TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 8, August 2014, pp. 6386 ~ 6392
DOI: 10.11591/telkomnika.v12i8.5737

Image Protection by Intersecting Signatures

Chun-Hung Chen[1], Yuan-Liang Tang*[2], Wen-Shyong Hsieh[1,3], Min-Shiang Hwang[4]
[1]Department of Computer Science and Engineering, National Sun Yat-sen University
[2]Department of Information Management, Chaoyang University of Technology
[3]Department of Computer Science and Information Engineering, Shu-Te University
[4]Department of Computer Science and Information Engineering, Asia University
*Corresponding author, e-mail: yltang@cyut.edu.tw

Abstract

In this paper, we propose an exact image authentication scheme that can, in the best case, detect image tampering with the accuracy of one pixel. This method is based on constructing blocks in the image in such a manner that they intersect with one another in different directions. Such a technique is very useful to identify whether an individual image pixel has been tampered with.

Keywords*: information security, digital watermarking, image authentication, digital signatures*

1. Introduction

As digital technologies advance, more and more publications are produced in digital formats and transmitted via the Internet. Accompanying such advance, however, unauthorized use, illegal copying, and malicious modification of digital products have become serious problems. Researchers thus try to find various ways to protect digital products; solutions include copyright assertion, content authentication, etc. In the area of image content authentication, the integrity of an image is regarded very important and must therefore be realized. A common approach is the use of digital watermarking techniques. Digital watermarking serves many purposes, for example, proof of ownership, content authentication, copy control, and so on.

Researchers have developed various image authentication techniques to detect if an image has experienced unauthorized modification. Some of them can only detect whether the image as a whole has been altered. Others may have the additional capability to detect if a certain part of the image has been tampered with. Liu *et al.* [1] studied the Zenike moment values which are generated from low DWT subbands. They found that the quantized values are robust to common processing operations but fragile to malicious attacks. Therefore, they embedded the watermark by quantizing the Zernike moment values, and the locations (i.e., blocks) suffered from malicious attacks can be identified through examining the extracted values. Their method has moderate robustness against JPEG compression. In Rawat and Raman's scheme [2], two chaotic maps are used in order to enhance the security of the watermarked images. The pixels in the image are disturbed using the first chaotic map and are further separated into bit planes with the least significant bit used for watermark embedding. A binary watermark is scrambled by the second chaotic map. The watermarked images can avoid counterfeiting attacks. Xi'an [3] scrambled a bi-level watermark by the Arnold transform, and the Human Visual System is used to determine the quantization step. The scrambled watermark is then inserted into the low DWT coefficients. Tamper areas can then be localized by comparing the extracted and the original watermarks. Patra *et al.* [4] convert the images into the DCT domain and quantize the low-frequency coefficients according to the target levels determined by the Chinese Remainder Theorem. Their method is computationally efficient and is able to withstand such attacks as JPEG compression, sharpening, and brightening. Qi *et al.* [5] used two content-based watermarks to protect the images. One of them is generated by an edge detector for the purpose of detecting tiny changes, and the other is generated from the relationship between the wavelet coefficients for localizing tampered regions. Both watermarks are embedded into middle- and high-frequency DWT coefficients. Finally, the generated watermarks and extracted watermarks are compared to authenticate the image, and a malicious attack is identified if error pixels are clustered together. Their method is robust against several

Received February 3, 2014; Revised April 6, 2014; Accepted April 25, 2014

image processing operations, including JPEG compression. In Wu's work [6], the image is divided into blocks, and all hashes derived from the MSBs of each block are further encoded using an error correcting code (ECC). The parities, rather than the codewords, of the ECC are separated and embedded into the LSBs of each block. During authentication, the original hashes can be recovered if the number of tampered blocks is less than a threshold. The hash of a block is produced and compared with the original to identity if the block is tampered with. This method has fine granularity on detecting tampered regions.

Although many techniques have been proposed for image authentication, most of them can only detect if an image or part of it, as a whole, is modified; very few can identify image tampers down to the granularity of one-pixel level. In some applications, such ability could be extremely essential. For example, if an image is used as a critical piece of evidence in the court or in a police investigation, a generalized answer as to whether the entire image or part of it is altered to some degree may not be acceptable by the law. To be exact, it may be mandatory that the image should not allow for even tiny modification ever since it was taken. In this paper, we propose an authentication scheme that is able to detect and locate image tampers with the accuracy of one pixel at the best case. This is done by first constructing linear blocks in the image in such a way that they intersect with one another in different directions. Second, a signature is created for each block for the purpose of authentication. And finally, the signature is embedded back into the image. As a result, each pixel is protected by four signatures and any tampered pixel can be pinpointed by examining its corresponding signatures. The rest of the paper is organized as follows. The proposed technique is described in Section 2, followed by experimental results in Section 3. Section 4 presents a security analysis. A comparison of detection granularity is shown in Section 5. Finally, Section 6 gives some concluding remarks.

2. Proposed Technique

2.1. Constructing the Authentication Blocks and Signature Blocks

Without loss of generality, we assume that 8-bit grayscale images are dealt with. For other formats, the same technique applies, too. The image is first divided into equal-sized $B\times B$ blocks, which are referred to as the *authentication blocks*. And then, in each authentication block, four sets of pixels are collected in four directions: horizontal, vertical, –45° and 45° wrap-around diagonals. These sets of linear blocks are referred to as *signature blocks*. Namely,

$H_i = \{p_{ij} \mid j = 0, \ldots, B-1\}$: horizontal blocks, $i = 0, \ldots, B-1$,
$V_j = \{p_{ij} \mid i = 0, \ldots, B-1\}$: vertical blocks, $j = 0, \ldots, B-1$,
$X_m = \{p_{ij} \mid i = 0, \ldots, B-1, j = (m+i) \textit{ mod } B\}$: –45° blocks, $m = 0, \ldots, B-1$, and
$Y_n = \{p_{ij} \mid i = (B+n-j) \textit{ mod } B, j = 0, \ldots, B-1\}$: 45° blocks, $n = 0, \ldots, B-1$,

Where p_{ij} denotes the image pixel and *mod* is the modulo operation. Figure 1 depicts such a construction.

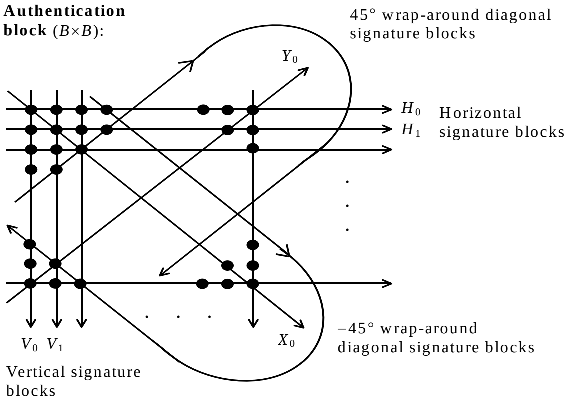
Authentication block (B×B):
45° wrap-around diagonal signature blocks
Y0
H0
H1
Horizontal signature blocks
V0 V1
X0
Vertical signature blocks
−45° wrap-around diagonal signature blocks

Figure 1. The Authentication Block and Signature Blocks

After establishing the signature blocks, a signature is created for each such block and then embedded into the image. If the DES system is used for signature generation, 64 bits are required for both the input and output data. Furthermore, if the last two bits (least-significant bits, LSBs) of each pixel are used for embedding, the size of an authentication block should be 128×128 (i.e., B = 128) since there are 4B signature blocks. Therefore, the collection of the first 6 bits of each pixel in a signature block is hashed first by such functions as MD5 or SHA to produce a 64-bit data. And then this data is encrypted using the DES system to produce a 64-bit signature, which is finally embedded back into the LSBs for the purpose of authentication. The above procedure is repeated on all signature blocks.

When performing authentication, the construction of the authentication blocks and signature blocks are repeated as before, followed by the same DES encryption process. Now, the results can be matched against those extracted from the LSBs in the image. The mismatched blocks will be recorded in the following four sets, respectively:

$E_H = \{H_i \mid i = i_0, i_1, \ldots, i_{h-1}\}$: horizontal blocks,
$E_V = \{V_j \mid j = j_0, j_1, \ldots, j_{v-1}\}$: vertical blocks,
$E_X = \{X_m \mid m = m_0, m_1, \ldots, m_{x-1}\}$: –45° blocks, and
$E_Y = \{Y_n \mid n = n_0, n_1, \ldots, n_{y-1}\}$: 45° blocks,

Where h, v, x, and y are the respective numbers of mismatched blocks. The basic idea of the algorithm is that since each pixel is protected by four signatures and the signature blocks intersect with one another, if a specific pixel is indeed tampered with, mismatches will occur in all of its four corresponding signatures. On the contrary, if some of the corresponding signatures are matched, it can be concluded that the pixel has not been altered. Figure 2 illustrates the algorithm of tamper detection in an authentication block. Each pixel, p_{ij} ($0 \le i, j \le B-1$), in a block is checked to see if it is tampered with. This is done by examining its corresponding four signatures, i.e., horizontal, vertical, and two diagonal ones. If all four signatures mismatch, the pixel will be reported as been tampered with.

```
for (i = 0, …, B–1)
  if (H_i □□E_H)  # Horizontal signature mismatches
    for (j = 0, …, B–1)
      if (V_j □□E_V)  # Vertical signature mismatches
        m ← j–i;
        if (m < 0)
          m ← m+B;  # Wrap around
        end if
        n ← i+j;
        if (n > B–1)
          n ← n–B;  # Wrap around
        end if
        if (m □ E_X) and (n □ E_Y)
          # Both diagonal signatures mismatch
          p_ij: tampered pixel;
        end if
      end if
    end for
  end if
end for
```

Figure 2. The Algorithm of Tamper Detection

2.2. Analysis

One of the main shortcomings of most other tamper detection techniques is that they usually create a signature for an image block, and if a mismatch occurs, the block as a whole is identified as being tampered with. There is no way of distinguishing which pixel (or pixels) is the victim. The essence of the proposed scheme lies on the fact that each pixel is protected by four intersecting signature blocks. Whenever one pixel is tampered with, it causes the four corresponding signatures to mismatch and, through the intersecting structure, the tampered pixel can be easily pinpointed. In other words, if less than four mismatches occur for a pixel, we can eliminate the possibility of tampering. This method is thus very accurate in identifying tampered pixels as well as their locations in the image. There are, however, some conditions in which this scheme will make false positive reports. Figure 3 illustrates such a situation, in which the black pixels represent those pixels that have been altered by attackers. The four sets of mismatched blocks are: E_H = {H_{i1}, H_{i2}, H_{i3}}, E_V = {V_{j1}, V_{j2}, V_{j3}}, E_X = {X_{m1}, X_{m2}, X_{m3}}, and E_Y = {Y_{n1}, Y_{n2}, Y_{n3}}. It is obvious that, besides the four black pixels, the system will erroneously report the center pixel (represented by a white pixel) as a tampered one, i.e., a false positive.

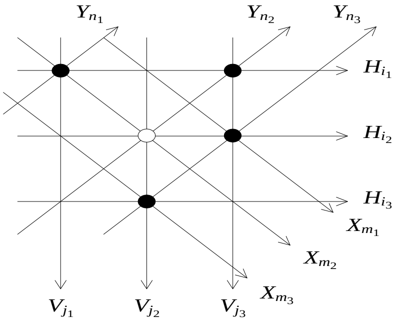

Figure 3. Example of a False Positive

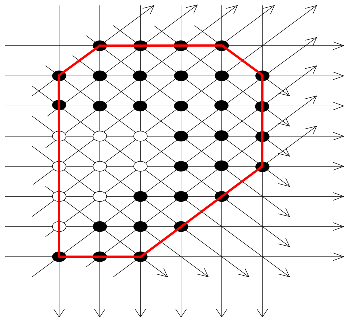

Figure 4. A Tampered Region (black pixels)

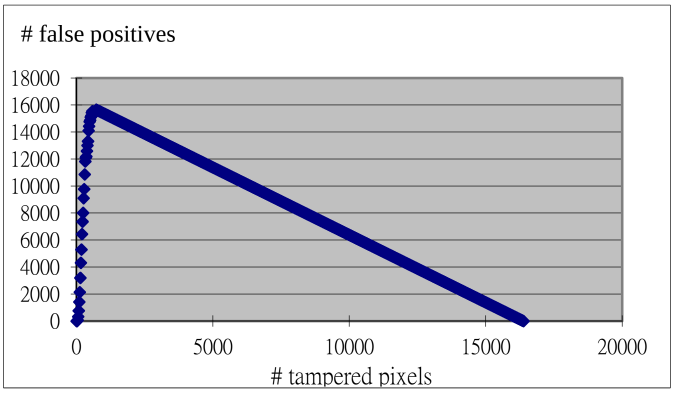

Figure 5. Number of False Positives vs Number of Randomly Tampered Pixels

Actually, the number of tampered pixels (or the size of the tampered region) determines the number of mismatch signatures. If the former increases, the latter increases, too. There is no constraint on the maximum size of a tampered region; however, the shape of the region does affect the number of false positives in the detection. Figure 4 illustrates an example, in which the black pixels represent the tampered pixels. As our algorithm identifies tampered pixels by four intersecting signatures, the set of reported pixels will form a convex shape (the red polygon in the figure). As all pixels in the convex shape are reported as tampered with, the unchanged pixels (white pixels) are false positives. The situation gets worse if the tampered pixels spread randomly across the image. The result of a simulation is shown Figure 5, in which the tampered pixels are generated randomly across the authentication block (128×128). It can be seen that the number of false positives increases rapidly as the number of tampered pixels increases, almost reaching 16,000 when the latter is only a few hundreds. Such a phenomenon verifies the above analysis that as the reported pixels form a convex shape, if the locations of the tampered pixels are randomly generated, the convex area will become very large, which results in a great number of false positives. Denoting the numbers of reported, tampered, and false positive pixels as *R*, *T*, and *F*, respectively, the following relationship holds:

$$R = T + F$$

Therefore, when *R* achieves its highest (i.e., the size of the block), increasing *T* will certainly decreases *F*, which explains the peak in Figure 5. The same false positive effect is also expected in other existing block-based authentication techniques. In practice, however, as an attacker usually tries to alter the semantics of the image, tampered pixels tend to cluster together (may be in several locations). Randomly altering the pixels is meaningless and hence not likely to happen.

2.3. Handling Irregular Image Sizes

If the image size is not multiples of the size of the authentication block, something must be done for the extra areas. Since 64 bits are required for a signature, every 32 pixels can be collected to form an individual authentication block. In those areas, however, if the signature mismatches, it can be only concluded that one or more pixels in such a block could have been altered. Figure 6 shows such a situation. The granularity of identification in those areas is thus 32 pixels.

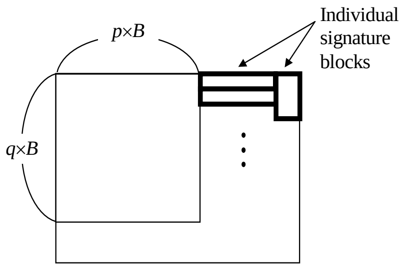

Figure 6. Handling the Situation in which the Image Size is not Multiples of *B*×*B*

3. Experimental Results

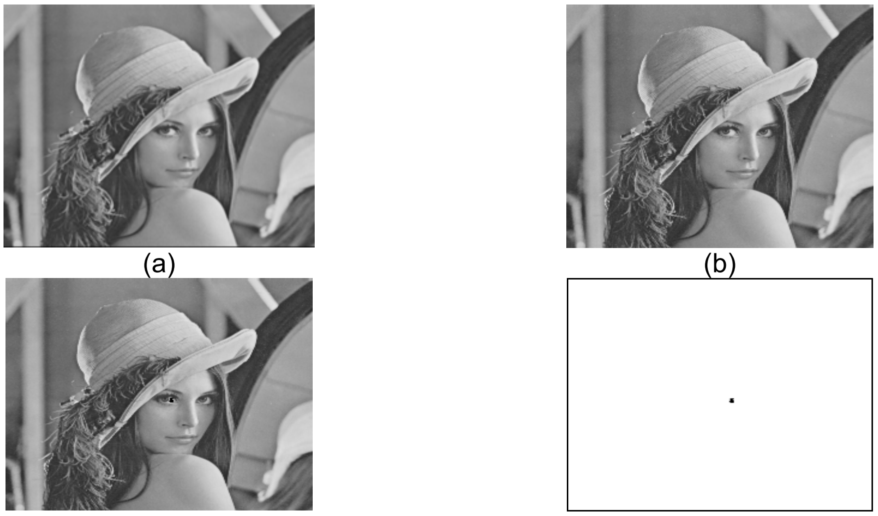

Fig. 7: (a) original image, (b) signatures embedded (PSNR = 42.7), (c) tampered image, (d) result of tamper detection

The grayscale “Lena” image is used to test the proposed algorithm: Figure 7(a) shows the original image and (b) is the version with signatures embedded. The PSNR between the original and the embedded version is 42.7, which is quite acceptable; (c) shows the image tampered in Lena’s right eye, and (d) is the result of tamper detection. The tampered pixels are correctly identified, together with only a few false positives.

4. Security Analysis

As the DES encryption system is used to generate the signatures, the proposed method is secure against the attack of manipulating individual image pixels. Three other attacks are the *search*, *collage*, and *cut-and-paste* attacks [7, 8], which are common for block-wise content authentication techniques. Because the attacked image has to maintain good visual quality, the size of the pasted blocks has to be very small in order for keeping the homogeneity of the block content. Therefore, the key requirement for these kinds of attacks to be successful is that the block size is small enough, usually less than or equal to 8×8 pixels. In the proposed method, as the size of the block is 128×128, it clearly makes these attacks infeasible. That is, even if the attacker may forge an authentic image from a database containing hundreds of thousands of authentic images, it will certainly have poor visual quality due to block effects and incorrect block content. In conclusion, our method is invulnerable to these attacks.

5. Comparison of Detection Granularity

The detection granularity of the proposed method is compared with those of Patra *et al.*’s [4], Qi *et al.*’s [5], and Wu’s [6] methods. Because the granularity of Qi *et al.*’s and Wu’s methods depends on the image size, a unified size of 256×256 pixels is used in the analysis here. The comparison is shown in Table 1, in which it is obvious that our method outperforms the others.

Table 1. Comparison of the Detection Granularity

The proposed	Patra *et al.*'s	Qi *et al.*'s	Wu’s
1×1=1 pixel	8×8=64 pixels	8×8=64 pixels	45 pixels

6. Conclusion

In this paper, we have described a technique to identify tampered pixels in an image. It is based upon dividing the image into authentication blocks and arranging linear signature blocks in such a way that they intersect at every pixel. As a consequence, each pixel is protected by four signatures and such an arrangement makes our technique capable of, in the best case, pinpointing a single altered pixel. This technique preserves the perceptual similarity of the original and the watermarked images, and it is also secure against various possible attacks. Although false positives are likely to be reported if altered pixels are spread randomly throughout the image, an attacker seems to have no reason to randomize the alterations. Therefore, our method is very useful for protecting the contents of the images at the granularity of one pixel.

References

[1] Liu H, Lin J, Huang J. *Image authentication using content based watermark*. Proc. IEEE Int. Sym. Circuits and Systems. 2005; 4: 4014-4017.

[2] Rawat S, Raman B. A chaotic system based fragile watermarking scheme for image tamper detection. *Int. J. Electron. Commun.* 2011; 65: 840-847.

[3] Xi'an Z. *A semi-fragile digital watermarking algorithm in wavelet transform domain based on Arnold transform.* Proc. IEEE Int. Conf. Signal Process. 2008: 2217-2220.

[4] Patra JC, Phua JE, Rajan D. *DCT domain watermarking scheme using Chinese Remainder Theorem for image authentication.* Proc. IEEE Int. Conf. Multimedia and Expo. 2010: 111-116.

[5] Qi X, Xin X, Chang R. *Image authentication and tamper detection using two complementary watermarks.* Proc. IEEE Int. Conf. Image Process. 2009: 4257-4260.

[6] Wu Y. *Tamper-Localization Watermarking with Systematic Error Correcting Code*. Proc. IEEE Int. Conf. Image Process. 2006: 1965-1968.

[7] Holliman M, Memon N. Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes. *IEEE Trans. Image Process*. 2000; 9: 432-441.

[8] Barreto PSLM, Kim HY, Rijmen V. Toward secure public-key blockwise fragile authentication watermarking. *IEE Proc. Vision, Image and Signal Processing*. 2002; 149: 57-62.

[9] Liu Q. An Adaptive Blind Watermarking Algorithm for Color Image. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11: 302-309.

[10] Tan X, Hu D. A Watermarking Method Based on Optimization Statistics. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11: 4794-4802.