Lattice approach to video steganography

Onur ÜNLÜ*, Hacer KARACAN
Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey

Abstract: As a result of the continuously increasing amount of important data, the demand for information security increases daily. Steganographic techniques can play a considerable role in solving this issue, since they can hide the necessary data. Generally, audio and image files have been utilized as hiding media in this area. However, because of their limited capacity, these media limit the amount of hidden information. On the other hand, the capacity limit can be exceeded by applying steganography on video files. There are few studies on video steganography in the literature. In this study, a steganographic library, which is independent of medium and method, was developed for hiding information in audio, image, and video files by using various methods. Furthermore, distinctly from other studies, the lattice approach was proposed for this issue. In this approach, security was increased by using the image and audio domain of video together in the hiding process. In addition, we were able to hide the real-time data in a video taken from a webcam. Thus, the size of the produced video file could be set, according to the size of the secret file, with this model. Meanwhile, the possibility of detecting the existence of steganography by comparing to the original video was limited. The success of the hiding process was also evaluated with various steganalyses and noise measurement methods, and the results are presented and discussed in the relevant sections.

Key words: Video steganography, steganographic library

1. Introduction

Nowadays, the rapid increase in information-sharing causes various security problems. New security gaps are emerging daily. Two main ways of providing security in information communication are cryptology and steganography.

Cryptology is based on the rule of encrypting the information to be sent and decrypting the information on the receiver side. In order to obtain the information, third parties that are listening to the communication media must break the password. Here, it is important is to prevent the content of the communication from being obtained by third parties or software programs that are not participating in the communication itself. Since cryptologic systems gain their power from the encryption algorithm and secrecy, they can be hacked when the existence of the algorithm is realized.

Steganography is a technique of hiding a meaningful group of data (implicit/secret object) into another meaningful group of data (explicit/cover object), which is larger in size. It is based on sending the confidential/secret information by hiding it in an object that seems innocent to third parties. Thus, it is important not to make anything appear suspicious to third parties or software. The system is hacked when it is understood that the file is carrying secret information.

*Correspondence: unlu0@yahoo.com
In the real world, burying treasure at a specific place and then finding and removing it can be a good analogy for steganography. In a virtual environment, the implicit/secret object can be a group of meaningful bits, and the explicit/cover object can be any appropriate file type.

Steganography is the science of secret communication. Etymologically the term ‘steganography’ comes from the Greek word for ‘concealed text’ [1]. The basic idea is to create the stego object by embedding the important information into an ordinary object (text, image, sound, or video). The stego object is sent/transmitted and stored in the same way as a cover object. For uninformed observers, stego objects display the content of the cover object. From the target user’s viewpoint, the stego object carries the important information hidden by the cover object. Generally, it is a three-phase process. Secret information is buried into the cover object by using an information-hiding algorithm, and in this way the stego object is created and is sent to the receiver via communication media. Then the receiver acquires the secret information from the stego object using a data extraction algorithm. This process is displayed in Figure 1.

![Steganography Operation Logic](image)

**Figure 1.** Operation logic of steganography.

Perceptibility, robustness, and capacity factors can be considered as the success criteria for steganographic procedures. Generally, these criteria adversely affect each other and are therefore known as the ‘magic triangle’ (Figure 2) [2]. Perceptibility is used as a measure of noticeability of the secret transmission in the medium by third parties intuitively or by other means. Robustness is used as a measure of acquisition of secret information after performing operations that cause changes in the object, such as cropping or filtering. Capacity is used as a measure of the maximum size that can be stored in the media. The relation between these 3 factors was previously examined in sound files [3]. Among these factors, perceptibility and capacity were mostly considered in the studies of the field.

![Magic Triangle](image)

**Figure 2.** Magic triangle.
2. Steganography in the literature

In the literature, studies on steganography can be grouped into 4 main types as text, sound, image, and video.

2.1. Text steganography

Text steganography is one of the oldest steganography methods. It is the most difficult area for the application of the steganography [4]. This is due to the lack of repeating unnecessary data compared to media that use sound and video. This method was successfully applied previously in different languages such as Thai, English, Japanese, Korean, Chinese, Persian, Arabic, and Turkish [5,6].

In their study, Shirali-Shahreza and Shirali-Shahreza [4] proposed a method where the vertical height of these letters is used for hiding with the help of letters in Arabic and Persian that have punctuation marks. In another study, Shahreza [7] performed a study that stored information by changing words that were different in US and UK English, yet held the same meaning. In a further study, Chinese was mentioned as using simplified and conventional characters interchangeably, and this property was identified as a possibility for steganographic purposes. In addition, the usage of conventional characters for ones and simplified characters for zeros was suggested for users [8].

2.2. Sound steganography

Researchers in Pakistan proposed an advanced type of the least significant bit (LSB) method [9]. With this method, 8 bits of sound samples are used in the hiding process, and sample and data hiding bits are selected according to a map (see Table 1). This gives the impression that the information hiding is done randomly, which increases security against statistical attacks.

Table 1. Secret bit selection map for a sound sample [9]. If the first 2 bits of the sound sample are 1, the secret data will be hidden in the 2nd LSB of the sound sample.

<table>
<thead>
<tr>
<th>First 2 bits</th>
<th>Secret bit index</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>3. LSB</td>
</tr>
<tr>
<td>01</td>
<td>2. LSB</td>
</tr>
<tr>
<td>01/11</td>
<td>1. LSB</td>
</tr>
</tbody>
</table>

Spontaneity in bit selection is provided by choosing the first 2 bits of the sound sample. Table 1 provides the conditions about which index to use. A similar method is used when selecting the sound sample. The first 3 bits of a sound sample indicate a subsequent sound sample that contains the secret. Average capacity is 1 bit for 4 sound samples. In their study, Yan and Wang performed steganography with the selection of a table in Huffman coding, which is a property of the MP3 compression standard [10]. Compared to other methods, this provides high capacity and also ensures security and low deformity requirements. Furthermore, Shirali-Shahreza proposed a new method that performs real time hiding and has robustness to the MP3 compression by changing the length of the silent period of speech sounds [11].

2.3. Image steganography

The LSB method is used for image steganography in most studies [12]. With this method, the bit that constitutes one pixel of the image and has the least effect on the color is changed. Thus, we obtain the area where we can nest the secret information (see Table 2 for examples).
Table 2. Changing of last bits during LSB hiding operations.

<table>
<thead>
<tr>
<th>Value before hiding</th>
<th>Secret bit</th>
<th>Value after hiding</th>
</tr>
</thead>
<tbody>
<tr>
<td>10111111 (191)</td>
<td>1</td>
<td>10111111 (191)</td>
</tr>
<tr>
<td>11110011 (231)</td>
<td>1</td>
<td>11110011 (231)</td>
</tr>
<tr>
<td>10000111 (135)</td>
<td>0</td>
<td>10000110 (134)</td>
</tr>
</tbody>
</table>

Westfeld and Pfitzmaan presented statistical and visual attack methods that are determined when the standard LSB method is used for hiding [13]. This attack method zeroizes the first 7 bits of the eight-bit color value, thus turning the color values of the image into zeros and ones. Then the ones are converted to 255, which is the highest color value, and the zeroes are left as they are. This method can be appropriate for images using simple LSB; however, when a complex image is chosen, it does not work. Images that have black and white values depending on the LSBs are given in Figure 3. It can be seen in the figure that some information is hidden in half of the second one.

When changes are performed on the last bits using the LSB method, there will be unwanted effects on the image. The ratio between the ones and zeros of the last bits is high in the original image; however, it will be lower in the images carrying hidden information. The attack using this statistical difference is called a chi-square attack (Figure 4).

![Figure 3. Example of a visual attack [13].](image)

![Figure 4. Chi-square attack [13].](image)
In order to get more robust images against the chi-square attack, a pseudorandom number generator (PRNG) can be used for the selection of a pixel that is used for hiding. The disadvantage of the PRNG is that it requires preliminary communication. In addition, if a PRNG is used, the size of the information to be hidden should be kept small for security purposes.

The least significant bit modulation (LSBM) method, which was first proposed by Sharp, differs from the LSB method in terms of changing method of the last bit [14]. In this method, the last bit of the cover is not changed with the hide bit. If the last bit of the cover is different from the hide bit, hiding is achieved by increasing or decreasing by 1. When this method is used in the frequency area, the stego image becomes more robust against statistical attacks.

In 2006, Mielikainen improved this method and was able to hide the same amount of information by changing less pixels. In the proposed method, pixels were used as pairs. One bit of information was hidden into the last bit of the first pixel, and it obtained the second bit of information from a function of the pixel pair. In this way, the cover image became more resistant to attacks [15].

Researchers from China developed the method proposed by Mielikainen in their study. They stated that the plain and smooth areas in the cover image would be visually weak and would provide low security for information hiding. In order to prevent this, they proposed a method that can choose hiding areas and perform hiding according to the size of the information and the difference between 2 successive pixels. When the information to be hidden is small, the method leaves the plain areas of the cover image as is and performs hiding on the areas where the sharpness is dense, thus increasing security [16].

It can be seen from Figure 5 that the method does not hide plain areas at low hiding ratios. With an increase in the hiding ratio, the method uses plain areas, and this decreases security.

![Figure 5](image-url)

**Figure 5.** a) Original image, b) 10% hidden data, c) 20% hidden data, d) 30% hidden data, e) 40% hidden data, and f) 50% hidden data [16].

Researchers from King Saud University proposed a different method, in which images are split into n equal blocks (i.e. splitting a 100 × 100 image into four 50 × 50 images) and LSB is applied spirally. They
proposed 3 different applications for this method: from the outer corners of the blocks, from the center of the blocks, or using both by applying them clockwise or counterclockwise. Results showed that this method is considerably resistant to the chi-square attack [17].

In another study, a method that performs hiding by weighing RGB values was proposed [18]. With this method, high hiding capacity was achieved with low visual corruption. Since this method depends on changing the last digit instead of the last bit, it is known as the least significant digit (LSD), which is a derivation from LSB. Although the last digits of the binary system are used in the LSB method, the last digit is used for hiding in the decimal system. The number is increased by 10, and the digit to be hidden is subtracted. There are some cases where exceptional outcomes arise. If the original G and B values are between 250 and 255, the algorithm produces invalid results. In order to overcome this problem, each G and B value is reduced by 10 [19]. With this hiding method more information can be hidden, whereas the noticeability of the hiding process increases. Although the maximum difference between the numbers before and after hiding is 9, when the binary system is used, it is seen that the upper bits can be changed (Table 3). This situation results in serious corruption of an image when the application that provides the image performs the calculation based on the bit values.

Table 3. Change of lower bits during LSD hiding operation [18,19].

<table>
<thead>
<tr>
<th>Value before hiding</th>
<th>Secret value</th>
<th>Value after hiding</th>
</tr>
</thead>
<tbody>
<tr>
<td>10111111 (191)</td>
<td>1</td>
<td>11000111 (190 + 10 – 1 = 199)</td>
</tr>
<tr>
<td>11110011 (231)</td>
<td>4</td>
<td>11111000 (230 + 10 – 4 = 236)</td>
</tr>
<tr>
<td>10000111 (135)</td>
<td>7</td>
<td>10000101 (130 + 10 – 7 = 133)</td>
</tr>
</tbody>
</table>

Wang and Chen proposed a two-way block-matching method that is based on block-matching of the cover image and hidden image [20]. Two images are split into blocks of m × n. The highest similarity block to the ith block in the cover image is found and matched. This process is repeated continuously. After the process, the index information of the well-matched blocks and unmatched blocks is compressed and stored in LSB to be distributed to the cover image. The results of the experiment indicated that it provides low noise and high capacity [20]. In another study, the present block-matching method was presented as expanding the block search field, and thus lowered the amount of information to be hidden in the block. In this way, a better quality cover was claimed to be achieved on the hidden image than in the previous study [21]. Another method, spread-spectrum image steganography (SSIS), was also presented for hiding the secret information inside the noise [22]. In order to retrieve the secret information successfully, restoration techniques were used. If the original image were to be achieved by restoration, the added noise would also be achieved. With these techniques, since the information cannot be achieved without errors, error control-coding was applied before hiding. In addition, it was seen that the hidden information could endure the compressed image with well-chosen parameters [22].

Shirali-Shahreza described another steganographic method that he developed for multimedia messaging services [23]. MMS messages can transmit text up to 30 KB, images up to 100 KB, and videos up to 300 KB. With the proposed method, the text and image fields in the message were used together. Using the eye’s lesser sensitivity to blue, the system was utilized with the images and abbreviations in the text field of messages (for/4, you/u) [23].

Methods using the frequency field generally use discrete cosine transformation (DCT). JPEGs are the image files where DCT steganography is mainly used. In JPEG files, LSB is utilized after DCT, and quantization
and rounding are applied to $8 \times 8$ image blocks. Steps of this method are shown in Figure 6. Since all blocks will be affected by a change in a value, choosing the values where the information will be hidden is very important in this method. J-Steg software that uses LSB on these coefficients was presented in 1997 (http://www.jsteg.org). Although the method was resistant to visual attacks, it was weak against statistical attacks. In their study, Westfeld and Pfitzmann detected the hidings performed by J-Steg software by using the chi-square attack technique [13].

Westfeld developed an F5 algorithm that is resistant to visual and statistical attacks [24]. With this algorithm, it was ensured that the histograms of the DCT coefficients were not corrupted and that the information was hidden into DCT coefficients in a distributed manner. One year later, Fridrich et al. detected the information that was hidden by F5 [25]. The key point in this method is the estimation of the histogram of the image before hiding. This can be achieved by cutting 4 pixels on both sides and compressing them again to remove quantization in the frequency field of the image. An attack is performed by comparing the histograms of 2 images [24]. In addition, 2 new methods, namely quantization index modulation (QIM) and histogram matching (HM), were proposed to allow more information-hiding capability than F5 [26]. These methods removed the histogram difference that occurred as a result of the attack against F5.

Moreover, there are several methods that can cheat the chi-square attack. One of these methods is based on sequential binary grouping of the DCT coefficient values (i.e. 2-3, 4-5) [27]. When 2 numbers belonging to a group are found, one-bit hiding can be performed. This reduces the hiding capacity by half. Hiding is performed in such a way that the numbers will not be outside the group, and this way a change in any group will not affect the others.

2.4. Video steganography

The capacity of sound files and motionless images is limited. This capacity constraint also restricts the amount of information that can be hidden. Anyone that attempts to hide a folder in the file system cannot perform this for most of the folders. This is because the folder to be hidden exceeds the hiding capacity of the cover image. In order to overcome this capacity limitation, video files are used.
Video files are similar to image files, yet with several important differences. First, there are more pixels in video files. This also increases the hiding capacity of the videos. In addition, changes in the videos of small differences cause the repetition of unnecessary information. Compressing methods that are used to prevent this unnecessary information render the hiding difficult.

In a study, a high rate of information-hiding was implemented using AVI video format [28]. This method made use of the blue sensitivity of the human eye and achieved information hiding by 1/3 by hiding information to 3 bits to red and green fields and 2 bits to blue field of one pixel. Additionally, the researchers reported that crowded videos are more suitable for hiding, since the human eye is more sensitive to the changes in wide and flat areas, such as sky and sea, than to the changes in mixed areas that include too many objects [28].

In another study, Çetin and Özcerit proposed a new method that uses the color histogram of the environment when choosing pixels to be hidden [29]. According to this method, the histograms of the color components of the frames are calculated, and a threshold value is determined by comparing the histogram values of the successive frames. This threshold value is used in pixel selection, and the method is applied as both frame-based and block-based. The block-based method was successful in terms of perception. In another study, only type “I” frame of MPEG files was used. During the generation of the “I” frame, it is recommended to perform hiding into coefficients above certain threshold values, following DCT [30]. In addition, in their proposed algorithm, researchers from China hid the hiding capacity of GOP and the data holding the starting point of the hidden information, referred to as control data in the “I” frame. Secret information was hidden in type “P” and “B” frames. Provided that the change in the motion vector was above a certain threshold value, the last bit of the vector was used for hiding [31]. In another study, type “I” frames were not used because of their vulnerability to steganalysis, and the researcher used type “P” and “B” frames for hiding instead [32]. Unlike other studies, when selecting the vectors for hiding, associated macroblock prediction error was used instead of vector magnitude and the angular properties of the vector [32].

3. Developed application

The aim of the developed software is to create a flexible library and an interface for the direct use of this software for steganographic applications. The library supports the LSB, LSBM, and LSD methods that were described in Section 2.3. It also supports raw media environments (WAV, BMP, and AVI) that do not use compressions. Figure 7 shows the data-hiding interface of the Stegofolderlock application.

![Figure 7. Stego folder lock data-hiding interface.](image)
For platform independency, the application has been developed by Java programming language. Since the application will be platform- and method-independent, flexible design was an important requirement. Therefore, various design patterns, such as bridge, facade, composite, delegation, and factory, were applied. Figure 8 shows a general class diagram of the developed steganographic library.

![Diagram](image.png)

**Figure 8.** System general class diagram.

### 4. Experiments

Input and output images of information hiding on a BMP image, using standard LSB, LSBM, and LSD, are given in Figure 9. As seen in the figures, there is no noticeable difference between the first 3 images. On the contrary, there is a noticeable difference in the fourth image. This is because the LSD method changes the last 3 bits of one byte densely, and the aircraft image is an image that does not show gray color variations. The second row of the experimental image shows the result of the visual attack against the last bit plane, the third row shows the visual attack against the second to last bit plane, and the fourth row shows the attack against the antepenultimate bit plane. Hidden data can be detected as a result of applied steganalysis on the second, third, and fourth images. The second row shows that there is a big difference between the last bit plane of the original image and the last bit planes of stego images, where data were hidden with the LSB, LSBM, and LSD methods. This difference shows that those bits were changed; in other words, there are hidden data. At this point, we should mention that the hidden data for all these methods were the same. Whereas the LSB and LSBM methods hide a single bit into one byte, the LSD method can hide a single byte into 3 bytes. Because of this capacity difference, the change in the scan area of LSD method is one-third that of the other 2 methods. The LSD method changes only the related bit of a byte. The LSBM method has the same effect as the LSB method on the last bit plane. The LSBM method is different from the LSB method by incrementing or decrementing the number by one. Even though this process changes the value of the number by one, there is the possibility of change on the upper bits of the byte. It is seen that the LSBM method changes the penultimate and antepenultimate bits of the byte in the third row of Figure 9. Similarly, the LSD method has the possibility of change on the upper bits of the byte even if it changes the value of the number up to 9.

As shown in Figure 10, when statistical attacks are performed against the aircraft images, it becomes clearer that steganography was applied in all 3 methods. It is observed that the “0” and “1” balance is equal in the gray marked area of the last part. As a result of this attack, it can be said that there are hidden data in the cover image.
There seems to be no difference between LSB and LSBM without PRNG against visual and statistical attacks. In the LSBM method, the change possibility of the penultimate bit, in addition to the last bit, is 50%. This kind of change may lead to a change in the upper bits, where it does not affect the image files, but causes sensible distortion to audio files.
PSNR, mean squared error (MSE), structural similarity (SSIM) [33], and universal image quality index (UQI) [34] values are compared in Table 4. It is seen that both LSB, in the first column, and LSBM, in the second column, have the same destruction effect on the image. This is because the amount of change that the LSB method caused by changing the last bit is equal to the amount of change that the LSBM method caused by randomly increasing or decreasing the number by one on the color value of one pixel. In the third column, it is seen that the noise added with the LSD method is more than in the other two. The fourth column shows no difference between the images where the LSB and LSBM methods were used. Even if these two methods achieve the same results on the last bits, the number values are different, and this is the reason for the difference.

<table>
<thead>
<tr>
<th></th>
<th>Original image-LSB</th>
<th>Original image-LSBM</th>
<th>Original image-LSB</th>
<th>LSB-LSBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>51.79 dB</td>
<td>51.79 dB</td>
<td>40.68 dB</td>
<td>48.81 dB</td>
</tr>
<tr>
<td>MSE</td>
<td>0.43</td>
<td>0.43</td>
<td>5.59</td>
<td>0.86</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.995</td>
<td>0.994</td>
<td>0.326</td>
<td>0.990</td>
</tr>
<tr>
<td>UQI</td>
<td>0.998</td>
<td>0.998</td>
<td>0.996</td>
<td>0.997</td>
</tr>
</tbody>
</table>

In Figure 11, which is filled with colors, the application of the LSB method can be detected, whereas the detectability of the LSD method’s application is very low when visual attack is used. Applied statistical attack becomes ineffective on color intensive complex images (Figure 12).

**Figure 11.** Colorful image that is more complex than aircraft image.
Applied methods were examined by adding noise and using compression in terms of robustness. The salt and pepper, Gaussian, and speckle noise methods were used. In addition, the images were compressed in JPEG format and then decompressed. It was seen that these methods do not provide robustness. For the purpose of redounding robustness, distributed and seldom hiding with PRNG and CRC can be used. It is obvious that they would reduce capacity.

Stego folder lock application can hide data by using both image and audio domains together in AVI 1.0 video files. For example, if we take a video file that has a $352 \times 240$ frame size, 29 fps frame rate, and 33 s length, the image domain of this video will be the size of $352 \times 240 \times 3 \times 29 \times 33 \approx 230$ MB. If we assume that the same video has a stereo audio that has a 16-bit sample size and 44,100 kHz sample rate, the audio domain of this video will be $44,100 \times 16/8 \times 2 \times 33 \approx 5$ MB. If the entire capacity of both the image and the audio domain of this video is used, $1/47$ percent of hidden information will be placed in the audio domain and $46/47$ percent of hidden information will be placed in the image domain. Because the application hides secret data as packages, some in the image domain and then some in the audio domain, one after another, this method can be named as the lattice approach. This way security is enhanced.

Experiments performed on the video steganography-applied files by using the zipper method with stego folder lock application are given below.

The video’s audio domain wave form and spectrogram graphics, as a result of hiding with the LSBM and LSD methods, are seen in Figures 13 and 14. The effect of the visual attack on the image domain of the same video shows that the effects are the same when plain image and single video frames are compared.

5. Conclusion

In this study, a flexible library was designed for data hiding. In addition to being independent from mediums and methods, the library was also platform-independent, since the Java programming language was used.

In order to increase complexity and security, the lattice approach, which uses image and sound fields of video files, was proposed (Figure 15). Real-time hiding of the webcam-captured video into the image field was also provided in the library. Thus, the detection of the difference from the actual video file was prevented, and the ideal video required for the data to be hidden was taken.

As a result of the tests that were applied on raw image files, it was seen that in flat images, the LSB and
LSBM methods did not cause a perceptible difference, although the LSD method did. All 3 methods applied to the flat images can be detected by visual attacks against the last bits and by chi-square attacks. On the other hand, all 3 methods seem to be imperceptible, and chi-square attacks show no result when the hiding is performed on complex images. All 3 methods can be detected by visual attacks; however, the LSD method was found to be more resistant than the other two.
The number of applications that can perform data-hiding in videos is very limited. In this study, unlike previous studies, the sound field as well as the image field within the video file were used for hiding. Consequently, complexity and security levels were increased.

References


