $n \times m$;
- *W* : generated watermark of size $n \times m$.

Output:

- f_w: watermarked image of size $n \times m$.

Steps:

- For each colored pixels $R(i,j)$, $G(i,j)$ and $B(i,j)$ do :
  1. Replace the two LSB bits of $R(i,j)$ by $W_1(i,j)$ and $W_2(i,j)$;
  2. Replace the two LSB bits of $G(i,j)$ by $W_3(i,j)$ and $W_4(i,j)$;
  3. Replace the two LSB bits of $R(i,j)$ by $W_5(i,j)$ and $W_6(i,j)$.

Fig. 3 presents the block diagram of the watermark embedding algorithm.

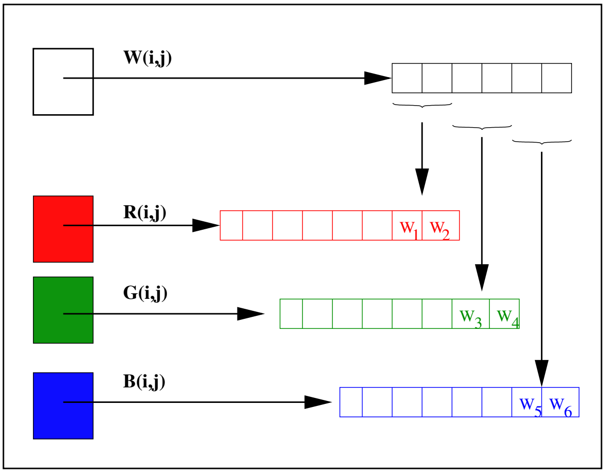


Fig. 3. Block diagram of the embedding algorithm

3.4 Detection Algorithm

The following algorithm summarizes the way the watermark is extracted from watermarked image f_w and $P(x)$.

Input:

- f_w: RGB color watermarked received image of size $n \times m$;
- P_X: generator polynomial matrix of size $n \times m$.

Output:

- ρ: decision parameter (confidentiality measure).

Steps:

- For each pixel $R_w(i,j)$, $G_w(i,j)$ and $B_w(i,j)$ do :

1. Construct the message *M'* by concatenating the 6 MSB bits of each pixel.
2. Extract the watermark *W* from the two LSB of each pixel.
3. Perform the CRC decoding:
   - The watermark *W* is appended at the end of *M'*;
   - Calculate the remainder ρ of the devising of *M'* by $P_X(i,j)$.

     if $\rho=0$ then the watermarked pixel $f_w(i,j)$ is not corrupted else $f_w(i, j)$ is corrupted.

Fig. 4 shows the block diagram of the detection algorithm.

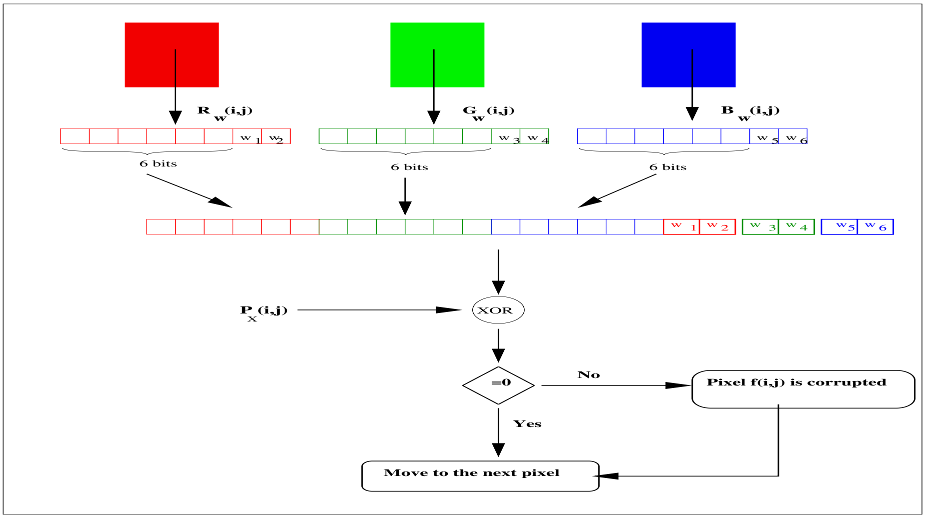


Fig. 4. Block diagram of the detection algorithm

4 Simulation and Experimental Results

In this section, we mainly demonstrate the imperceptibility and the fragility of our watermarking method. The experimental results reported here have been separated into two parts: the first one is for testing the imperceptibility property and the other one is for evaluating the fragility to malicious manipulations.

Imperceptibility Property

In order to test the imperceptibility property of our watermarking method, several typical RGB color images with size 128×128 such as *Baboon*, *Sailboat*, *Lena* and *house* have been watermarked. These original host images with their watermarked images have, respectively, been shown in Fig. 5.

From these result images, we could see that the differences between the original images and their corresponding watermarked images are hard to be perceived by human eyes.

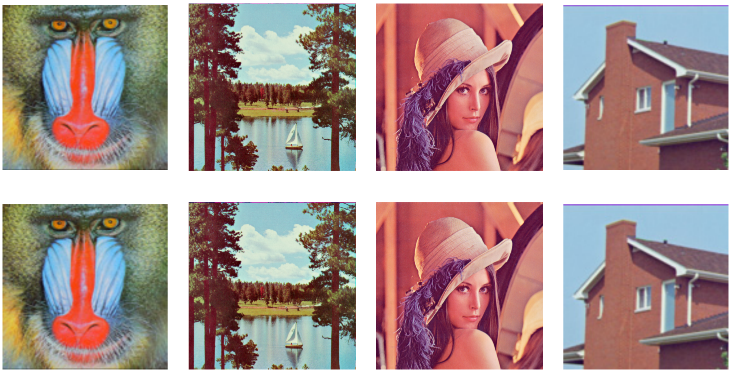


Fig. 5. Host images with their corresponding watermarked images

To concretely estimate the quality of our method, we employed the *Peak Signal to Noise Ratio (PSNR)* to evaluate the distortion of the watermarked images. Table 2 depicts the name of some of the experimental images and their PSNR values of RGB components.

Table 1. Quality of the watermarked images

Host image	Baboon	Sailboat	Lena	House
PSNR	47.2633	47.2407	47.2578	47.4048

It can be seen from the PSNR values that the distortion between the watermarked images and the original ones is imperceptible. One can also conclude that the watermarked images have a good quality when the block size increases.

Fragility Property

In order to evaluate the fragility to attacks, the CRC image is the matrix of the remainders of division of each pixel $f(i,j)$ by $P_x(i,j)$, if this image is black then the image is not corrupted, else it is corrupted. Fig. 6 presents the CRC images calculated from the watermarked images.

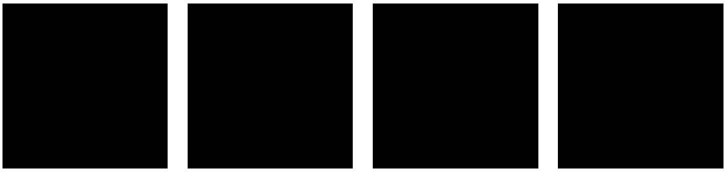
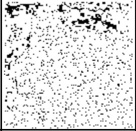
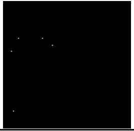
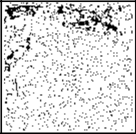
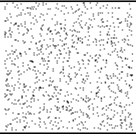
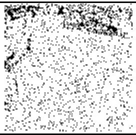
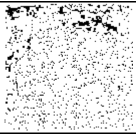
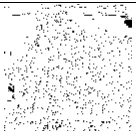
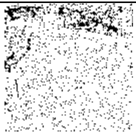
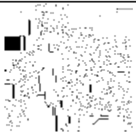
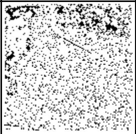
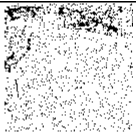


Fig. 6. CRC images extracted respectively from *Baboon*, *Sailboat*, *Lena* and *house* watermarked images

To highlight the fragility of our method, we have also taken into account many kinds of image watermarking attacks. Table 2 shows the extracted CRC images after different typical and standard attacks.

Table 2. Performances against several typical and standard attacks

Attack	CRC image	Attack	CRC image	Attack	CRC image
Rotation *Angle= 0.01°*		Average filter *3x3*		Salt & Pepper noise *D=0.002*	
Rotation *Angle=1°*		Gaussian filter *3x3*		Gaussian noise *M = 0.0, V = 0.001*	
Resize *128 ==>256*		Laplacian filter *Default parameters*		Winner filter *3x3*	
JPEG *Q=90*		Median filter *3x3*		Blurring *radius = 0.1*	
JPEG *Q=30*		Sharpen filter *1.0*		Blurring *radius = 1.0*	

Form Table 2, one can see that our watermark embedding method has strong fragile against many image attacks.

5 Conclusion

In this paper, we propose a fragile watermarking for RGB color images. The proposed method based pixel detection use CRC checksum and public key cryptosystem. The watermark is generated depending to the three colored pixels and using the generator polynomial. The generated watermark is inserted in the 2 LSB of each corresponding colored pixels. At the detection, the extracted watermark from the LSB of each colored pixels is appended at the end of the 18 MSB of the three colored pixels. This sequence is divided by the corresponding generator polynomial, if the remainder is null, then the pixel is not corrupted. The generator polynomial is encrypted using RSA public key cryptosystem.

The simulation results show that the proposed system performs fairly well when it is required to detect all kind of alteration, indicating precisely its altered region.

Acknowledgments. We are grateful to Dr Redha BENZID and Dr Rachid SEGHIR for early stimulating discussion and for their valuable comments and suggestions which lead to substantial improvements of this paper.

References

1. Voyatzis, G., Nikolaidis, N., Pitas, I.: Digital Image Watermarking: An Overview. In: 9th IEEE European Signal Processing Conference, vol. 1, pp. 9–12 (1998)
2. Barni, M., Cox, I., Kalker, T., Kim, H.J.: Digital Watermarking. In: 4th International Workshop, IWDW, Siena, Italy, September 15-17, Proceedings Series. Lecture Notes in Computer Science, vol. 3710 (2005)
3. Mitthelholzer, T.: An Information- Theoric Approach to Steganography and Wateramarking. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 1–17. Springer, Heidelberg (2000)
4. Golea, N.E.H., Seghir, R., Benzid, R.: A Bind RGB Color Image Watermarking Based on Singular Value Decomposition. In: IEEE/ACS International Conference on Computer Systems and Applications - AICCSA 2010, pp. 1–5 (2010)
5. Zhao, X., Ho, A.T.S.: An Introduction to Robust Transform Based Image Watermarking Techniques. In: Collection of Intelligent Multimedia Analysis for Security Applications, pp. 337–364 (2010)
6. Wong, P.W., Memon, N.: Secret and public key image-watermarking schemes for image authentication and ownership verification. IEEE Transactions on Image Processing 10(10) (2001)
7. Fridrich, J.: Security of Fragile Authentication Watermarks with Localization. In: Proc. SPIE, vol. 4675, pp. 691–700 (January 2002)
8. Lin, P., Huang, P., Peng, A.: A Fragile Watermarking Scheme for Image Authentication with Localization and Recovery. In: IEEE Sixth International Symposium on Multimedia Software Engineering (MSE 2004), Florida, USA (2004)
9. Tanebaum, A.S.: Computer Networks, 4th edn. Pearson Education International, The Netherlands (2003)
10. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystem. Communications of the ACM 21, 120–126 (1978)
11. Bellare, M., Rogaway, P.: Optimal Asymetric Encryption – How to Encrypt with RSA. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)