Image Steganography using Secret Key & Gray Codes

Sunny Dagar, Vinay Kumar, Yogendra Bagoriya

Abstract*—Steganography is an art of hiding some data into another data. Steganography is a very ancient technique which is used to send secret messages inside a simple message e.g. message written through invisible ink etc. Image steganography is a science of hiding secret data i.e. text, audio, video etc. inside an image. In this paper, an image steganography algorithm is proposed which uses secret key and gray codes to hide the secret file inside the cover image. This algorithm takes image of any format like .jpeg, .gif, .bmp etc. as a carrier and converts it into .bmp format. As .bmp image uses lossless compression techniques so compression of .bmp image doesn't lose any information. Although this paper will not emphasis on image compression. Then the secret data bits are encrypted using gray codes and then this encrypted file is hidden in the LSB of carrier image. The main aim is to prevent the identification of presence of secret data in the carrier image. But use of key increases the security of the secret data

***Index Terms*— Steganography, Cryptography, Secret Key, LSB Coding, Gray Codes.**

I. INTRODUCTION

Steganography is the art and science of hiding messages inside a carrier so that only sender and intended recipient of the message knows about the presence of hidden message. Steganography derived from Greek words whose actual meaning is "hidden writing" (Greek word "Stegos" means "cover" and "gratia" means "writing"). Steganography comes under the assumption that if attacker has the knowledge of presence of secret data then he/she tries to decrypt it but if he/she doesn't know about presence of any secret data then how and on which he /she apply decryption. This is the basic idea behind steganography [1, 2, 3].

Cryptography is the art and science of converting the readable text into non readable text. This is generally done to protect the message to read by the unauthorized user. Unauthorized user is the user who doesn't have the authority to read the data. Cryptography is mainly done by two ways i.e. Substitution and transposition [4].

Gray Code is a form of binary that uses a different method of incrementing from one number to the next [5].

Table I

Decimal	Natural Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Table I, depicts the Decimal, Natural Binary and Gray Code Representation. With Gray Code, only one bit changes state from one position to another. This feature allows a system designer to perform some error checking (i.e. if more than one bit changes, the data must be incorrect).Gray Code is the most popular Absolute encoder output type because its use prevents certain data errors which can occur with Natural Binary during state changes. For example, in a highly capacitive circuit (or sluggish system response), a Natural Binary state change from 0011 to 0100 could cause the counter to see 0111. This sort of error is not possible with Gray Code, so the data is more reliable [5].

The advantage of steganography over cryptography alone is that messages do not attract attention of attackers. Encrypted text, don't matter how unbreakable it is, attacker somehow finds a way to decrypt but if attacker don't know what to decrypt then in this way security of secret data is enhanced. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties [6].

II. RELATED WORK

Steganography is a very vast field and lots of work has been done till date. Carrier in steganography can be an image, audio or video. Every technique has some shortcoming and some benefits.

Image steganography technique can be divided into two groups: those in the Image Domain and those in the Transform Domain [7]. Image domain technique embed message in the intensity of the pixels directly. In transform domain technique, images are first transformed and then the message is embedded in the image [8] [9].

Image domain techniques encompasses bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterized as "simple system" but in the transform domain involves the manipulation of algorithms and image transforms

Manuscript received on April, 2013.

Sunny Dagar, School of Information Technology, Centre for Development of Advanced Computing (CDAC), Noida, India.

Vinay Kumar, School of Information Technology, Centre for Development of Advanced Computing (CDAC), Noida, India.

Yogendra Bagoriya, School of Information Technology, Centre for Development of Advanced Computing (CDAC), Noida, India.

[10]. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format but many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression [9][11]. Image s is the most common over internet and also at a very large scale. So Image steganography is the most powerful and efficient technique for carrying very sensitive and top secret data as it is very difficult for the attacker to identify the potential target and read the unauthorized data.

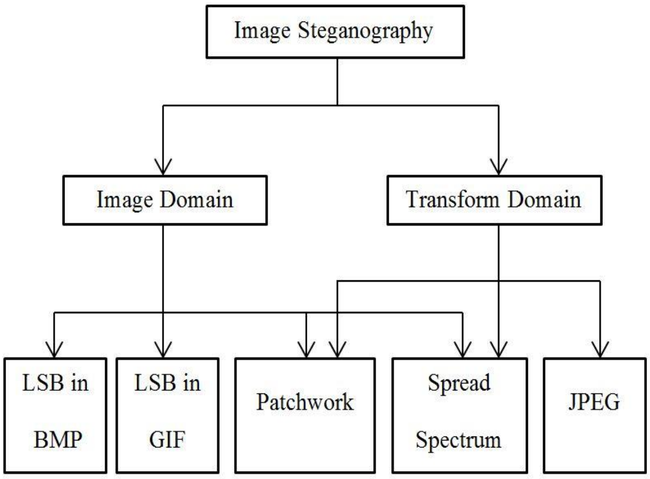


Fig. 1: Category of Image Steganography

Five main techniques of image steganography i.e. LSB in BMP, LSB coding in GIF, Patchwork, Spread Spectrum, JPEG can be compared on different parameters. These techniques have some advantages and disadvantages and are shown in the form of a table. We can easily compare these techniques. Out of these techniques LSB coding in BMP image is the simplest one with great invisibility and payload capacity. Other techniques may be stronger than LSB in BMP in some context but these techniques are very complex to understand and difficult to code.

Table II

Characteristics	**LSB in BMP**	**LSB in GIF**	**JPEG Compression**	**Patch Work**	**Spread Spectrum**
Invisibility	High	Medium	High	High	High
Payload Capacity	High	Medium	Medium	Low	Medium
Robustness Against Statistical Attacks	Low	Low	Medium	High	High
Robustness Against Image Manipulation	Low	Low	Medium	High	Medium
Independence of File Format	Low	Low	Low	High	High
Unsuspicious File	Low	Low	High	High	High

LSB coding is a technique of embedding the data bits at the Least Significant Bit of the image pixel. Example: Inserting the binary value for A (10000001)

We need 3 pixels.

Before hiding A:

00100111 11101001 11001000
00100111 11001000 11101001
11001000 00100111 11101011

After hiding A:

00100111 1110100**0** 11001000
0010011**0** 11001000 1110100**0**
11001000 00100111 11101011

So in this way only 3 bits of the 3 pixels are changed to embed A into the image. These 3 changed bits are shown bold as well as underlined.

III. PROPOSED APPROACH

In the proposed approach, LSB coding is used for hiding data in the image but before hiding data is encrypted using a symmetric key (k_S) which is known to both sender and receiver. Data is encrypted using SN function which uses gray code and symmetric key. The main advantage of SN function is that it can encrypt the plain text as well as decrypt the cipher text with small changes. So in this way SN function is very easy to understand and implement.

A. *Encryption*

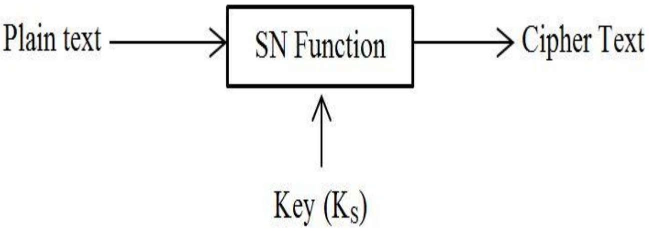


Fig. 2: Encryption Process

B. *Decryption*

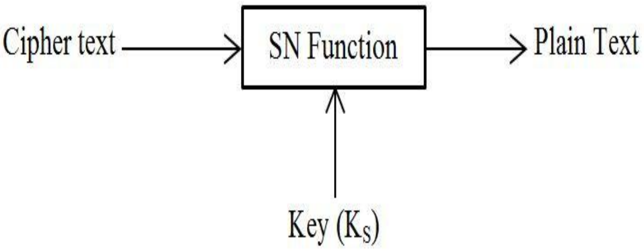


Fig. 3: Decryption Process

Encrypting and Embedding Process:

1. Take image of any format as an input.
2. Convert this input image into .bmp image format.
3. Take a data file (Text, Image, audio, Video) as an input.
4. Take a symmetric key as an input.
5. Extract every single byte of data file.
6. Apply SN function on it.
7. Take every single pixel of .bmp image.
8. Extract its RGB.
9. Hide information of data file like name, type, size etc. into the LSB of right most bottom pixels.
10. Hide the encrypted data bits into the LSB of RGB of the rest of the pixels.
11. Send this .bmp image file to the receiver.

Extracting and Decrypting process:

1. Take the received .bmp image as an input.
2. Take symmetric key of 8 bit as an input.
3. Take every single pixel of input image.
4. Extract its RGB.
5. Extract data bits from LSB of RGB.
6. Arrange these extracted bits into group of 8 bits i.e. byte.
7. Apply SN function on these bytes.
8. Extract the data file information from the rightmost bottom pixels of the received image.
9. Create and view the encrypted hidden file.

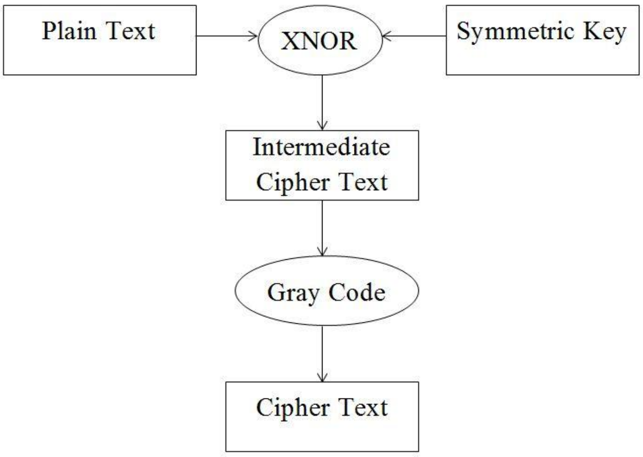


Fig. 4: SN Encryption Function

Example of SN Encryption Function:
Plain Text- 01010011
Key - 10011001
Encryption: (Plain Text) XNOR (Key)

01010011
XNOR 10011001
00110101

Gray code for 00110101 is 00101111
So our calculated cipher text is 00101111

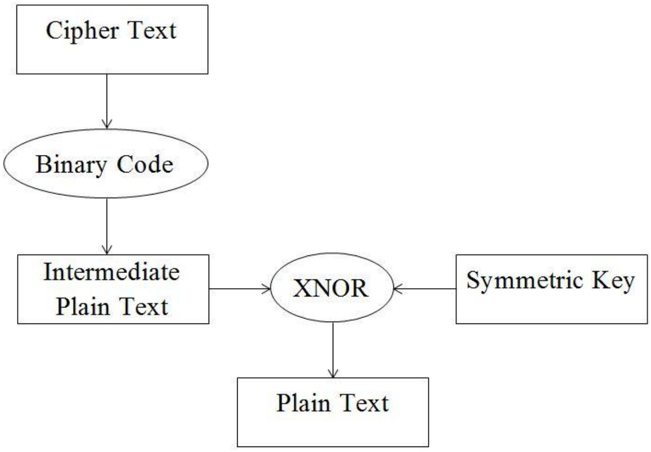


Fig. 5: SN Decryption Function

Example of SN Decryption Function:
Cipher Text- 00101111
Key - 10011001
Decryption:

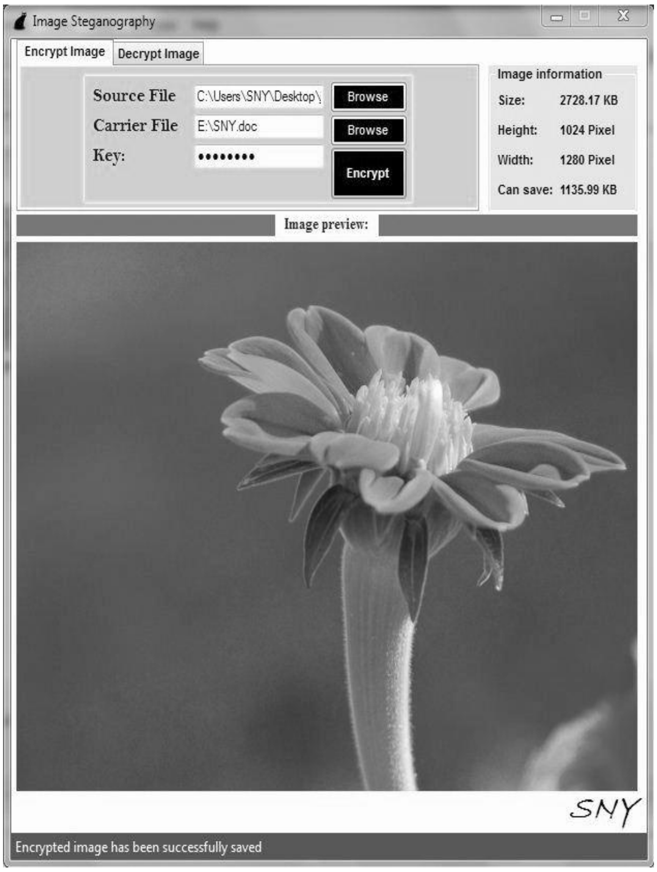
Binary code for 00101111 is 00110101
Now, (Cipher Text) XNOR (Key)

00110101
XNOR 10011001
01010011

So our required plain text is 01010011

IV. EXPERIMENTAL WORK AND RESULTS

It is very helpful to look at the screenshots of the software developed using this approach. The developed software is used for both i.e. hiding the secret information and extracting it. These snapshots easily describe the process of hiding and extraction.



Source File C:\Users\SNY\Desktop\
Carrier File E:\SNY.doc
Key: ••••••••

Image information
Size: 2728.17 KB
Height: 1024 Pixel
Width: 1280 Pixel
Can save: 1135.99 KB

Encrypted image has been successfully saved

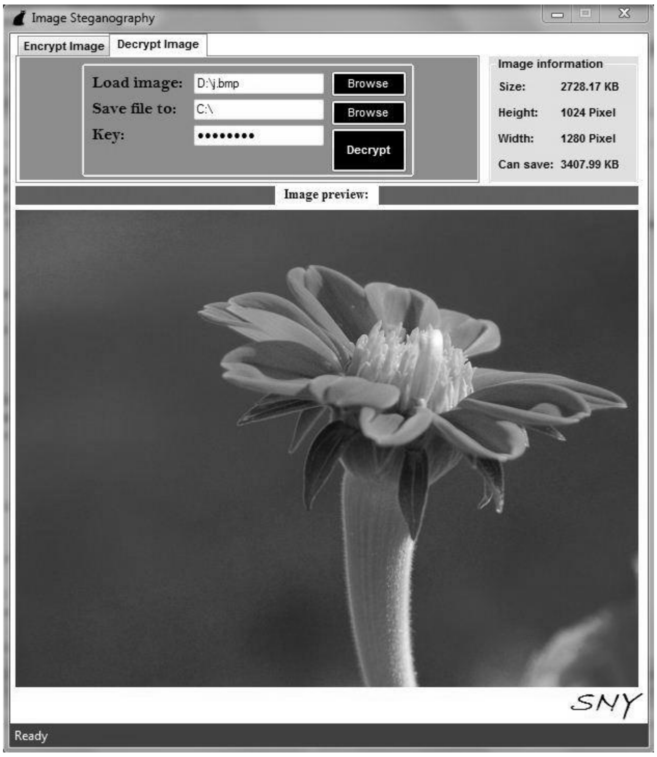
Fig. 6: Encode Form

A. Hiding Process at Sender Side

- There are two browse buttons. One loads the source file while another loads the carrier file.
- Image information is shown at the top right corner. Image information includes size in KB, height and width in pixels, can save in KB.
- Secret key ensures the confidentiality of the secret message.
- Output image is saved at the defined position.

B. Extraction Process at Receiver Side

- There are two browse buttons. One browse button loads the source image while another browse button loads the position where we wants to save the secret file which will be extracted from source file.
- Secret key is required to extract this secret file.



Load image: D:\j.bmp
Save file to: C:\
Key: •••••••••

Image information
Size: 2728.17 KB
Height: 1024 Pixel
Width: 1280 Pixel
Can save: 3407.99 KB

Fig. 7: Decode Form

Result:

Cover file: sny.bmp (2728.17 KB)
Secret File: SNY.doc (30.5 KB)
Key Used: 12352274
Output File (Stego Object): 2728.17 KB (SAME SIZE)

Fig. 8: Original Cover Image

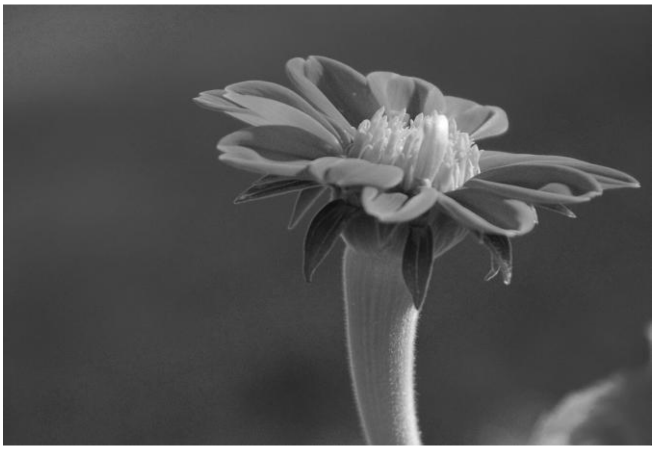


Fig. 9: Stego Image

V. COMPARISON

It is clear from the approach that use of asymmetric key increase the robustness of secret information against statistical attacks. Also, the presence of gray codes enhances the robustness against image manipulation. Taking cover image of any format makes this approach independent of file format.

Table III

Character-istics	**This Approach**	**LSB in BMP**	**LSB in GIF**	**JPEG Compression**	**Patch Work**	**Spread Spectrum**
Invisibility	High	High	Medium	High	High	High
Payload Capacity	High	High	Medium	Medium	Low	Medium
Robustness Against Statistical Attacks	High	Low	Low	Medium	High	High
Robustness Against Image Manipulation	Medium	Low	Low	Medium	High	Medium
Independence of File Format	High	Low	Low	Low	High	High
Unsuspicious File	High	Low	Low	High	High	High

VI. CONCLUSION AND FUTURE WORK

This approach eliminates the previous shortcoming of LSB coding in BMP images like low robustness against statistical attacks, low robustness against image manipulation by using very efficient SN function cryptography. Problems of dependency of file format is also corrected by this approach as this approach takes image of any format as an input and then converts it into .bmp format.

Gray coding is the most popular absolute encoder technique which helps in preventing most data errors which can occur with natural binary numbers. So this is a very secure technique.

The conceptual idea and implementation has been discussed in this paper. Future work focuses on making this approach secure, robust and efficient in terms of hiding capacity, embedding and extraction time.

ACKNOWLEDGMENT

We would like to thank our college faculty for providing us with support and guidance. We would also like to thank our parents for their continuous encouragement and support.

REFERENCES

[1] Rig Das, Themrichon Tuithung, "A Novel Steganography Method for Image Based on Huffman Encoding".

[2] A. Nag, S. Biswas, D. Sarkar, P. P. Sarkar, "A Novel Technique for Image Steganography Based on Block-OCT and Huffman Encoding". International Journal of Computer Science and Information Technology, Volume 2, Number 3, June 2010.

[3] Neil F. Jhonson, Sushil Jajodia, “Exploring Steganography: Seeing the Unseen", IEEE paper of February 1998.

[4] Kahate Atul, Cryptography and Network Security, the McGraw Hill Companies.

[5] R. Varalakshmi, Dr. V. Rhymend Uthariaraj, "A New Secure Multicast Group Key Management Using Gray Code", IEEE-International

Conference on Recent Trends in Information Technology, ICRTIT 2011 MIT, Anna University, Chennai. June 3-5, 2011.

[6] Wayner, Peter (2002), Disappearing cryptography: information hiding: steganography & watermarking. Amsterdam: MK/Morgan Kaufmann Publishers. ISBN 1-55860-769-2.

[7] J. Silman, Steganography and Steganalysis: An Overview, SANS Institute, 2001.

[8] Y.K. Lee and L.H. Chen, High capacity image steganographic model, Visual Image Signal Processing, 147: 03, June 2000.

[9] Anupam Kumar Bairagi, "ASCII based Even-Odd Cryptography with Gray code and Image Steganography: A dimension in Data Security", http://ijcit.org/jcit_papers/vol-1_no-2/IJCIT-110112.pdf.

[10] N.F. Johnson and S. Jajodia, Steganalysis of Images Created Using current Steganography Software, Proceedings of the 2nd Information Hiding Workshop, April 1998.

[11] S. Vendatraman, A. Abraham and M. Paprzycki, Significance of Steganogrpahy on Data Security, Proceedings of the International Conference on Information Technology: Coding and Computing, 2004.

Sunny Dagar received B.Tech. degree from Rajasthan Technical University, Kota, India in 2011. He is currently working towards M.Tech. degree in school of information technology, CDAC. He published 2 research papers in international journals in the field of algorithms. His research interest includes steganography, cryptography and algorithms.

Vinay Kumar received B.Tech. degree from DCRUST, Murthal, India in 2011. He is currently working towards M.Tech. degree in school of information technology, CDAC. His research interest includes cryptography, steganography and algorithms.

Yogendra Bagoriya received B.Tech. degree from Rajasthan Technical University, Kota, India in 2010. He is currently working towards M.Tech. degree in school of information technology, CDAC, Noida. His research interest includes Image Processing (Optical Character Recognition) and steganography.