FPGA Implementation for Image Steganography

Technique Using X-Box Mapping

Mrs.Manjula.Y¹, Mr.Jagadeesha.D.H², Dr. K.B ShivaKumar³

Asst.prof, dept. of E&C, SSIT, Tumkur,india¹
M.Tech [DE], Dept of E&C, SSIT, Tumkur,india²
Prof, Dept of TCE, SSIT, Tumkur,india³

Abstract— Image steganography is a method of hiding the information in a cover image in such a way that only intended recipient can know that there is a hidden message. Least Significant-Bit (LSB) based approach is most popular steganographic techniques in spatial domain due to its simplicity and hiding capacity. This paper presents a new technique based on LSB for Image steganography using X-box mapping where we have used several X-boxes having unique data. The embedding part is done according to the X-boxes and this technique give more security to the secret information without knowing the mapping rules no one can extract the secret data. In this paper a novel image steganography technique based on x-box mapping is implemented in FPGA . It is observed that this implementation algorithm gives better PSNR values and security compared to previous algorithms.

Keywords— Steganography, X-Box, LSB Technique, Information Hiding.

I. INTRODUCTION

Image steganography is a method of hiding the information in the cover image. Steganography, the word steganography is coming from the Greek words, stegos, means covered and the graphia means writing, it is the art and science of hiding the message in such way that only recipient can know the their is a exists of the message . Using this method, you can embed a secret message inside a piece of unsuspicious information and send it without anyone knowing of the existence of the secret message. Steganography and cryptography are very closely related. Cryptography hides the content of the messages so they cannot understand. Steganography on the other hand, will hide the both content of the message and also the existence of the message. In some situations, sending an encrypted message will arouse suspicion while an “invisible” message will not do so. Both sciences can combine to produce good protection of the message. when the steganography fails and the message can be detected, it is still of no use and it will be similar to encrypted using cryptography techniques. The principle defined by Kerckhoffs for cryptography, also holds good for steganography. The quality of a cryptographic technique should only depend on a small part of information, that is secret key. The same is valid for steganographic technique: knowledge of the system that is used in these technique, should not give any information about the existence of secret messages. Finding a message should only be possible with knowledge of knowing the secret key or stego key.

As the development of Internet technologies increases, the transmission of digital media is now-a-days convenient over the networks. But secret message transmissions over the Internet system suffer from serious security overhead. So, protecting of secret messages during transmission becomes an important issue In this generation, steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Figure.1 shows the block diagram of a simple image steganographic system.
II. RELATED WORKS

Least significant bit (LSB) insertion is a common, simple approach in the image steganography technique. In this method embedding information in a cover image in such a way that the least significant bit (the 8th bit) of some or all of the bytes inside cover image is changed to a bit of the secret message. For example a grid for 3 pixels of a 24-bit image can be as follows:

\[
\begin{align*}
&00101101 \quad 00011100 \quad 11011100 \\
&10100110 \quad 11000100 \quad 00001100 \\
&11010010 \quad 10101101 \quad 01100011 \\
\end{align*}
\]

When the number 200, which binary representation is 11001000, and it is embedded into the least significant bits of this part of the cover image, the resulting grid is as follows:

\[
\begin{align*}
&00101101 \quad 00011100 \quad 11011100 \\
&10100110 \quad 11000100 \quad 00001100 \\
&11010010 \quad 10101101 \quad 01100011 \\
\end{align*}
\]

Although the number was embedded into the first 8 bytes of the grid, only the three underlined bits needed to be changed according to the secret message. On average, only half of the bits in an image will need to be modified to hide a secret message and it requires maximum cover size. Since there are 256 possible intensities of each primary color, changing the least significant bit of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye, thus the secret message is successfully hidden.

With a well chosen image, even we can hide the message in the least as well as second to least significant bit and still not see the difference. In the example discussed above, consecutive bytes of the image data – from the first byte to the end of the message – are used to embed the information. This approach is very easy to detect the secret message. A slightly more secure system is for the sender and receiver to share a secret key that specifies only certain pixels to be changed. It should an adversary suspect that LSB steganography has been used, no one has way of knowing which pixels to target without the secret key. LSB matching (LSBM), LSBM revised (LSBMR) [2] and Edge Adaptive based LSBMR [3] steganography techniques are popular LSB like steganography methods.

III. PROPOSED IMAGE STEGANOGRAPHY

**IMAGE ENCODING:** This technique is based on mapping of different values from X-Boxes.

X-Boxes are a 2x2 matrix, where 16 (0 to 15) values are stored as given fig[5].

**Generation of four different X(X-OR)-boxes:**

![Figure 1.2 X-Boxes](image)

To put values in X-boxes, we use X-OR property:

- \(0 \text{ XOR } 0 = 0\), \(1 \text{ XOR } 1 = 0\) and \(0 \text{ XOR } 1 = 1\), \(1 \text{ XOR } 0 = 1\).

For example 13 is inserted in any one of the four X-Boxes as follow:

\[13 = 1101 = 11 \text{ XOR } 01 = 10\]

Thus the position of 13 is 2nd row and 1st column.
Bit Division:

Then, we need to take the cipher encrypted image; say with dimension 64x64. Now, we convert the values from decimal to binary. For example, the first pixel value of the encrypted image = 149. Then, binary of

\[(149)_{10} = (10010101)_2\]

Now, we need to divide this 8-bit value into 4 parts taking 2 bits in each.

\[(149)_{10} = (10010101)_2\]

X-box Mapping:

Now we just map the values of b1, b2, b3, b4 from the X-mapping box. First we take b1 = 10; then we search the value of 1st row and 0th column of the X-I box; After mapping we get the value \((13)_{10} = (1101)_2\). Similarly we get mapping values for the b2, b3, b4; we get in the same way 11, 14, 1 sequentially.

Bit insertion into the cover image:

After getting the new mapping values we insert these values into the cover image. We placed these values into the 4 bit LSB of cover image sequentially. First we take the pixels one by one from the cover image. The 4 LSB bits are replaced by 13, 11, 14, 1 respectively.

\[
\begin{array}{cccc}
23 & 110 & 225 & 197 \\
167 & .... & .... & .... \\
...... & .... & .... & .... \\
...... & .... & .... & .... \\
...... & .... & .... & .... \\
...... & .... & .... & .... \\
...... & .... & .... & .... \\
\end{array}
\]

Figure 1.4 cover image
Here we take the pixels sequentially.

\[(23)_{10} = (00010111)_2\]
\[(110)_{10} = (01101110)_2\]
\[(225)_{10} = (11100001)_2\]
\[(197)_{10} = (11000101)_2\]

**Formation of Stego Image:**

After getting the new pixel values we form the stego image. The pixel values 29, 107, 239, 193 are placed into the position of the previous values. Similarly we take the pixels one by one and insert the cipher image into them and replaced them. Thus we get the Stego image.

**Image Decoding:**

To decode the stego image in the receiver side we just perform the following steps:

**Generate the 4LSB bit s from the Stego image:**

We take the pixels one by one from the stego image. Transfer it into the binary values and get the 4 bits (LSB) values from it.

**Retrieve the inserted bits of cipher image:**

We take the 4 LSB bit of the stego image that are 1101, 1011, 1110, 0001; then we perform the XOR operation of the 4 bits. First we the 2 bits, and we do the XOR operation with the other 2 bits.

**Concatenation of the result of the XOR operation:**

Now we concatenate the 4 results of the XOR operation. After that we get the 8 bits. Then from them we transfer it into the decimal value. Concatenated value is:
Concatenation the results of XOR operation

Generation of cipher image:
Now the generated value is placed into the first position. Similarly we take the next value of the stego-image and repeat the steps 1 to 4. And we get the 210, 37 etc. Ultimately we get the total cipher image.

| 29 | 107 | 239 | 193 | ... | ...
|----|-----|-----|-----|-----|-----
| 159| ... | ... | ... | ... | ...
|    | ... | ... | ... | ... | ...
|    | ... | ... | ... | ... | ...
|    | ... | ... | ... | ... | ...

Fig 1.8 (64 x 64) Cipher Image
These are the total process of the X-box Steganography.

IV. EXPERIMENTAL RESULTS AND SECURITY ANALYSIS
In this we will discuss the experimental results along with the security analysis. This embedding technique is no doubt a strongest Steganography technique than normal LSB encoding technique. Because, we embed each 2 bits of Cipher Image into the 4 bit of Cover Image. Again before insertion we coded these two bits by some mapping box into another form. So if one can understand that something is embedded in it, but the mapping will be totally unknown to him. So to extract the image is really a tough job.

<table>
<thead>
<tr>
<th>image</th>
<th>Capacity</th>
<th>Size(pixel)</th>
<th>PSNR(db)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena.jpg</td>
<td>25%</td>
<td>64</td>
<td>31.11</td>
</tr>
<tr>
<td>Cameraman.jpg</td>
<td>25%</td>
<td>64</td>
<td>31.78</td>
</tr>
<tr>
<td>Koala.jpg</td>
<td>25%</td>
<td>64</td>
<td>31.87</td>
</tr>
<tr>
<td>Penguin.jpg</td>
<td>25%</td>
<td>64</td>
<td>31.80</td>
</tr>
</tbody>
</table>

V. CONCLUSION
In this paper, FPGA implementation is done for X-Box mapping based image steganography process to improve security and image quality compared to the existing algorithms. Looking at the PSNR values it is observed that proposed implementation is fared with better payload security and retaining the imperceptibility of the image.

REFERENCES
[4] “A new scheme for image hiding based on digital images” by Yan-ping Zhang, Juan Jiang, Chao Xu, Bo Hua.