Volume 4, Issue 1, January 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Invisible and Blind Watermarking Method Using Hybrid Digital Embedding Technique

Y.R.Janardhan Reddy, P.Penchala Prasad, A.Vineela

Assistant Professor, Department of CSE,

G.Pulla Reddy Engg College, Kurnool, India

Abstract - In this paper, we propose a technique for embedding watermark bi-color image into color image. At the source, the handwritten signature image (bi-color image) is encoded at the end of the color image. Double folded security of handwritten signature can be achieved over the untrusted network. The starting point of encoding the image data is depended on the size of the images., after that, the starting point of bi-color image encoding in the color image is stored within a four-byte block in encoded form. This four-byte location encoding is done by public key. At the target, firstly, the starting point of encoding bi-color image data is decoded by private key and then, extraction of the encoded bi-color image data from the color image is started. This technique does not require knowledge of the color image for the recovery of the handwritten signature image. At the receiver, the algorithm reconstructs the original handwritten signature image. We examine the proposed technique for embedding a bi-color handwritten signature image of 70×50 pixels size in the color host image of 512×512 pixels. Simulation results show very low visible distortions in the host image and algorithms provides high degree of robustness.

Keywords – Digital Watermarking, Invisible Watermarking, Data Hiding, untrusted network, public key, private key, security and transmission.

I. INTRODUCTION

With onset of the World Wide Web, authors of digital media can easily distribute their works by making them available on Web pages or other public forums. Anyone having access to those forums can copy the author's media. By the nature of digital media, a copy is an exact, perfect duplicate of the original. This brings to front a potential problem. One of the methods for authors for claiming ownership rights of digital media if multiple persons have exact copies is to embed additional information and only distribute the media that contains this additional information. The embedded information is known as a watermark can provide, for example, information about the media, the author, copyright, or license information. Interest in digital watermarks has grown out of an increasing interest in intellectual property and copyright protection. Our discussion will focus primarily on the watermarking of digital images, though digital video, audio, and documents are also routinely watermarked. Digital Watermarking is the technique for embedding information into a digital signal. A general watermarking scheme definition can be found at [1]. The watermarking technique provides a persistent link between the authenticator and the content it authenticates. The Digital Image Watermarking (will be referred to as watermarking for rest of the paper) can be classified into two categories - Visible and Invisible Watermarking [4], our focused area is Invisible Watermarking. In this kind of watermarking the information is added as digital data to the original, but it cannot be perceived as such. Moreover it is highly sensitive to any type of data alteration and tampering can easily be detected [2]. Data hiding can be defined as the process by which a message signal or image is imperceptibly embedded into a host or cover to get a composite signal. The general framework of a data hiding system is shown in Figure-1. Behind some year back the research work on invisible image watermarking has started. At that time the researchers were restricted in the boundary of grayscale host image and black & white watermark image. With the development of technology, the working domain was getting increased. Some of the researchers were able to embed grayscale watermark image in grayscale host image. Regarding this a thorough survey has been found in [4]. The technique for embedding information into image, which could survive attacks on the untrusted network, is proposed. A significant issue here is to embed bytes without causing visual degradation to the host image. This requires embedding data in such a way that adapts with the local characteristics of an image. Next, a hybrid digital embedding technique is proposed for hiding an image into another image in such a way that the quality of the recovered image improves significantly. The proposed technique can be applied for ownership protection, copy control, annotation and authentication of digital media. This work is specifically focused on protection and authentication of Handwritten Signature. The design of this technique is based on extensive analytical as well as experimental modeling of the data-hiding process.

The proposed framework also ensures the following:

- The extracted watermark remains intact. i.e., image authentication is guaranteed.
- Quality of generated watermarked image is much improved with respect to others.[5]
- The security issues [3] related to the watermark by using the keys.

In current communication, the technique for embedding the watermark and technique for extraction of the watermark is discussed in Section III. Experimental analysis is reported in Section IV and Conclusions are made in Section V.

II. RELATED WORKS

The advent of digital age has destroyed the security and protection of digital multimedia information. To protect this numerous schemes have been proposed in the line of Data Hiding Techniques. Some of them on what have been concentrated are discussed hereunder. Pradosh Bandyopadhyay, Soumik Das, Shauvik Paul propose [6] a framework that will form the watermark from the host image itself by combining the unique zone of the host image, and then it will be embedded to the host image with our already reported LSB scheme. It allows a user with an appropriate secret key and a hash function to verify the authencity, integrity and ownership of an image. If a forger performs the watermark extraction with an incorrect key and inappropriate hash function, the user obtains an image that resembles noise. In this manner they are providing an integrated solution for ownership authentication where the watermark is unique for that particular host image, thus the authentication is ensured in an efficient way. At the watermark extraction end, they used blind extraction method, i.e., neither the host image nor the watermark image is required at the time of watermark extraction.

Computer scientists Samir Kumar Bandyopadhyay, Debnath Bhattacharyya and Anindya Jyoti Pal examine [8] digital watermarking, the process that embeds data called a watermark into an object such that the watermark can be detected and extracted later to make an assertion about the object. Watermarking is either “visible” or “invisible”. They said that the key techniques involve using secure functions to generate and embed an image mark that is more detectable, verifiable, and secure than existing protection and detection techniques.

Yongjian Hu and Byeungwoo Jeon have proposed a reversible visible watermarking algorithm to satisfy a new application scenario where the visible watermark serves as a tag or ownership identifier, but can be completely removed to resume the original image data. It includes two procedures: data hiding and visible watermark embedding. In order to losslessly recover both the watermark-covered and nonwatermark- covered image contents at the receiver end, the payload consists of two reconstruction data packets, one for recovering the watermark covered region, and the other for the nonwatermark- covered region. The data hiding technique reversibly hides the payload in the image region not covered by the visible watermark [10].

Debnath Bhattacharyya and Deepsikha Choudhury and Samir Kumar Bandyopadhyay have proposed [11] a data hiding method where the size of the carrier image must be double (or more) the size of source image. If necessary then additional bytes or noise has to be injected into carrier image to attain the required size. To do this, the header is updated by the new value. Rehab H. Alwan, Fadhil J. Kadhim, and Ahmad T. Al- Taani,, have explained a method with three main steps [9]. First, the edge of the image is detected using Sobel mask filters. Second, the least significant bit LSB of each pixel is used. Finally, a gray level connectivity is applied using a fuzzy approach and the ASCII code is used for information hiding. The prior bit of the LSB represents the edged image after gray level connectivity, and the remaining six bits represent the original image with very little difference in contrast. The given method embeds three images in one image and includes, as a special case of data embedding, information hiding, identifying and authenticating text embedded within the digital images. Upamanyu Madhow, B. S. Manjunath and Shivkumar Chandrasekaran shown that the hidden data can be recoveredreliably under attacks, such as compression and limited amounts of image tampering and image resizing.

The three main findings are:

1) In order to limit perceivable distortion while hiding large amounts of data, hiding schemes must use image-adaptive criteria in addition to statistical criteria based on information theory.

2) The use of local criteria to choose where to hide data can potentially cause desynchronization of the encoder and decoder. This synchronization problem is solved by the use of powerful, but simple-to-implement, erasures and errors correcting codes, which also provide robustness against a variety of attacks.

3) For simplicity, scalar quantization-based hiding is employed, even though information-theoretic guidelines prescribe vector quantization-based methods. However, an information- theoretic analysis for an idealized model is provided to show that scalar quantization-based hiding incurs approximately only a 2-dB penalty in terms of resilience to attack [7].

Min Wu and Bede Liu, June, proposed [12] a new method to embed data in binary images, including scanned text, figures, and signatures. The method manipulates “flippable” pixels to enforce specific blockbased relationship in order to embed a significant amount of data without causing noticeable artifacts. Shuffling is applied before embedding to equalize the uneven embedding capacity from region to region. The hidden data can be extracted without using the original image, and can also be accurately extracted after high quality printing and scanning. Yusuk Lim, Changsheng Xu and David Dagan Feng, described the web-based authentication system consists of two parts: one is a watermark embedding system and the other is authentication system. In case of watermark embedding system, it is installed in the server as application software that any authorized user, who has access to server, can generate watermarked image. The distribution can use any kind of network transmission such as FTP, email etc. Once image is distributed to externally, client can access to authentication web page to get verification of image.

III. TECHNIQUE FOR EMBEDDING WATERMARK AND EXTRACTION OF WATERMARK

An Embedding high volume of information into images without causing perceptual distortion has been quite challenging. Here the problem of image-in- image hiding, in which an image, called the handwritten signature image, is to be embedded into another image, called the host image, to get a composite image is considered. On experiment, improved image quality after decoding at the target is obtained. Here hybrid data-hiding technique resulting into invisible watermarking is used.

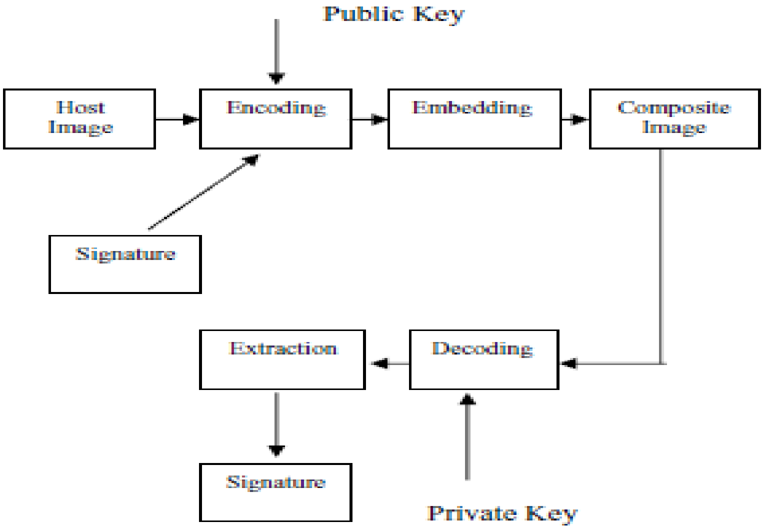


Figure - 1 Block Diagram for Embedding and Extraction

Hybrid digital-embedding algorithm at source:

```
1. Open Host and Signature Image files in Input Mode
2. Open Target Image (Composite) file in Output Mode
3. Initialize Integer i by 0; byte byte1, byte2 by 0; Integer size1, size2 by 0;
4. Find sizes of Host and Signature Images and store to size1 and size2
5. Compute size = ((size1 - size2) - 1)
6. While (not end of Host Image file)
7. Loop
8. read byte from Host Image file and store to byte1
9. increase i by 1
10. if ((i >size) and (not end of Signature Image file)) Then
11. While (not end of Signature Image file)
12. Loop
13. read byte from Signature Image file and store to byte2
14. read byte from Host Image file and store to byte1
15. Write byte2 to Target Image file
16. increase i by 1
17. End Loop
18. else
19. Write byte1 to Target Image file
20. End If
21.End Loop
22.Close Host, Signature and Target Image files
```

Basic idea is that starting point (byte) of encoding Signature Image (bi-color) file to Host Image (color) file is calculated. In the algorithm this is computed and stored in the size. This byte position is a value and that is converted to binary number. For every bit position the weight is taken and in the super increasing order in combination, denoted as ‘easy key’ or private key. Public key is extracted from this private key, using ‘Knapsack Algorithm’. By this public key the computed byte position is encoded and stored in the 4- byte (55 to 58th bytes) of the Host Image file. In this encoding technique both the file will end at the same point.

Handwritten Signature Extraction Algorithm at target:

```
1. Open Composite Image file in Input Mode
2. Open Recovery Image file in Output Mode
3. Initialize Integer i by 0, byte byte1 by 0
4. While (not end of Composite Image file)
5. Loop
6. read byte from Composite Image file and store to byte1
7. increase i by 1
```

```
8. if (i >size) Then
9. Write byte1 to Recovery Image file
10. End If
11.End Loop
12.Close Composite and Recovery Image files
```

In this algorithm ‘size’ is calculated from 4-byte (55 to 58th bytes) of the Composite Image f ile which already encoded using public key. Decoding is done by private key or ‘easy key’. Creation of Recovery Image file starts from decoded byte position of the Composite Image file and end at the end of Composite Image file.

IV. Result and Explanation

Here in this work the Superincreasing Knapsack for encoding and decoding the byte position from where the embedding starts is used. The Superincreasing Knapsack Problem is an easy knapsack problem in which the weights are in a Superincreasing sequence. A Superincreasing sequence is one in which the next term of the sequence is greater than the sum of all preceding terms. For example, the set {1, 2, 4, 9, 20, 38} is Superincreasing, but the set {1, 2, 3, 9, 10, 24} is not because 10 < (1+2+3+9). It is easy to solve a Superincreasing knapsack. Simply take the total weight of the knapsack and compare it with the largest weight in the sequence. If the total weight is less than the number, then it is not in the knapsack. If the total weight is greater than the number, it is in the knapsack. Subtract the number from the total, and compare with the next highest number. Keep working this way until the total reaches zero. If the total doesn't reach zero, then there is no solution.

Example: Superincreasing increasing sequence: e.g. {1, 2, 4, 10, 20, 40}.

Multiply all the values by a number, n, modulo m. The modulus should be a number greater than the sum of all the numbers in the sequence, for example, 110. The multiplier should have no factors in common with the modulus. So let's choose 31.

Normal knapsack sequence would be
1 * 31 mod 110 = 31;
2 * 31 mod 110 = 62;
4 * 31 mod 110 = 14;
10 * 31 mod 110 = 90;
20 * 31 mod 110 = 70;
40 * 31 mod 110 = 30.
So the public key is: {31, 62, 14, 90, 70, 30} and the private key is {1, 2, 4, 10, 20.40}.

Let's try to send a message that is in binary code: 100100111100101110.
The knapsack contains six weights so we need to split the message into groups of six:
100100111100101110.
This corresponds to three sets of weights with totals as follows
100100 = 31 + 90 = 121
111100=31+62+14+90=197
101110=1+14+90+70=205.
So the coded message is 121 197 205.

Now the receiver has to decode the value. For example here we consider the first set value alone ie.121. The person decoding must know the two numbers 110 and 31 (the modulus and the multiplier). Let's call the modulus “m” and the number you multiply by “n”. We need n^-1 (n-1), which is a multiplicative inverse of n mod m, i.e. n (n^-1) = 1 mod m. In this case we have calculated n^-1 to be 71 (inverse of 31 mod 110, to find out Extended Euclidean Algorithm is used). Then multiply each of the codes 71 mod 110 to find the total in the knapsack which contains {1, 2, 4, 10, 20, 40} and hence to decode the message.

The coded value is 121: 121 * 71 mod 110 = 11 =100100.

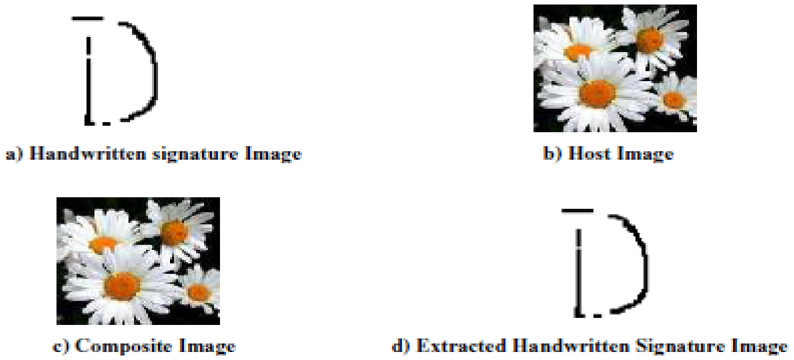
a) Handwritten signature Image

b) Host Image

c) Composite Image

d) Extracted Handwritten Signature Image

Figure - 2 Tested on Sample 1



a) Handwritten signature Image

b) Host Image

c) Composite Image

d) Extracted Handwritten Signature Image

Figure - 3 Tested on Sample 2

V. Conclusion

The detailed experiment has been performed and it is found that the proposed framework is able to form watermark dynamically from the host image. In such a manner the ownership authentication is guaranteed in an efficient way. This framework is also able to embed the formed bicolor watermark into color host images and perceptually the watermark is remains invisible in the watermarked image. At the time of extraction neither the host image nor the watermark image is needed. Thus it guarantees blind extraction method. Extracted image at the target shows excellent visual quality as well as it ensures that the extracted watermark remains intact. Here private / public key is used for very special purpose, which is explained mathematically in the result part and excellent PSNR values is achieved.

References:

1. Christine I. Podilchuk, "Image-Adaptive Watermarking Using Visual Models.", Proceedings of IEEE Journal on Selected Areas in Communications, Vol. 16, no. 4, May 1998.
2. Oktay Altun, Gaurav Sharma, Mehmet U. Celik, and Mark F. Bocko, "A set Theoretic Framework for Watermarking and Its Application to Semifragile Tamper Detection", Proceedings of IEEE Transactions on Information Forensics and Security, Vol. 1, no. 4, December 2006.
3. R. Wolfgang and E. J. Delp, "A watermark for digital images," Proceedings of IEEE International Conference on Image Processing, Vol. 3, pp.219-222, 1996.
4. M. Banerjee, "Theory and Application of cellular and Automata for authentication and watermarking", PhD thesis, Jadavpur University, 2007.
5. Chang-Tsun Li and Yue Li, "Random Index Modulation Based Fragile Watermarking Scheme For Authenticating Colour Images", Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2008.
6. Pradosh Bandyopadhyay, Soumik Das,Shauvik Paul, "A Dynamic Watermarking Scheme for Color Image Authentication", International Conference on Advances in Recent Technologies in Communication and Computing, 2009.
7. Upamanyu Madhow, B. S. Manjunath and Shivkumar Chandrasekaran, "Robust Image-Adaptive Data Hiding Using Erasure and Error Correction", IEEE Transaction on Image Processing, Vol. 13, No. 12, December 2004.
8. Samir Kumar Bandyopadhyay, Debnath Bhattacharyya and Anindya Jyoti Pal, "Secure Delivery of Handwritten Signature", ACM Ubiquity, Vol. 7 Issue. 40 October 16, 2006.
9. Rehab H. Alwan, Fadhil J. Kadhim, and Ahmad T. Al-Taani, "Data Embedding Based on Better Use of Bits in Image Pixels", International Journal of Signal Processing Vol 2, No.2, 2005, ISSN 1304-4494.
10. Yongjian Hu and Byeungwoo Jeon, "Reversible Visible Watermarking and Lossless Recovery of Original Images", IEEE Transactions on Circuits and Systems for Video Technology, Vol.16, No. 11, November, 2006.
11. Debnath Bhattacharyya, Deepsikha Choudhury and Samir Kumar Bandyopadhyay, "Bi- Color Nonlinear Data Embedding and Extraction of Handwritten Signature", IEEE Electro Information Technology Conference, EIT-2007, May 17-20, 2007, Illinois Institute of Technology, Marriott O'Hare Chicago, Illinois, U.S.A.