Journal of Mobile, Embedded and Distributed Systems, vol. II, no. 1, 2010
ISSN 2067 – 4074

Image Processing Oriented to Security Optimization

Ion IVAN, Adrian VISOIU, Mihai DOINEA

Economic Informatics Department, Academy of Economic Studies
Pta. Romana 6, sector 1, Bucharest, ROMANIA
ionivan@ase.ro, adrian.visoiu@csie.ase.ro, mihai.doinea@ie.ase.ro

Abstract: This paper presents the main aspects of the digital content security. It describes the content of watermarking, presenting the steganography concept. SteganoGraphy application is presented and the algorithm used is analyzed. Optimization techniques are introduces to minimize the risk of discovering the information embedded into digital content by means of invisible watermarking. Techniques of analyzing the digital content results and identify the possible countermeasures for optimizing the steganography algorithm are presented.

Keywords: security, watermarking, digital content, image, steganography, algorithm, image filters.

1. Digital content security

Digital content security is a concept which is trying to protect and conserve all the aspects related to the characteristics of digital content. The characteristics of digital content are:

- authenticity – the characteristic that demonstrate that digital content isn't a fake;
- integrity – proves that an original digital content wasn't changed without authorization;
- non-repudiation – the bound between the digital content and his master;
- confidentiality – the way through which a digital content is protected against unauthorized access;

The security characteristics are reflected onto the digital content giving new ways of creating security methods and techniques for assuring them.

A way of protecting the digital content is by doing a content management without too much flaws. Content management is described as a set of processes and technologies that manage the life cycle of digital information. For a high level of security content management must be able to manage digital content distributions and digital rights.

The future of digital content security lies in the power of binding uniquely the owner of a material with the product in cause assuring non-repudiation characteristic and in the same time denying any unauthorized access, assuring confidentiality.

For example, if someone tries to buy an interpret album from a online music store and listen it to a device, the price paid for this service should give the person entitled the right to listen it but not the right to multiple it and distribute it without any prior permissions.

Watermarking had reduced in a certain case the digital content theft by engraving additional information such as serial numbers, copyrights messages into digital content without altering it. The digital signal can be, that way, passed as long with the copyright information without the possibility of removing it to any user.

An application of invisible watermarking is steganography, a method through which messages can be hidden in multimedia content by modifying redundant information without the possibility of discovering it unless special analyze techniques are used.

Even so information can be encrypted using common algorithms and then embedded into digital content.

The term of steganography is very old, dating from 1499, recorded by Johannes Trithemius in his Steganographia, a treatise on cryptography and steganography disguised as a book on magic but the use of it is even older dating from 440 BC when Herodotus mentioned two examples of it in *The Histories of Herodotus*:

This is a post conference paper. Parts of this paper have been published in the Proceedings of the SECITC 2009 Conference (printed version).

- Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface. Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand;
- Histiaeus shaved the head of his most trusted slave and tattooed a message on it. After his hair had grown the message was hidden. The purpose was to instigate a revolt against the Persians.

The word steganography is of Greek origin meaning *concealed writing*. That is what the SteganoGraphy application intend to do, concealing messages inside of a image. The multimedia objects used for concealing messages are bitmap files. Bitmap files are stored in a device – independent bitmap, so called DIB format which allows the operating system to display the bitmap on any type of display device.

2. Watermarking

Watermarking is the process of embedding information into digital signals like video, audio and pictures. The embedded information is intended to be transmitted or copied along with the signal. Watermarking is necessary in order to identify the owner of the original data. For digital images, this is the case of visible watermarking, where logos are superimposed or blended with an original image in order to keep visible the owner whenever the image is copied.

When using invisible watermarking information is added to the original data but this information is not directly perceivable. An application of this type of watermarking is copyright protection. If the data is copied illegally, then the owner may identify its property by detecting and decoding the hidden information.

Invisible watermarking algorithms have peculiar characteristics that give the strength of the algorithm:

- the embedded message is difficult to perceive by a human observer;
- the embedded message is difficult to remove; the image may be subject to processing like geometric distortion, resizing, data loss compression, enhancing, adjustments; in such cases, the message should be still recognizable;
- the algorithm is applicable to all multimedia types;
- by decoding the embedded image the owner is identified accurately.

Steganography is an application of watermarking for exchanging messages between two parties. An algorithm embeds the message into the exchanged data using watermarking principles. An important application is checking the authenticity of an image. This application requires two parties, a sender and a recipient. They share a secret key, which is used by the sender to encode a hidden message inside a image file. The recipient uses the key to extract the message. If the message does not match a pattern or is undistinguishable or is altered, then the sender is not authentic and the file is not accepted as valid.

In the presented application of steganography, fragile watermarking is needed. Fragile watermarking refers to a message embedding that is easily altered when the image suffers changes. If the decoded message is altered, the recipient becomes aware that the image has been modified from its initial state and it is not the version issued from the authentic sender. This is the main difference from basic watermarking, as there is a need to observe changes in the exchanged information.

3. Encryption software and optimization

For understanding the way that application encrypts a message we first must understand the structure of a bitmap file which consists in a 3 or 4 parts depending of the way that color information is presented:

- BITMAPHEADER – contains various information about the header of the file such as: the signature of the file which is *BM*; file size, a reserved 4 bytes zone and the offset to which image representation begins stored on 4 bytes; bitmap header size is 14 bytes;

ISSN 2067 – 4074

- BITMAPINFOHEADER – with a total of 40 bytes used presents the size of the info header zone, value 28 in hexadecimal; image dimensions represented by height and width; number of planes; the color depth given by the number of bits per pixel; compression, if there is any; total image size if compression was specified; horizontal and vertical resolution stored on 8 bytes both; number of colors used; the number of important colors;
- OPTIONAL PALLETE – represented by the color table, is present only if the number of bits per pixel is less or equal to 8;
- IMAGE DATA – the actual zone where useful information is stored.

In figure 1 are presented the first 54 bytes which are selected, meaning the information of both bitmap header and bitmap info header from a bitmap file with the following attributes:

- signature – 2 bytes, value *BM* – ASCI and 4D42h;
- file size – 4 bytes, value 0010B476h meaning 1.094.774 bytes;
- 4 bytes reserved;
- data offset – 4 bytes, value 36h meaning that the actual image data will begin after 54 bytes at the end of the selected zone;
- info header size – 4 bytes, value 28h equivalent of 40 bytes;
- width – 4 bytes, value 26Eh meaning 622 pixels;
- height – 4 bytes, value 1B8h meaning 440 pixels;
- and other useful information.

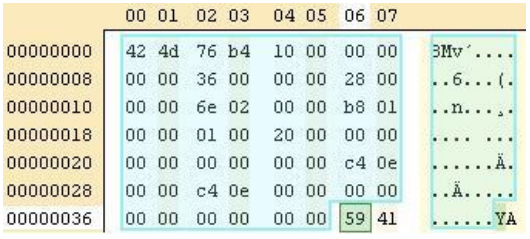

Fig. 1 Hexadecimal view of a bitmap

For hiding a message in a bitmap file, a password is needed representing the actual step at which parts of the message will be stored by modifying the pixels found in the image.

The process of encrypting a message is presented in the following algorithm:

1. message length calculation;
2. storing the length of the message, only 2^{24} bytes meaning that the message is limited by the maximum value stored in 3 bytes, representing the RGB colors;
3. total number of pixels count, width*height – 1;
4. calculating the X and Y coordinates based on the value of the ASCI code of every byte read from the password code and the dispersion component calculated based on the length of both message and password;
5. writing every byte of the message stream along with a byte of the password code read in reverse order at calculated positions.

The application developed that implements this algorithm is briefly presented in the following pictures.

In figure 2, is represented the information to hide, *HIDDEN MESSAGE* with the key password *PASS*. The picture in which the encryption will take place is a BMP image of 100x100 pixels, 32 bit representation.

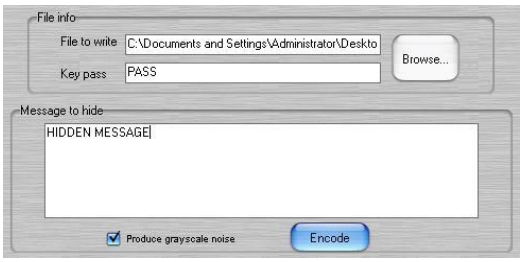

Fig. 2 Hidden information

The process of hiding the message will take every byte of the message and write it to the image as follows:

- the dispersion calculated for a stream length of 28, 2 bytes for every character, and a length of the key password of 8 bytes, is 8, meaning that every step at which a byte of the message will be written will be multiplied with the dispersion value;
- the first character of the message, *H*, ASCI code 72, will be written after the first 54 bytes along with S, the first character in reverse order of the key password, at an offset of 80*8, where 80 is the ASCII value of the first character in the key password,

P and 8 is the dispersion value with the results in figure 3.

	00	01	02	03	04	05	06	07	
0000922a	ff	ff	48	48	48	ff	53	53	ÿÿHHHÿSS
00009230	53	ff	ff	ff	ff	ff	ff	ff	Sÿÿÿÿÿÿÿ
00009238	ff	ff	ff	ff	ff	ff	ff	ff	ÿÿÿÿÿÿÿÿ

Fig. 3. First character encryption

For testing a white image of 100x100 pixels was chose and the results of the encryption process are presented in figure 4.

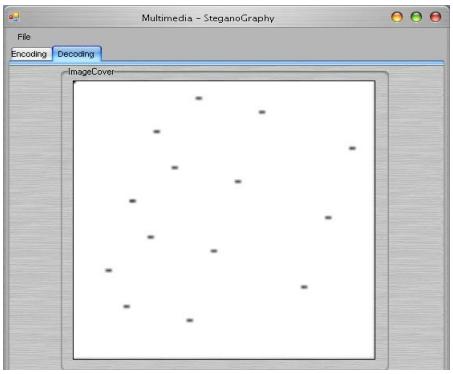

Fig. 4 Encryption results

The result of the process of decrypting the image is presented in figure 5, the decryption algorithm being the reverse process of the encryption one.

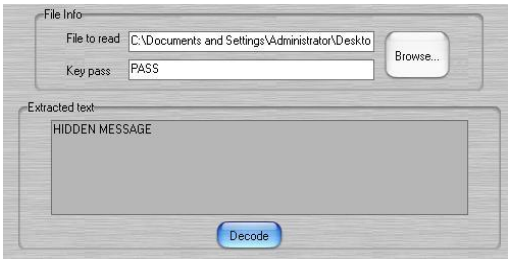

Fig. 5 Decryption result.

For decrypting the message that was embedded into the image, it is first being calculated the message length and then based on the password code the dispersion will help to read every single byte of the message encoded.

4. Detecting techniques and improvements

The proposed algorithm is fit for steganography with fragile watermarking for checking sender authenticity.

The proposed algorithm makes the embedded message to cause little changes to the original image and is difficult to perceive. A human observer cannot easy tell the difference between two images, an original image, and a image with embedded message. Figure 6 presents two images. First image is the original image. The second image is embedded with the message "Academia de Studii Economice Bucuresti", using the key "ASE".

Fig. 6 Original and watermarked image

As observed in figure 6, the visual impact of the watermark is maximum on areas of uniform color where foreign pixels are visible. The textured areas hide effectively the pixels carrying the embedded message.

It is recommended to use the algorithm with highly textured and complex images.

It has been shown that in order to identify changes in the image, the watermark must be fragile. The proposed algorithm works with color BMP image format. This is a lossless storage format that records all the pixels in the image. The embedded message is altered when the image is subject to common transformations: resizing, sharpening, blurring, noise reduction, format change with a lousy compression algorithm, painting.

The algorithm makes the embedded message difficult to identify. The best case to identify the message is when the attacker has both the original image and the watermarked one. Applying the difference operator between the two images yields the difference image. The difference image shows all the pixels that changed in the watermarking process. However, the key used for embedding distributes the message over the entire image, making difficult to reconstruct the

ISSN 2067 – 4074

hidden information. When using the proposed algorithm, it is recommended to use pictures that are not publicly available, in order to reduce the probability of finding an unaltered image, an attacker would use to extract hidden information.
Figure 7 shows the difference image obtained from the two images above and its Fourier transform.

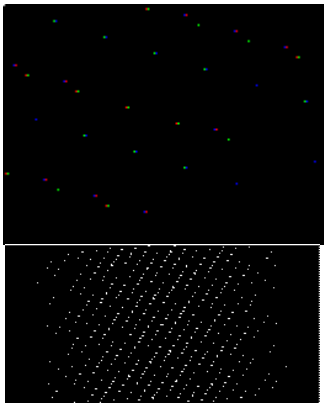

Fig. 7 Difference image and its Fourier transform

Figure 7 shows that information is embedded in the image using steps derived from the key password. The changed pixels in the watermarked image form a line pattern. The Fourier transform also confirms the existence of a pattern in the difference image. However, the message is not easy distinguishable when comparing the Fourier transforms of the original and watermarked real images, as shown in Figure 8.

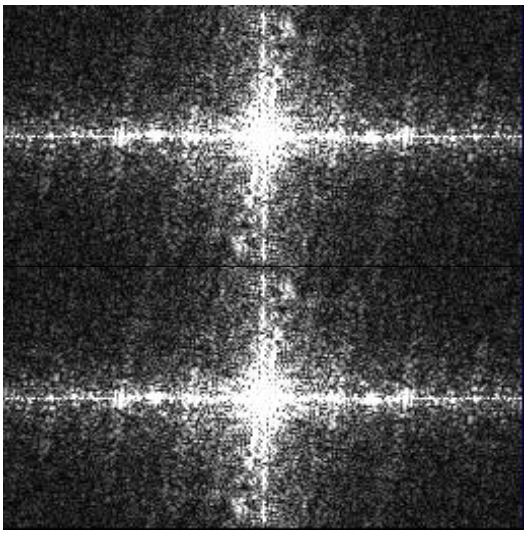

Fig. 8 Fourier transforms for the original and watermarked images

In figure 8, it is observed that watermark patterns are hard to identify. Again, the proposed method shows a good performance when used on complex images. The same time, other tests also showed that the message is hard to separate when the picture contains salt and pepper noise. An improvement of the algorithm would be a more random distribution of the information in the image.
The algorithm gives the possibility to the recipient of the watermarked image to identify the sender. The key used to embed the information is shared between the two parties. The recipient uses the key against the watermarked image and retrieves the hidden message correctly. If the message is partially or totally affected by changes, there is a proof that a third party altered it, and the authenticity is not validated.
The algorithm is adaptable in terms of image color type. The watermark consists of colored pixels, when using color images and is grayscale when using grayscale images.
The algorithm has built in limits for hidden message length. If the message is too long, the large number of changed pixels in the watermarked image affects image quality and betrays the presence of the watermark.
Assuming that the length of the key is smaller than the length of the message, the number of needed pixels is determined by the formula:

$$RP = \left(\sum_{i=1}^{length(key)} key_i \right) * \frac{length(msg)}{length(key)}$$

where:

RP – the number of required pixels;
key – the key used to embed the message as byte stream;
msg – the message as byte stream.

This includes the length of the message and the space left between pixels carrying message information. Message pixels are inserted with a dispersion step given by:

$$ST = \frac{CP}{RP},$$

where:

ST – dispersion step for inserting message pixels;
CP – the total number of pixels in the image;
RP – the required number of pixels to encode the image.

The number of modified pixels in the watermarked image, MP, is proportional to the length of the message as the image records extra information regarding the message length. When changing individual image pixels in order to alter the image, the probability of changing a message pixel is:

$$PC = \frac{MP}{CP},$$

where:

PC – the probability of changing a message pixel;
MP – the number of modified pixels;
CP – the number of pixels in the image.

If changes are applied to individual pixels for *n* times, the probability of affecting a message pixel TPC is:

$$TPC(n) = n * PC$$

It is obvious that applying image filters that affect the whole image, thus MP=CP, conduct to message alteration.
When applying changes locally, there are small windows defined by message pixels where the image is altered without changing the message. Assuming that message pixels are equally distributed across the image, they are likely to be found every 1/PC pixels. The area determined by two adjacent message pixels is likely to have sqrt(1/PC) pixels in length and sqrt(1/PC) pixels width. The smaller the modifiable window area is, the more fragile is the message to discrete image changes, which is desired in the application of the algorithm.

When identifying alterations in the decoded message, and the message still has understandable parts, the process of watermarking the image is redone and the recipient observes what parts of the image have been modified, as message elements have corresponding pixels in the image. Correlated with the content of the image, this offers valuable information about the purpose of the attack.
The algorithm is simple to implement, fast and reliable. It has been implemented in C# language, the main algorithm class having 120 statements. It is easy to port the algorithm to other platforms.

5. Conclusions

Important progresses in the field of digital content security were made, but the same has happened with the methods and techniques of attackers. For the safety of digital content, security must face the whole attackers' arsenal, who will try to make unauthorized copies. Cryptography will remain the most valuable asset of security using encryption keys increasingly powerful. Digital content quality is directly affected by the security measures applied for copyright purposes. For preserving the digital content quality characteristics, light protective algorithm must be applied but this goes to a exclusive paradigm that of the efficiency of those algorithms.
The future of digital content is assured, that's for sure, though the aggressiveness of copyright faults of IT pirates is widely spread.

References

[1] Ingemar J. COX, Matthew MILLER, Jeffrey BLOOM, Jessica FRIDRICH, Ton KALKER – Digital watermarking and steganography, Printing House Morgan Kaufmann, 2008, ISBN 0123725852, 593 pg.

[2] Gregory KIPPER – Investigator's guide to steganography, Printing House CRC Press, 2004, ISBN 0849324335

ISSN 2067 – 4074

[3] Peter WAYNER – Disappearing Cryptography: Information Hiding: Steganography & Watermarking, Printing House Morgan Kaufmann, 2008, ISBN 0123744792.

[4] (2009, Sept.) Wikipedia Site. [Online] www.wikipedia.com

[5] Victor Valeriu PATRICIU, Ion BICA, Monica ENE-PIETROSEANU, Justin PRIESCU – Semnaturi electronice si securitate informatica, BIC All Printing House, 2006, ISBN 973-571-564-3.