Binary Document Images Authentication by Thinning Digital Patterns

Department of Computer Science and Information Engineering,
Tunghai University, 407 Taichung, Taiwan
{chlin, wkc, g95280074, g95280086}@thu.edu.tw

Abstract

As most documents have been digitized and transmitted over the Internet, it becomes very crucial to protect the integrity of them. Integrity is the process of ensuring that a document is changed by unauthorized users. In this paper, inspired by the Yang and Kot’s method, we proposed a novel scheme to verify the integrity of binary documents by using its digital patterns. We compute the black patterns and white patterns of a document and embed watermark information in it. The experimental result shows that by using our proposed method one can verify the integrity of a digital document, any little change made to it can be detected easily.

Keywords: binary document images, integrity protection, watermarking, thinning, black (white) patterns.

1. Introduction

Digital documents are vulnerable to illegal modifications by unauthorized parties when spreading over the Internet. In order to protect the intellectual property right and authenticate the originality, a watermark is embedded in the digital documents. Digital watermarking provides a method to control the integrity of various multimedia documents. The owner embeds watermarks to protect the integrity of their documents. If an opponent tampers the documents, the legitimate users can check the integrity of the document by verifying the embedded watermarks.

However, it is always more difficult for the protection of text documents than for colorful or gray images. Since most of text documents are in pure binary format and thus there is no enough capacity for embedding information in the document. It has become an interesting problem to undetectably embed information into a text document and then use that to verify the integrity of the document. There are some related researches in this area can be found from the literatures [1-5]. In this paper, inspired by the Yang and Kot’s method [1], we proposed a novel method for protecting the integrity of binary documents based on thinning digital patterns. We compute the black pattern and white pattern of the document and embed watermark information in it. As the document contents are altered, the patterns are changed as well. The proposed scheme thus can protect the integrity of binary documents.

This paper is organized as follows. In Section 2, we first briefly review the background of document integrity and the thinning algorithm. In Section 3 and we depict our proposed method and show the experimental results in Section 4. Finally, we have a conclusion in Section 5.

2. Background

People are used to transmit documents via the Internet using their computers or handheld devices. Digital documents are pervasively used in our daily life and the security issue has become more and more important in current society. Nowadays, cryptographic technology is booming and provides useful tool for document integrity and authentication services.

In 2005, Yang and Kot [1] proposed a novel scheme for text document authentication by embedding watermark information. It uses blind data hiding technique to preserve the connectivity of pixels in a local neighbor. First the method finds pixels possessing the flippability property. A pixel with this property, called a flappable pixel, is used to embed watermark information.

A host media is partitioned into blocks. In a block, we remove the flippable pixels and record their positions. And this is done for all the blocks in the host media Y. The left part of Y, namely Y_l, is operated using a one-way hash function to obtain H_0 which is then encrypted using the private key K_e, the result is called W_r. Afterwards, W_r is concatenated with the payload watermark W_p to form W_r'. That is, W_r = W_r || W_p Then W_r is embedded into the host media on the chosen flappable pixels to obtain the watermarked image Y_w.

To verifying the integrity of a document, we first divide the watermarked image into blocks, and extract the embedded information. The embedded information is then separated into W_p and W_r'. If W_p is different from the original payload watermark W_p, it indicates that the document has been tampered. Otherwise, we can use the same manner as in the embedding to process Y_w, by using a hash function on it to obtain H_w. Further, W_r can be decrypted and we obtain H_r. If H_r is coincident with H'_0 and W_p is coincident with W_p, the integrity of the document can be assured.
3. The Proposed Method

In the section, we introduce the proposed binary document authentication scheme by thinning the digital patterns. We compute the digital patterns of a host document image using thinning technique, and select the locations which could be used to embedding, then embed the data. The processed result could protect the integrity of the host document. To embedding and extracting the watermark information, we use the method proposed by Lu, Kot, and Cheng’s method [1].

3.1. Pattern Extraction

We use Zhang-Suen thinning algorithm [5] to compute the black pattern and the white pattern of a document. At first, we define the two functions $X(p)$ and $B(p)$ for a block, e.g., of size $3 \times 3$. In this block, shown in Figure 1, the central point $p$ has eight neighbors. $X(p)$ is the number of value changes, from 0 to 1, in the order of $p_1, p_2, p_3, \ldots, p_8$. Note that 1 indicates an object pixel and 0 is the background pixel.

$$\begin{align*}
\text{Figure 1} & \quad \begin{bmatrix} p_1 & p_2 & p_3 \\
p_4 & p_5 & p_6 \\
p_7 & p_8 \
\end{bmatrix} \\
\text{Figure 2} & \quad \begin{bmatrix} 0 & 1 & 0 \\
0 & p & 1 \\
1 & 0 & 1 
\end{bmatrix}
\end{align*}$$

And $B(p)$ is the number of nonzero neighbors to $p$, that is:

$$B(p)=p_1+p_2+p_3+\ldots+p_8.$$  

As in Figure 2, we have that $X(p)=3$ and $B(p)=4$. The Zhang-Suen (ZS, for short) thinning algorithm [5] is parallel, the new value only depends on the value of the previous one iteration. It is fast and simple for implementation. A pixel $p$ in the $3 \times 3$ block could be deleted using the following two alternating iterations:

In the first iteration:

1. $X(p)=1$
2. $2 <= B(p) <= 6$
3. $p_2 \cdot p_4 \cdot p_6=0$
4. $p_4 \cdot p_6 \cdot p_8=0$

In the second iteration:

1. $X(p)=1$
2. $2 <= B(p) <= 6$
3. $p_2 \cdot p_4 \cdot p_8=0$
4. $p_2 \cdot p_6 \cdot p_8=0$

In the block, pixels satisfying these conditions would be deleted, that is, we set $p=0$.

In this paper, we apply the ZS thinning algorithm to compute both of the black and the white patterns. The pseudo-code is as follows.

```
if \(p = 1 \&\& (2 <= B(p) <= 6) \&\& X(p) = 1\)
  if \(p_2 = 0\) \&\& \(p_4 = 0\) \&\& \(p_6 = 0\) \&\& \(p_8 = 0\)
    then set \(p = 0\)
else
  if \(p_2 = 0\) \&\& \(p_4 = 0\) \&\& \(p_6 = 0\) \&\& \(p_8 = 0\)
    then set \(p = 0\)
```

3.2. Embedding Process

In this section, we describe the embedding process of our proposed scheme. We define some parameters:

- $W_p$: Payload watermark
- $K_p$: Verifier’s public key
- $E_i()$: Encrypt function
- $I_w$: Encrypted data
- $I_h$: Hash code

The embedding step is described as follows.

**Input:** Host document, watermark, and verifier’s public key.

**Output:** The stego document (watermarked document).

- **E1.** Compute the black and the white patterns of the host document, by using the ZS algorithm.
- **E2.** Combine the black and the white patterns to form a single pattern.
- **E3.** Generate the hash code $I_h$ of the single pattern using a one-way hash function.
- **E4.** Conduct an exclusive-OR operation on the hash code and the watermark.
- **E5.** Encrypt the result of the E4 using the verifier’s public key to obtain the result $I_e$, where
  $$I_e = E_{K_p}(W_p \oplus I_h).$$
- **E6.** Divide the host document into blocks, and then calculate the odd/even value and choose the best embedding locations.
- **E7.** Embed the information $I_e$ in the best embedding blocks.
- **E8.** Output the stego document.

Note that here we define the odd/even value for a block. If the number of black pixel is odd, then we embed 1. If the number of black pixel is even, then we embed 0. All-black or all-white blocks would not be used, because changing a pixel in such a block is visually detectable.

As an example, Figure 3 indicates an odd block and 1 is embedded. If we want to embed 0, we need to change the pixel P from white to black.

Besides, a pixel which is selected as an embedding location should not change the black or the white patterns. As an example, pixel A in Figure 4 should be filtered out in the embedding process. The reason is that if pixel A is changed from black to white, then the $X(p) = 2$ and the black patterns would be altered. So pixel A should not be selected as an embedding location. As for pixel B, the change of pixel B from black to white will not change the black pattern.
patterns, thus B could be selected as an embedding location.

3.3. Authentication Procedure

In this section, we have the authentication process of our proposed scheme. Some parameters are used:

- $W_B$: Extraction watermark
- $I_B^r$: Hash code
- $K_x$: Private Key
- $D()$: Decrypt function
- $I_E^r$: Encrypted data

The verification process is as follows.

**Input:** The stego document and the private key of the verifier.

**Output:** The watermark.

1. Extract the black and the white patterns by ZS algorithm.
2. Combine the black and the white patterns to form a single pattern.
3. Hash the single pattern to produce a sequential hash data $I_H^r$.
4. Divide the stego document into blocks and calculate the embedding locations.
5. Extract the embedded data.
6. Decrypt the above data using the corresponding private key to obtain $I_E^r$.
7. Perform an exclusive-OR operation on the encrypted data $I_E^r$ and the hash code $I_H^r$. Then the watermark is revealed.

On revealing the watermark, we can check visually the integrity of the input document.

4. Experimental Results

In this section, we describe the experimental results of our proposed scheme. First we have our system environment. We use Windows XP operation system, JBuilderX IDE, Java Advanced Imaging package (JAI) library, and the language is JAVA jdk 1.6.0. The attack tool is PhotoShop 8.0.

From the experiments, we see that our proposal can protect the embedded image effectively. The attacker cannot obtain the embedded information. Further, a little changes made to the document would be detected by our scheme.

To ensure that the proposed scheme is promising, we conduct a series of experiments. As an example, we show some of the experimental results as follows. Figure 5 shows a host document image. Figure 6 shows the extracted black and white patterns. Figure 7 shows the watermark to be embedded. Figure 8 is the stego document. (or watermarked document)

For attacking experiment, in Figure 10 we first do some changes to the stego document, e.g., add a noise point to the character “k”. In the verification stage, we found that the extracted watermark has been blurred seriously, shown in Figure 11. Further, Figure 12 and Figure 14 show changes of a character, respectively. Figure 13 and Figure 15 show the result of deleting some characters.

We also do experiments on non-textual documents. We transform Lena to binary image, as shown in Figure 16. Figure 17 is the stego image. Figure 18 shows the changes made on the area of the eyes. We see the extracted watermark has been blurred seriously from Figure 19.
5. Conclusion

In this paper, we propose a novel method which can protect the integrity for binary documents. By extracting the black and the white patterns and using watermark technique, a document can be protected effectively. From the experiments, we can see that our proposed scheme is promising for binary documents. Even a little change made to a document could be detected visually from the extracted watermark.

Since the Zhang-Suen thinning algorithm is parallel, it can be implemented fast. Therefore, our proposed scheme is efficient. Further, the block size affects the capacity of embedding information in implementation. If the block size is large, the amount of the black and the white patterns would be less. On the contrary, if the block size is too small, we would have large amount of patterns. We should have proper amount of patterns such that not to lose the important patterns and to have enough capacity for the embedding information.

Acknowledgement

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC-95-2218-E-001-001, NSC-95-2218-E-011-015 and NSC95-2221-E-029-020-MY3.

References