An Image Authentication Scheme Based On Digital Signatures *

Min-Shiang Hwang   Yuan-Liang Tang   Ching-Rong Yang

Department of Information Management †
Chaoyang University of Technology
168, Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@cyut.edu.tw
Fax: 886-4-23742337

October 31, 2012

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC90-2213-E-324-004.

An Image Authentication Scheme Based On Digital Signatures

Abstract

It is important to protect digital pictures and detect tampered image locations in digital Cyberspace. In this paper, we propose an image authentication scheme based on digital signatures. The proposed scheme is capable of detecting if certain blocks of an image have been altered. The block can be as small as 86 image pixels.

Keywords: Authentication, Cryptography, Digital signature, Image authentication.

1 INTRODUCTION

It is necessary to protect digital media because information such as images; audio/video data, text, and software are transmitted via the Internet. Consider the following case. When an image is used as a piece of evidence in the prosecution of a crime, the court must make sure that the evidence has not been altered. Without this guarantee, the image can not be used as legal evidence. Such an image can be a matter of life and death in a criminal case. The ability to accurately verify the authenticity of an image, as well as detecting the altered locations in a tampered image is extremely important. When an image is transferred over the network, hackers may intercept it and make changes to it. Similarly, an image file produced by a digital camera [2] may be doctored for certain purposes.

Recently, many approaches [2, 6, 7] have been proposed to solve the problem of image authentication, including the legal use of images and trusted camera and medical image archiving. In these methods, the owner or receiver of the image must have a priori knowledge about the image size [6] or its look-up table [7]. For example, Wong proposed a public key watermark [6] for image verification and authentication. This approach can not only authenticate an image, but also identify the tampered location. However, one of the drawbacks is that the smallest detectable block is 172 pixels. It is thus desirable to develop a scheme that can effectively detect altered blocks smaller than 172 pixels. In this paper, we introduce a new method for image authentication, which is able to detect an altered image block as small as 86 pixels.

An effective authentication and verification scheme should have the following features:

- Determine if is the image or data has been modified.
- Enable a defense for any attack.
- Be independent from the image size or the look-up table.

Our method used the RSA cryptosystem to generate and put a signature into the least significant bits (LSB) of each block. This scheme can be applied to a "secure" digital camera [2] for authenticating and detecting altered image pixels.

This paper is organized as follows. Section 2 introduces the basic idea of how to encrypt image blocks into LSB. Section 3 provides some experimental results. The analysis is performed in section 4, and section 5 gives the concluding remarks.

2 OUR SCHEME

The main idea of our method is to perform set operations to detect altered blocks. Our scheme is based on the RSA signature with 512 bits [1]. Five hundred and twelve bits are used for protecting each image block. This method can detect altered locations as sub-blocks. The following discussions are focused on gray level images. For color images, the same technique applies.

Let I_{mn} be a gray level image of size m by n pixels. The most significant bits (MSB) and least significant bits (LSB) of each pixel in our scheme are 5 and 3 bits, respectively. Since each signature block is 512 bits in the RSA cryptosystem, we require 172 pixels (as a block) to input a signature. The total number of MSBs and LSBs in a block (172 pixels) are 860 and 516 bits, respectively. In a straightforward method, we can hash the 860 bits (an MSB block) to 516 bits. A signature can then be made with 516 bits and placed into the LSB block using the RSA cryptosystem. If the block has not been altered, the MSBs of the signature will be equal to the LSBs. If the MSB of the signature are not equal to the LSBs, we can be certain that a portion of this block has been altered. This straightforward method only can determine if the block (172 pixels) has been altered. In this section, we will propose a method that can reduce the detectable size to 86 pixels. This method consists of 6 steps as follows.

Step 1: Collect and construct the first block.

According to the RSA standard, at least 512 bits are necessary for an encryption to effectively resist any attack. Because the LSB is defined as 3 bits/pixel, 172 pixels must be collected to form a signature ($172 \times 3 = 516$ bits). Now every first 5 bits (MSB) of the 172 pixels are collected to form the block b_1, which represents the image data. The final 3 bits (LSB) of each pixel are collected to form w_1, i.e. the signature. In our method, b_1 is further divided into two 86 pixel sub-blocks, denoted sb_1 and sb_2, as shown in Figure 1, and then all bits in w_1 are set to zero. The block with the signature structure is illustrated in Figure 2(a).

Step 2: Second stage block collection and construction. In this

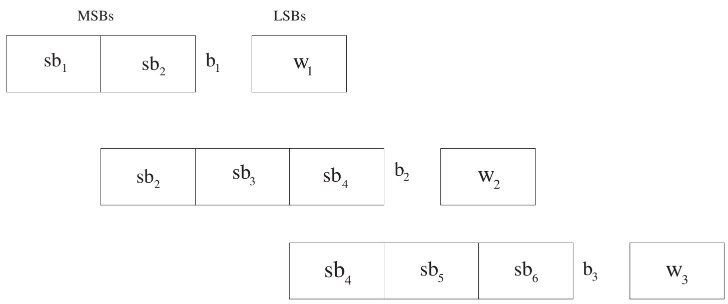

Figure 1: Blocks with MSBs and LSBs structure.

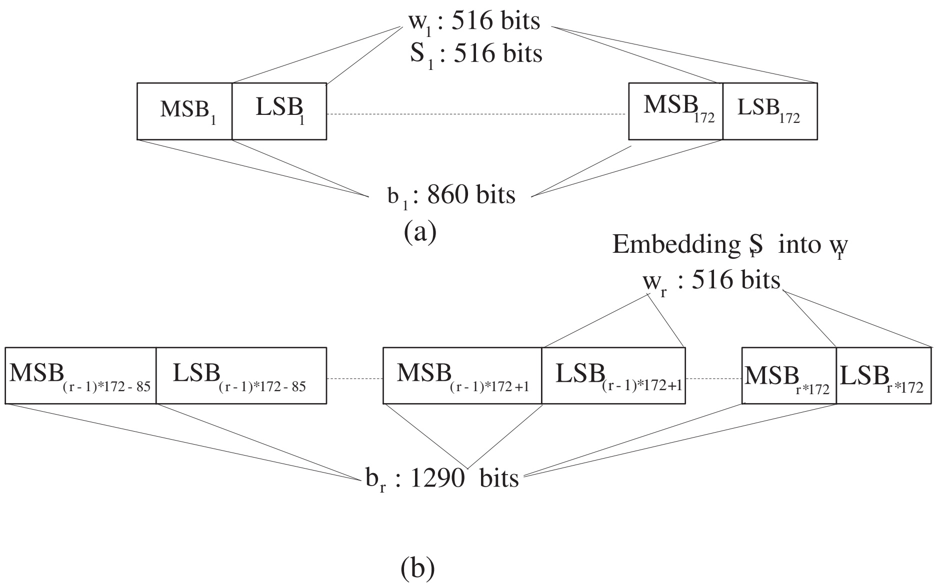

Figure 2: Blocks with signature S_i structure. Here, $H(b_i) = Hb_i$ and $S_i = E_d(Hb_i), i = 1, 2, \cdots, r$.

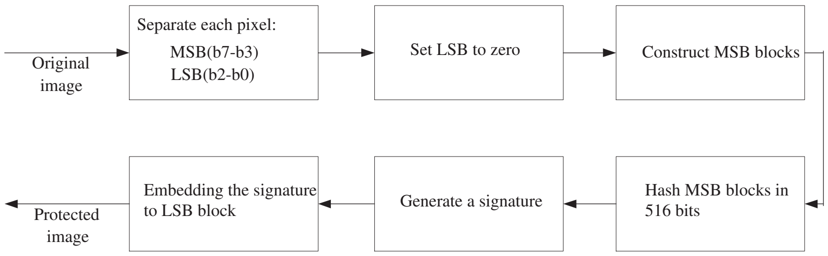

Figure 3: Embedding a signature into an image.

step, we denote the second to the last block as b_2 to b_r, respectively, where $r = \lceil \frac{m \times n}{172} \rceil$. Every block b_r is divided into three sub-blocks ($sb_{2(r-1)}$, sb_{2r-1}, sb_{2r}). An example of blocks with a signature structure is shown in Figures 1 and 2(b). The steps for embedding a signature into an image are shown in Figure 3.

In the previous steps, 1 and 2, the block $b_1 = sb_1 + sb_2$, $b_2 = sb_2 + sb_3 + sb_4$, $\cdots$, $b_r = sb_{2(r-1)} + sb_{2r-1} + \cdots + sb_{2r}$, where + denotes concatenation. Since, the size of sub-block sb_i is equal to 86 pixels, the number of pixels in b_1 is 172 and the number of pixels in the others b_i, $i = 2, \cdots, r$, are equal to 258.

Step 3: One-way hash function process. Let H be a one-way hash function, such as MD5 [3]. H can be formulated as:

$$H(b_i) = Hb_i, i = 1, 2, \cdots, r. \tag{1}$$

Here, the length of b_1 is 860 bits (= 172 × 5); The lengths of b_i, $2 \le i \le r$, are 1290 bits (= 258 × 5); The lengths of Hb_i, $i = 1, 2, \cdots, r$, are 516 bits.

Step 4: Generating and embedding the signature. The RSA cryptosystem is used to generate the digital signature S_i as follows.

$$S_i = E_d(Hb_i), i = 1, 2, \cdots, r. \tag{2}$$

Here, d denotes the system's private key. E denotes the RSA algorithm [4] with 516 bits. The length of S_i is 516 bits. The signatures S_i are embedded into w_i, $i = 1, 2, \cdots, r$.

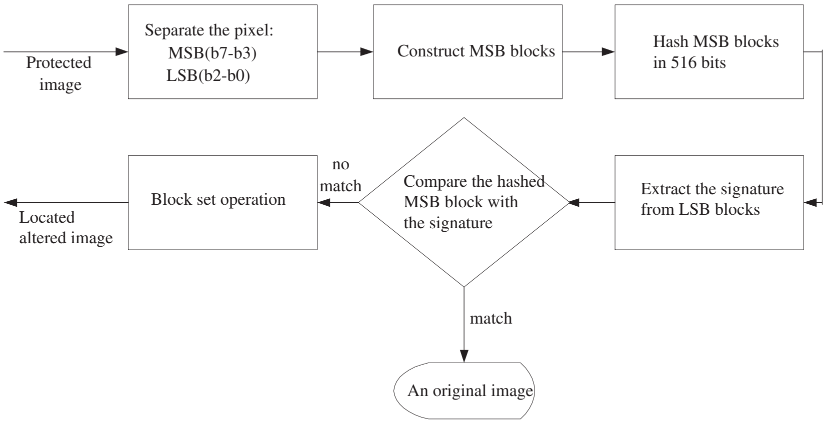

Figure 4: Extracting a signature from a protected image.

Step 5: Image authentication. When an image is received, it is necessary to perform an authentication process and determine if and where the image has been altered. The authentication process consists of 6 consecutive tasks as delineated in Figure 4. The first three tasks are the same as in Step 1. i.e., the hash values HB_i', $i = 1, 2, \cdots, r$ are obtained. The fourth task uses the RSA cryptosystem [4] to recover the signature S_i as follows.

$$DHb_i = D_e(S_i), i = 1, 2, \cdots, r. \tag{3}$$

Here, e denotes the system's public key. D denotes the RSA deciphering algorithm. If the image has not been altered, each pair of Hb_i' and DHb_i should match. Otherwise, the image has been altered.

Step 6: Identify the tampered location. If one or more pairs Hb_i' and DHb_i do not match, we know that the image has been altered. In this step, we propose a method to locate the altered location within the range of 86 pixels. The following rule is used to confine the altered location. We define three sets as follows.

- $A = \{sb_i|$ a set of the pairs Hb_i' and DHb_i do not match. $\}$.

- $B = \{sb_j |$ a set of the pairs Hb'_j and DHb_j match. $\}$.

- $C = \{sb_k |$ a set of probable altered sub-blocks. $\}$. In other words, $C = A - B$. Here, $-$ denotes a difference set operation.

For example, if all Hb'_i and DHB'_i match except for Hb'_2 and DHB'_2, we know that the block b_2 has been altered. By our rule, $A = \{sb_2, sb_3, sb_4\}$, $B = \{sb_1, sb_2, sb_4, sb_5, \cdots\}$, and, $C = A - B = \{sb_3\}$. Therefore, we obtain a probable altered sub-block sb_3. This means that the altered locations can be detected at a range of 86 pixels.

3 EXPERIMENTAL RESULTS

The experimental result is shown in Figure 5. We used the "Lena" image (256 $\times$ 256 pixels) as the test image. Figure 5(a) is an original image. Figure 5(b) is an image with a signature embedded into the original image. Figure 5(c) is an image with a black eyeball that has been altered. Figure 5(d) is a detection image using our scheme. Where the image color is black denotes that this location has not been altered. The other color denotes a location that has probably been altered. In this experimental result, the length of the tampered area is 86 pixels.

4 ANALYSIS

Some tampered image cases are analyzed in this section. The sub-blocks in shadow denote that these sub-blocks have been altered.

Case 1: Sub-block sb_1 is altered (shown in Figure 6).

Block b_1 does not match in Step 5 of Section 2, while b_2, b_3, and other blocks do match. Using our method in Step 6 of Section 2,

(a) (b) (c) (d)

Figure 5: Experimental result

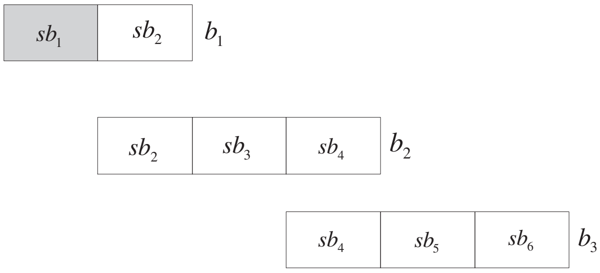

Figure 6: Case 1: a sub-block is altered.

- $A = \{sb_1, sb_2\}$.
- $B = \{sb_2, sb_3, sb_4, \cdots, sb_r\}$.

The probable tampered sub-block is $C = A - B = \{sb_1\}$.

In the straightforward method (described in Section 2), the detected tampered sub-blocks are sb_1 and sb_2. The length of detected tampered pixels is 172 pixels. However, the length of the detected tampered pixels is only 86 pixels using our method.

Case 2: Sub-block sb_1 and sb_2 are altered (shown in Figure 7).

Blocks b_1 and b_2 do not match in Step 5 of Section 2, while b_3 and other blocks do match. Using our method in Step 6 of Section 2,

- $A = \{sb_1, sb_2, sb_3, sb_4\}$.
- $B = \{sb_4, sb_5, sb_6, \cdots, sb_r\}$.

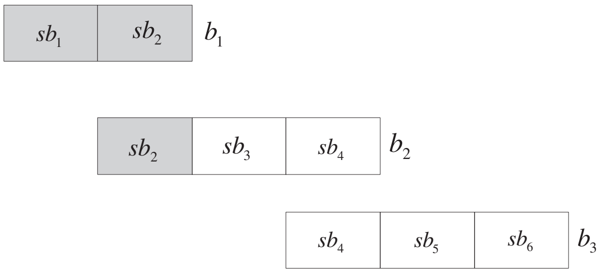

Figure 7: Case 2: Two sub-blocks in a block are altered.

The probable tampered sub-block is $C = A - B = \{sb_1, sb_2, sb_3\}$.

In the straightforward method (described in Section 2), the detected tampered sub-blocks are sb_1 and sb_2. The length of detected tampered pixels is 172 pixels. However, the length of the detected tampered pixels is 258 pixels using our method. We conclude that if two sub-blocks in a block are altered, then the straightforward method is better than our method.

Case 3: Sub-blocks sb_2, and sb_3 are altered (shown in Figure 8).

Blocks b_1 and b_2 do not match in Step 5 of Section 2, while b_3 and other blocks do match. Using our method in Step 6 of Section 2,

- $A = \{sb_1, sb_2, sb_3, sb_4\}$.
- $B = \{sb_4, sb_5, sb_6, \cdots, sb_r\}$.

The probable tampered sub-block is $C = A - B = \{sb_1, sb_2, sb_3\}$.

In the straightforward method (described in Section 2), the detected tampered sub-blocks are sb_1, sb_2, sb_3, and sb_4. The length of detected tampered pixels is 344 pixels. However, the length of the detected tampered pixels is 258 pixels using our method. We conclude that if two consecutive sub-blocks sb_{2i} and sb_{2i+1} in different blocks b_i and b_{i+1}, respectively, are altered, then our method is better than the straightforward method.

Case 4: Sub-block b_2 is altered (shown in Figure 9).

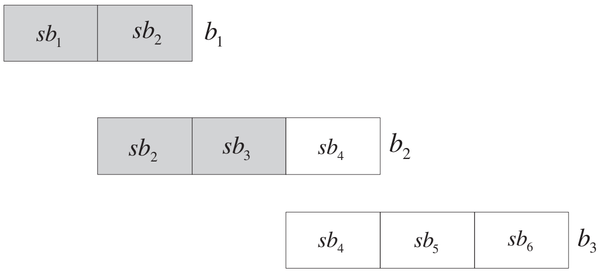

Figure 8: Case 3: Sub-blocks sb_2 and sb_3 are altered.

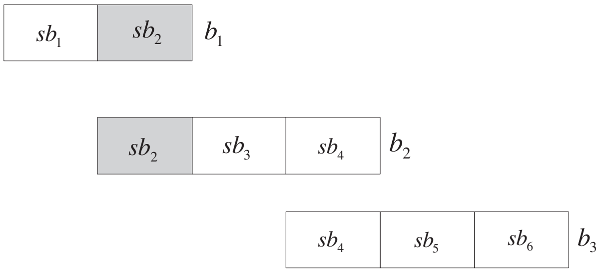

Figure 9: Case 4: Sub-block b_2 is altered.

Blocks b_1 and b_2 do not match in Step 5 of Section 2, while b_3 and other blocks do match. Using our method in Step 6 of Section 2,

- $A = \{sb_1, sb_2, sb_3, sb_4\}$.
- $B = \{sb_4, sb_5, sb_6, \cdots, sb_r\}$.

The probable tampered sub-block is $C = A - B = \{sb_1, sb_2, sb_3\}$.

In the straightforward method (described in Section 2), the detected tampered sub-blocks are sb_1 and sb_2. The length of detected tampered pixels is 172 pixels. However, the length of the detected tampered pixels is 258 pixels using our method. We conclude that if a sub-block sb_{2i} in a block b_i is tampered, then the straightforward method is better than our method.

Case 5: Sub-block sb_3 is altered (shown in Figure 10).

Block b_2 does not match in Step 5 of Section 2, while b_1, b_3, and other blocks do match. Using our method in Step 6 of Section 2,

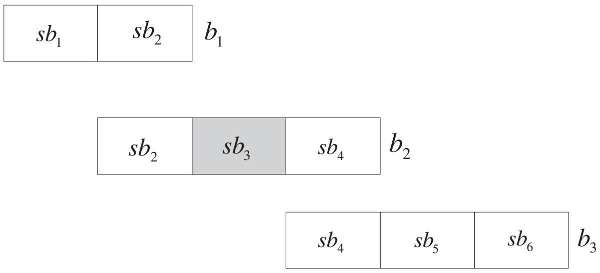

Figure 10: Case 5: Sub-block sb_3 is altered.

- $A = \{sb_2, sb_3, sb_4\}$.
- $B = \{sb_1, sb_2, sb_4, sb_5, sb_6, \cdots, sb_r\}$.

The probable tampered sub-block is $C = A - B = \{sb_3\}$.

In the straightforward method (described in Section 2), the detected tampered sub-blocks are sb_3 and sb_4. The length of detected tampered pixels is 172 pixels. However, the length of the detected tampered pixels is 86 pixels using our method. We conclude that if a sub-block sb_{2i-1} in a block b_i is altered, then our method is better than the straightforward method.

Although, our method is not better the straightforward method in some cases, the main merit of our scheme is that our scheme can detect a tampered location within 86 pixels.

5 CONCLUSIONS

In this paper, we have presented a new method for image authentication, which can detect a tampered block with size as small as 86 image pixels. In addition, this method has the following features:

- A prior knowledge is not necessary for detecting tampering in an image.
- Any method that an attacker uses to tamper with an image is detectable by our scheme.

References

[1] D. Atkins, M. Graff, A. K. Lenstra, and P. C. Leyland. The magic words are squeamish ossifrage. *In Advance in Cryptology, Asiacrypt '94*, pages 263–277, 1994.

[2] G. L. Friedman. The trustworthy digital camera: Restoring credibility to the photographic image. *IEEE Trans. On Consumber Electronic*, 1993.

[3] R. L. Rivest. The md5 message digest algorithm. *Internet RFC 1321*, April 1992.

[4] R. L. Rivest, A. Shamir, L. Adleman, and P. C. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, February 1978.

[5] Bruce schneier. *Applied Cryptography.* Proc. IEEE Intl. Conf. On Image Processing, second ed edition, 1996.

[6] P.W. Wong. A public key watermark for image verification and authentication. *Proc. IEEE Intl. Conf. On Image Processing*, pages 455–459, Oct 1998.

[7] M. Wu and B. Liu. Watermarking for image authentication. *Proc. IEEE Intl. Conf. On Image Processing*, pages 437–441, Oct 1998.