Pakistan Journal of Information and Technology 2 (2): 116-127, 2003
ISSN 1682-6027

High Quality Steganography Model with Attacks Detection

Ahmed Al-Jaber and Ibrahim Aloqily
Department of Computer Science, King Abdullah II School for Information Technology, University of Jordan, Amman-Jordan

Abstract: Steganography is the art of hiding a message in a secret way that only the receiver of the cover media knows the existence of the message. This secret message must be unobserved by a human eyes, Studies have shown that Human Vision Systems (HVS) is unable to detect changes in uncorrelated areas of the digital media, due to the complexity of such areas, where it is easy to detect changes in correlated areas. Security and quality are two important issues of steganography. In this paper we will introduce an algorithm that uses the LSB and those inhomogeneous areas of the cover image to hide a message. In this algorithm error correction code is used to increase the probability of retrieving the message, as well as the receiver will be able to detect if there is any alterations in the cover Media, in this case the receiver Informs the sender about this alterations.

Key words: Information hiding, steganography, human vision system, error correction codes, Least Significant Bit (LSB)

Introduction

Recent advances in computer technology and the development of the internet technology-has led to an increased interest in private communication, which can be achieved by using cryptography and/or steganography. Cryptography conceals the message by scrambling the data being communicated; while steganography hides the message in innocent computer files- such as digital pictures or digital audio (Chang-Hasing Lett and Yeuan-Kuen Lee, 1999).

The word steganography comes from the Greek *steganos* (covered or secret) and *-graphy* (writing or drawing) and thus means, literally, covered writing (Petiticolas *et al.,* 1999). It conceals the message being communicated in another message or digital media.

In steganography image quality and security are two important factors. Quality implies that stego cover should not be visually distinguishable from the original digital cover; while security implies that the message should be undetectable and no one other than the eligible personnel should be able to extract the secret message. Image quality can enhance the security of the message (Chin-Chen Chang *et al.,* 2002).

There exist a large number of Steganography techniques for hiding a message in different digital medias, among these- and possibly the easiest one- hides the message in the LSB of a bitmap graphic. Changing the LSBs causes an imperceptible change to the digital image. Without a direct comparison between the original image and the altered image there isn't any way to tell that something is changed.

Data hiding is usually achieved by altering some non-essential information in the cover image. One simple approach is to use the Least Significant Bit (LSB) of each pixel in the cover image in order to hide one bit of the intended hidden message (Bender *et al.,* 1996); as this is unlikely to affect the cover image. Therefore a number of software programs have adopted this method (Wayner *et al.,* 1993). There are two types of LSB methods,- fixed sized and variable sized. The former embeds the same number of message bits in each pixel of the cover image. In the variable sized the number of LSBs used depends on the contrast and luminance characteristics (Yeuan-Kuen Lee and Ling-Hwei Chen, 2000). To decrease the effect of hiding information in the cover image a genetic algorithm is proposed in (Wang *et al.,* 1998); while hiding algorithm based on the conventional key stream generator is proposed in (Franz *et al.,* 1996). Hiding information for security documents is discussed in (Gruhl and Bender, 1998) and a review of data hiding techniques is discussed in (Schyndel *et al.,* 1994); (Petitcolas *et al.,* 1999) and (Anderson and Petitcolas, 1998).

Background and motivation

The model for invisible communication was first proposed by (Simmons, 1998) as the "prisoners' problem." In this scenario Alice and Bob are in jail for some crime they committed and are thrown in two different cells. And wish to develop an escape plan; all there communication must go through a warden named Windy. She will not let them communicate through encryption and if she noticed any suspicious communication, she will place them in solitary confinement. So both must communicate invisibly; they have to set up a subliminal channel. Subliminal channel are discussed in (Simmons, 1998).

Steganography and data hiding can be defined as follows:

Given a cover message C and an embedded message M, a steganography scheme should provide function F_c and a data retrieving function F_r such that

$$C' = F_c (N, M, K)$$
$$F_r (C', K) = F_r (F_c (C, M, K)) = M$$

Where K is a secret key. That is, F_r can extract the embedded message M from the Cover image C hidden by F_c. Further more it should be hardly discovered that C' has been hidden with data. Under the context of using any cover image, this could mean that C' is looks like C.

Information hiding has applications in many Military and intelligence agencies that require unobtrusive communications, Criminals, Law enforcement and counter intelligence agencies and Schemes for digital elections and digital cash (Petotcolas *et al.,* 1999).

One of the very well known algorithms used to embed a message M, in a cover image C, is the least significant bit (LSB), in which the embedding process consists of choosing a subset $\{ j_1, \ldots, j_{l(m)} \}$ of cover elements and performing the substitution operation by exchanges the LSB of pixel Cji by m_i, where mi can be 0 or 1. One could also imagine a substitution operation that changes more than one bit of the cover. In the extraction process, the LSB of the selected

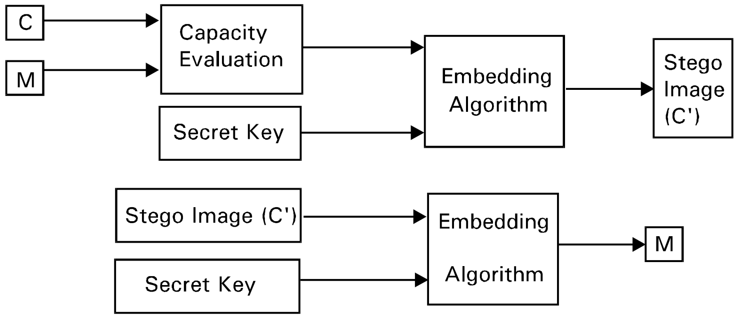


Fig. 1: The proposed model

cover elements are extracted and lined up to reconstruct the secret message.

Steganography systems are extremely sensitive to cover modifications, such as image processing techniques (like smoothing, filtering, compression, ... Etc.). Compression can result in total information loss. Lossy compression techniques try to reduce the amount of information by removing imperceptible signal components and so often remove the secret information, which has previously been added. If the stego cover undergoes any modification after being transmitted we say that an attack took place.

The LSB algorithms described in literature can be attacked in various ways, by simply changing the LSB's of the stego image and no matter what was the algorithm used to embed the secret message, the secret message will be destroyed. In such case the receiver of the message has no way to tell that an attack took place. In this paper we will develop an algorithm that uses the LSB to hide a secret message and the error correction code will be used to increase the probability of retrieving the secret message if the stego cover undergoes a slightly simple modification, as well as the receiver will know if there was an attack happened to the stego cover during or after transmission. In this case the receiver will be able to inform the sender about the attack, which will help both of them to insure the secret communication. This motivates the work in this paper.

High quality steganography model with error correction

Using the LSBs to hide a message inhibits a risk of loosing all or most of the message if an attacker changes some or all of the LSBs. In this paper we will introduce an algorithm that uses the LSB of the cover image to hide a message and makes use of the Error Correction Code to detect attacks and to increase the probability of retrieving the message. Fig. 1 shows the proposed model. This model takes as an input a cover object (C) that will be used to hide the secret message (M), (In our case the cover object is a Gray-scale image while the secret message could be of any kind). This system produces as an output a stego image that will be used later to extract the message from it.

The implementation consists of an Embedding Phase and an Extracting Phase. In the embedding process the cover object pass through two major steps. The first step divides the cover into nxm Blocks and calculates statistical information according to the following equation:

$$*b = \sum_{i=1}^{p-1} *Cbi - Cb(i+1)* \qquad (1)$$

Where:

P = n*m, that is the number pixels in block. Cbi : is the pixel *i* at a cover block

For highly correlated blocks we can expect to be a small value and it will have a larger value in the very complex blocks. The following example shows Equation 1 in action:

Example

Suppose we are given two 3x3 blocks a and b as shown bellow.

108	210	80	210	212	213
40	50	240	209	210	209
60	120	110	210	213	212
Block (a)			Block (b)		

Then *b for the two blocks will be 768 and 16 respectively, so block (a) represents an inhomogeneous block with large value, while block (b) is a highly correlated block with small value. In our algorithm we will concentrate more in blocks with large value.

*b Will be used to specify the blocks that will be used later to hide the message bits. Using a predefined threshold (J) we choose all blocks having *b greater than J to construct the Position Matrix (a matrix contains the coordinates of the blocks used for hiding the message)**.** Once these blocks are known, the number of these blocks determines the Maximum Message Size (MMS). At the end of this step the MMS is returned along with the Position Matrix.

In the second step the size of the secret message is checked to be less than or equal to the MMS. If it is not, then a different cover will be chosen, or the MMS will be increased by decreasing J. In the Extracting Phase the same threshold (J) used in the Embedding Phase is used -along with the stego key- to specify the locations being used to store the message and then extracting the message bits.

Embedding phase

I. Algorithm: Embedding Algorithm

Input : The cover image (C), The secret message (M).

Output : Stego Image

Step 1 : Convert the secret message (M) to a stream of bits, L(M) is the length of M in bits.

Step 2 : Run the Media-Test algorithm

Step 3 : If flag is false then the cover is not suitable for the secret message.
Choose another cover and go to step 2.

Step 4 : Run the Hide algorithm.

Return.

II. Algorithm: Media-Test

Input: The cover image (C), L (M).

Output: position matrix, threshold value and flag.

Step 1: Divide C into nxm Blocks.

Step 2: For each block calculate *b using Equation 1.

Step 3: Set a threshold value (J).

Step 4: If J is above a certain value, set flag to false. Return.

Otherwise each block with the property (*b>J) will be used for embedding. The number of such blocks is the MMS.

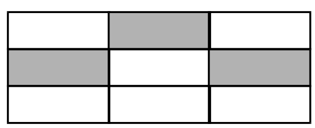
Step 5: If the size of the secret message L (M) is greater then MMS go to step 3 and decrease the value of J.

Step 6: Construct the position matrix such that (*b>J).

Return.

Hiding the message

In this stage, the secret message will be embedded inside the cover image. The block locations that will be used for embedding the secret message are located in the position matrix. In order to show what will happen in this stage, consider a block of 3X3, as shown below we will hide three copies of the same message bit in one block in the shaded areas. This redundant information will enable use to check for the consistency of the extracted message bits. Note that we could use different shaded areas each time but they are identical for all blocks, so the exact locations used inside the block must be included in the stego key.



3x3 block, the shaded areas will be used to hide the message.

The embedding is done according to the following formula:

$$C1\,'\,C6\,1,0› \begin{bmatrix} C1 \text{ ¿ } c7\,'\,0 & \text{if } m\,'\,c8 \\ C1 \text{ ¿ } c7\,'\,1 & \text{if } m <> c8 \end{bmatrix} \qquad (2)$$

$$c2\,'\,c1 \text{ ¿ } c7 \quad \text{for all pixels} \qquad (3)$$

In (2) we compare the message bit with the most significant bit of the cover pixel. If they match, we set the exclusive-or of c1 and c7 to 0 by changing only the LSB c1 if necessary. In (3) we set c2 to hold the logical exclusive-or between c1 and c7. This way we are embedding one message bit by changing at most two bits, namely c1 and c2.

III. Algorithm: Hide

Input: The cover image (C), The secret message (M),
The position Matrix (P)

Output: The stego image.

For I=1 to the size of secret message L(M)
Choose mi

For j=1 to the number of copies
Choose a pixel in position Pi (Cp), Such that Cp is not used before.

If mi = most significant bit of Cp

Set the exclusive-or between bit 7 in Cp and LSB of Cp to zero by changing the value of the LSB of Cp.
Set the exclusive-or between the LSB of Cp and bit 7 of Cp to be the value of bit 2 of Cp.

Else
Set the exclusive-or between bit 7 in Cp and LSB of Cp to one by changing the value of the LSB of Cp.
Set the exclusive-or between the LSB of Cp And bit 7 of Cp to be the value of bit 2 of Cp.

End If

End For

End For

Return.

Extracting phase

In the Extracting Phase a reverse process is used. From the secret key we will first identify the threshold value (J) -which will be used to identify the blocks containing the secret message- then the correct sequence of the message bits and the targeted pixels at those blocks will be identified.

We extract each message bit from a cover pixel according to the following:

$$c1^* = c2 \oplus c7 \qquad (4)$$

$$m = \begin{bmatrix} c8 & \text{If } c1^* \oplus c2 = 0 \\ \overline{c8} & \text{otherwise} \end{bmatrix} \qquad (5)$$

First the Position matrix is constructed using J and Equation (1), this will identify the blocks used for embedding the message. Also the stego key locates positions of the copies inside the blocks. Using (4), we compute the LSBs of the Stego cover to construct the computed LSB matrix (CLSB), The actual LSBs of the stego cover conforms the ALSB matrix, It will be of the same size as CLSB (nX3):

$$CLSB = \begin{bmatrix} L11 & L12 & L13 \\ L21 & L22 & L23 \\ . & & \\ . & & \\ Ln1 & Ln2 & Ln3 \end{bmatrix} \quad ALSB = \begin{bmatrix} A11 & A12 & A13 \\ A21 & A22 & A23 \\ . & & \\ . & & \\ An1 & An2 & An3 \end{bmatrix}$$

Each row in CLSB and ALSB represents three LSBs from the same block (either computed or actual), n = L(M) the number of message bits.

$$R_{ij} = |CLSB_{ij} - ALSB_{ij}| \qquad (6)$$

Using Equation 6 we get the result matrix R that has the following properties:

- R = 0 ⇒ ALSB and CLSB are identical, that means the stego cover hasn't been attacked and the extracted message will be correct and 100% similar to the original message. CLSB will be used along with Equation 5 to retrieve the message.
- R ≠ 0 ⇒ ALSB and CLSB are not identical; in this case we might suspect that an attack changes some or all of the message bits. To work around this case, we note that the result matrix R will has 1's in the locations of the modified pixels and 0's in those locations that were not affected by the attack. Now we calculate the probability that R has 1's p(R=1), which reflects the percentage of possibly lost information.
- If P(R=1) < T, (where T is a value represents the amount of acceptable information loss upon which the message will still readable), the extracted message is readable and accepted and we will try to reduce P(R=1). As mentioned before each row in CLSB represents three copies that can be used independently to extract the same message bit, which means that if a certain row in CLSB has a value equals to zero in one of its three values, then we can extract the hidden message bit correctly. Those rows that have their three values equals to one have a permanent error that can't be corrected. We will construct a modified CLSB containing the locations of zeros in CLSB and any location if none exists. This will increase the probability of retrieving the hidden message. The modified CLSB will be used along with Equation 5 to retrieve the message.

C If P(R=1) > T, that means the stego cover undergoes a hard attack and the message could not be retrieved correctly, we will try to reduce the error –as mentioned in the above step- and test for P(R=1) once more, if it is still greater than T, the message will be discarded otherwise the message is acceptable.

In the last two cases listed above the receiver of the message will be able to know that there is an attack on the stego cover, he can inform the sender that their communication channel is being watched and its better to be changed.

Algorithm: Extract

Input: The Stego Image (C) , the Secret Key (K)
Output: Embedded Message. Attacked (true, false).

Step 1: Identify the threshold value J, the sequence of the secret message bits, the number of copies and the block size.
Step 2: Divide C into nxm Blocks.
Step 3: For each block calculate *b using Equation 1.
Step 4: Using the threshold J and *b determine the blocks containing the secret message, reconstruct the position matrix.
Step 5: Construct CLSB matrix using Equation 4. and construct ALSB matrix.
Step 6: Construct the result matrix R using Equation 6.
Step 7: **IF** R=0

Use the CLSB and Equation 5 to retrieve the message bits and go to step 10.

Else
IF P(R=1) > T
Set attacked = true.

For all R =1 choose according to majority one of the three bits in CLSB at the same row position in R. and build a modified CLSB

Use the modified CLSB and Equation 5 to retrieve the message bits and go to step 10.

Else
Set attacked = true.

Construct 4 messages as described above.
End IF
End IF

Step 8: Line up the extracted bits according to the correct sequence in order to form the secret message.

Return.

Experimental Results

The proposed model is tested on a number of Gray scale images. Table 1 below shows the three main images (Lena, Baboon and the OldMill image) used in the tests and their full capacity in bits using different threshold values. Note that the proposed model suggests hiding three copies of the same message in each image block in order to increase the probability of retrieving the original message –to recover from slightly simple attacks. So the actual embedding capacity is three times the values listed in Table 1.

Table 1: Maximum message size for different thresholds

Image	Threshold (J)		
	50	100	150
Lena (256x256)	8904	4992	3093
Baboon (512x512)	76569	54876	39276
OldMill (512x512	78294	66231	51693

Fig. 2a below shows the original Lena image and fig. 2b shows the same image after embedding the full capacity message size, the secret messages are generated randomly using a Pseudo Random Number Generator (PRNG), The Peak Signal to Noise Rate (PSNR) is used to evaluate the image quality. The PSNR of a gray-level image is defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \, dB$$

The Mean Square Error (MSE) for an N x N gray-level image is defined as follows:

$$MSE = \left(\frac{1}{N}\right)^2 \sum_{i=1}^{N} \sum_{j=1}^{N} \left(X_{ij} - \overline{X_{ij}}\right)^2$$

Where Xij is the cover pixel value and $\overline{X_{ij}}$ is the corresponding stego cover pixel value.

Table 2 shows the PSNR of the three images after embedding the full capacity. In steganography image quality is a very important issue. The more image quality the stego image has the more secure the steganography system will be. From Table 2 we can conclude that our model produces high image quality with no visual difference between the original and the stego image. This is proven in our experiments -as shown in Fig.2b, Fig.3b and fig.4b, (which shows the stego image after embedding the full capacity.) Fig.2a, Fig.3a and Fig.4a are the original images.

Table 2: Peak Signal to Noise Ratio (PSNR)

Image	PSNR
Lena (256x256)	48.048
Baboon (512x512)	44.280
OldMill (512x512	44.165

Table 3: percent of error in the extracted message(average quality factor)

	Percent of error	
Image	E1	E2
Lena (256x256)	0.1498	0.0012
Baboon (512x512)	0.1561	0.0017
OldMill (512x512	0.1427	0.0010

Table 4: percent of error in the extracted message(low quality factor)

	Percent of error	
Image	E1	E2
Lena (256x256)	0.4901	0.0442
Baboon (512x512)	0.5009	0.0433
OldMill (512x512	0.4973	0.0423

Table 5: percent of error in the extracted message(mean and median filtering)

	Percent of error (Mean Filtering)		Percent of error (Median Filtering)	
Image	E1	E2	E1	E2
Lena (256x256)	0.4979	0.0433	0.2877	0.0114
Baboon (512x512)	0.4988	0.0413	0.3221	0.0165
OldMill (512x512	0.4939	0.0402	0.3200	0.0179

Fig. 2(a): Lena (original)

Fig. 2(b): Lena (with a secret message)

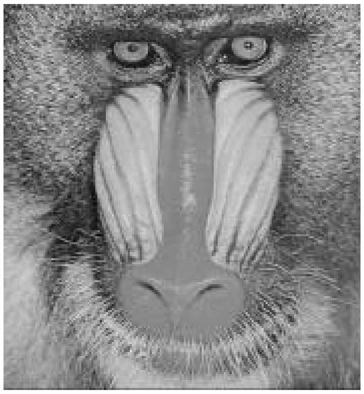


Fig. 3(a): Baboon (original)

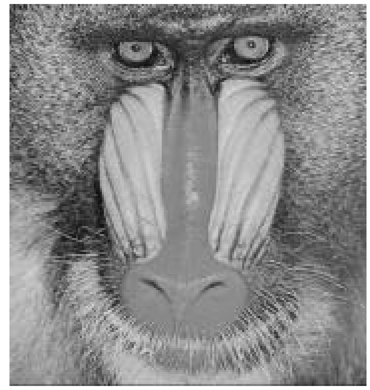


Fig. 3(b): Baboon (with a secret message)

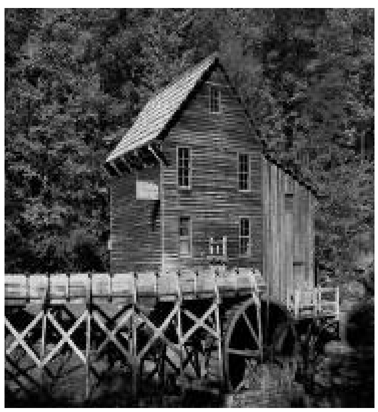


Fig. 4(a): OldMill (original)

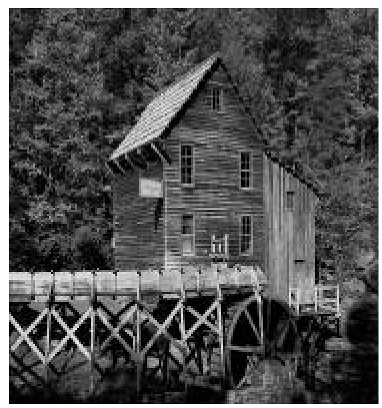


Fig. 4(b): OldMill (with a secret message)

The resistance of the proposed method to various distortions was studied in a series of experiments on grayscale images. First a set of experiments dealt with the resistant of the method to JPEG compression, we generated different messages and used the described algorithm to embed them inside the three pictures mentioned above. Then, we compressed the images using the JPEG algorithm and a certain quality factor. The results shown in Table 3 represents an average quality factor, while table 4 represents low quality factor.

In Table 3 and 4, E1 is the amount of error in the extracted message after being compressed using JPEG algorithm and E2 is the amount of error in the message after being reduced using our algorithm. In table 3 E1 is about 15% of the message, which means that 15% of the message is lost by compression, the proposed algorithm was able to reduce that error to 0.1% . In table 4 about 50% of the message is lost and the algorithm was able to reduce that error to 4%. This leaves the extracted message in an acceptable state. Further tests shows that a message with $T < 0.04$ is an acceptable message.

In the other set of experiments we tested the robustness of the proposed algorithm against 3x3 mean and median filtering. Results can be seen in Table 5. We can observe that the algorithm is able to reduce the error resulting from median and mean filtering to 4 and 1%, respectively.

Finally the proposed steganography model produces high quality stego Images, robust and more secure comparable to the well-known usual LSB scheme. In addition to that the concept of error correction code was introduced and implemented to increase the probability of retrieving the secret message as well as to detect attacks. This model makes use of Human Vision System (HVS) properties and embeds the message in the most important areas of the image. Experimental results show that this method is efficient and effective- and that it produces high quality stego images.

Future research may focus on applying this model for the color images, also further investigations can be done over different values of T (the acceptable error in the message) for other types of steganography, such as text, images, audio and video

References

Anderson, R.J. and F.A.P. Petitcolas, 1998. On the Limits of Steganography, IEEE J. on Selected Areas in Communications, 16: 474-481.

Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. Techniques for Data Hiding, IBM System, 35: 3–4.

Chang-Hsing Lee and Yeuan-Kuen Lee, 2002. An Adaptive Digital Image Watermarking Technique for Copyright Protection, IEEE Transactions on Consumer Electronics.

Chin-Chen Chang, Tung-Shou Chen and Lou-Zo Chung, 2002. A Ssteganographic Method Based Upon JPEG and Quantization Table Modification. Information Sciences, 141: 123-138.

Franz, E. *et al.*, 1996. Computer-Based Steganography, in Information Hiding, Springer Lecture Notes in Computer Science, 1174: 7–21.

Gruhl, D. and W. Bender, 1998. Information Hiding to Foil the Casual Counterfeiter, in Proceedings of the Workshop on Information Hiding, IH'98, Portland, Oregon, USA.

Petitcolas, R. Anderson and M. Kuhn, 1999. Information hiding – A Survey, Preceeding of the IEEE, 87: 1062-78.

Simmons, G.J., 1998. The prisoners' Problem and Subliminal Channels, in CRYPTO'83, pp: 51-67.

Simmons, G.J., 1998. The History of Subliminal Channels, IEEE J. on Selected Areas in Communications, 16: 452-462.

van Schyndel, R.G., A.Z. Tirkel and C.F. Osborne, 1994. A Digital Watermark, in IEEE Int. Conf. Image Processing, 2: 86–90.

Wang, R.Z., C.F. Lin and J.C. Lin, 1998. Image Hiding by LSB Substitution and Genetic Algorithm, in Proceedings of International Symposium on Multimedia Information Processing, Chung-Li, Taiwan, R.O.C.

Wayner, P., 1993. Should Encryption Be Regulated, Byte, McGraw-Hill, pp: 129-134.

Yeuan-Kuen Lee and Ling-Hwei Chen, 2000. High Capacity Image Steganographic Model, IEEE Proceedings Vision, Image and Signal Processing, 147: 288-294.