ANALYSIS OF LSB BASED IMAGE STEGANOGRAPHY TECHNIQUES

Introduction:

Definition: Steganography is the art and science of writing hidden messages in such a way that no-one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. The word steganography is of Greek origin and means "concealed writing".

But practical implementation of the definition is not feasible, so pragmatic approach would be to make the algorithm as strong as possible. Steganography is most widely formulated in terms of the prisoner's problem where Alice and Bob are two inmates who wish to communicate in order to hatch an escape plan. However, all communication between them is examined by the warden, Wendy, who will put them in solitary confinement at the slightest suspicion of trouble. Specifically, in the general model for steganography, we have Alice wishing to send a secret message M to Bob. In order to do so she ”embeds” M into a cover-object C, to obtain the stego-object S. The stego-object S is then sent through a public channel. The warden Wendy who is free to examine all messages exchanged between Alice and Bob can be passive or active. A passive warden simply examines the message and tries to determine if it potentially contains a hidden message. If it appears that it does, then she takes appropriate action else she lets the message through without alteration. An active warden on the other hand can alter messages deliberately, even though she does not see any trace of a hidden message, in order to foil any secret communication that can nevertheless be occurring between Alice and Bob.

There have been many techniques for hiding messages in images in such a manner that the alterations made to the image are perceptually indiscernible. However, the question whether they result in images that are statistically indistinguishable from untampered images has not been adequately explored. The paper under study describes LSB based Steganography and under what condition can an observer distinguish between Stego-images (Images with a secret message) and Cover-images (Images without any secret message).

This report includes the study and implementation of “ANALYSIS OF LSB BASED IMAGE STEGANOGRAPHY TECHNIQUES by R.Chandramouli and Nasir- Memom” and the problems encountered while understanding and implementing the paper.

Approach:

The paper under study describes LSB based Steganography and under what condition can an observer distinguish between Stego-images (Images with a secret message) and Cover-images (Images without any secret message).

- First we study an algorithm implementing the proposed method.
- Then we study the derivation of the Steganographic capacity that gives an approximation on how many bits can be hidden in an Image, such that steganalyst cannot detect the presence of secret message.
- And finally we discuss the performance of this technique using Numerical Results obtained by considering an example.

There is no particular algorithm for LSB technique, as it depends on how many LSB's you want to use for storing the secret message, and the relative size of the secret message compared to Cover-image.
According to the author of this paper, I implemented LSB based steganography by using one LSB of Cover image in MATLAB and the results are discussed in the below section.

Discussion:

LSB based steganographic techniques either change the pixel value by +1 or -1 or leave them unchanged. This is dependent both on the nature of the hidden bit and the LSB of the corresponding pixel value.

Let $I = \{x_i, \;\; i \in \Omega\}$ where Ω is an index set denote the mean subtracted cover image.
The set R can be partitioned into three subsets A1, A2, and A3, where, $\Omega = \bigcup_{i=1}^{3} \Lambda_i$ and $\Lambda_i \bigcap \Lambda_j = \phi$ for $i \neq j$.
Then, the pixel values in a LSB based stego-image, $I_s = \{y_i, \;\; i \in \Omega\}$can be represented as:

$$y_i = \begin{cases} x_i + 1 & \text{if } i \in \Lambda_1 \\ x_i - 1 & \text{if } i \in \Lambda_2 \\ x_i & \text{if } i \in \Lambda_3 \end{cases}$$

Here the goal of steganalyst is to find if 'I' has any hidden data and its same as finding if A1 and A2 are non-empty sets. If so, then what are the elements of the sets ?
We make the following simplistic but realistic assumptions in order to compute the steganographic capacity $x_i, \;\; i \in \Omega$ is Gaussian distributed with zero mean and variance σ^2 $(i.e., x_i \sim N(0, \sigma^2))$.

The steganalysis process can then be formulated as the following multiple hypothesis testing problem for each $i \in \Omega$:

$$H_j : y_i = x_i + d_i, \quad j = 1, 2, 3$$

where $d_i = -1, 0$ or 1. This means $H_1 : y_i \sim N(1, \sigma^2)$,
$H_2 : y_i \sim N(-1, \sigma^2)$, and $H_3 : y_i \sim N(0, \sigma^2)$.

Here H3 indicates that there is no difference between Cover Image and Stego-Image. This leaves the Steganalyst with detecting H1 and/or H2. So we can safely ignore the case where LSB of Stego Image and Cover Image are same.

Let us suppose the probability of a data bit equal to 1 is 0 < pd < 1 and the probability of a LSB (denoted by li) being 1 is equal to 0 < pl < 1. Assuming the hidden bits and the LSB's are independent of each other, the joint probability,

$$P(d_1 = 1, l_1 = 1, \ldots, d_{|\Omega|}, l_{|\Omega|}) = (p_d p_l)^{|\Omega|}$$

To detect which one the 3 hypothesis is true for each pixel, we use minimum probability of error criteria as cost function of this process. The minimum probability of error detector is maximum a posteriori probability (MAP) detector and the true hypothesis is given by

$$H = \arg\max_j P(H_j) P(y_i | H_j)$$

As y_i is Gaussian the MAP detector becomes,

$$H = \arg\max_j P(H_j) exp \frac{-(y_i - d_j)^2}{2\sigma^2}$$

where dj = 1, - 1, or 0 corresponding to H I , H2, or H3. This gives an estimate of the pixel locations that have been modified by hiding data. As error will be made during the estimation, and let $p_{kj} = P(\text{decide } H_j | H_k \text{ true}),\ j, k = 1, 2, 3.$ denote the error probabilities. The values of p_{jk} depends on the variance of the image and techniques used for the estimation.

We now proceed to design a test for the presence of hidden message in the image. Towards this goal, once the first pass of steganalysis is over the second pass is begun. Here, a second detector combines the output decisions of the first pass. The output of this detector will tell us if there is any hidden data at all (with a certain probability). Let u_i denote the decision of the first detector for pixel i and $M = |\Omega|$

Using the MAP criteria we observe that hidden data is detected if,

$$P(H_1)\prod_{i=1}^{M}P(u_i|H_1)\begin{cases} > P(H_2)\prod_{i=1}^{M}P(u_i|H_2) \text{ and}\\ > P(H_3)\prod_{i=1}^{M}P(u_i|H_3)\end{cases}$$

or

$$P(H_2)\prod_{i=1}^{M}P(u_i|H_2)\begin{cases} > P(H_1)\prod_{i=1}^{M}P(u_i|H_1) \text{ and}\\ > P(H_3)\prod_{i=1}^{M}P(u_i|H_3)\end{cases}$$

Due to symmetry both these have the same probabilities of error, so we can consider only the first case for further study and only the detection of H1 versus H3 is considered because they are statistically closer than H1 versus H2. So the above multiple hypothesis has been simplified to binary hypothesis testing and we have

$$\frac{\prod_{i=1}^{M}P(u_i|H_1)}{\prod_{i=1}^{M}P(u_i|H_3)}\begin{cases} > \frac{P(H_3)}{P(H_1)} & \text{decide Hidden Data}\\ \text{else} & \text{decide No Hidden Data}\end{cases}$$

which gives

$$\prod_{S_1}\frac{P(u_i=1|H_1)}{P(u_i=1|H_3)}\prod_{S_2}\frac{P(u_i=-1|H_1)}{P(u_i=-1|H_3)}\prod_{S_3}\frac{P(u_i=0|H_1)}{P(u_i=0|H_3)}$$
$$\begin{cases} > \frac{P(H_3)}{P(H_1)} & \text{decide Hidden Data}\\ \text{else} & \text{decide No Hidden Data.}\end{cases}$$

This in turn implies,

$$\prod_{S_1}\frac{p_{11}}{p_{31}}\prod_{S_2}\frac{p_{12}}{p_{32}}\prod_{S_3}\frac{p_{13}}{p_{33}}\begin{cases} > \frac{P(H_3)}{P(H_1)} \text{ decide Hidden Data}\\ \text{else decide No Hidden Data}\end{cases}$$

Here S1, S2, S3 denotes the cases where 1,-1 and 0 is detected. Now we denote $P_d = P(\text{decide } H_1|H_1 \text{ true})$, the probability of correction detection. $P_f = P(\text{decide } H_1|H_3 \text{ true})$, denotes the false alarm probability of the detection. We can see that these are functions of |S1| and |S2| and there are 2^{3^M} possible detection rules the second detector can employ. Instead of computing the parameters of the Global detection rule we sacrifice optimality for tractability by making the second detector use J out of M possible detection rules then,

$$P_d = \sum_{k=J}^{M}\sum_{r=0}^{M-k}\frac{M!}{k!r!(n-k-r)!}p_{11}^{k}p_{12}^{r}p_{13}^{M-k-r}$$
$$P_f = \sum_{k=J}^{M}\sum_{r=0}^{M-k}\frac{M!}{k!r!(n-k-r)!}p_{31}^{k}p_{32}^{r}p_{33}^{M-k-r}$$

So the steganalyst need to achieve a given value of Pd and Pf, then the number of bits that can be reliably hidden is obtained by solving the above equations for J.

Algorithm:

Encoding:

1) Read Cover Image into F.
2) Read Secret image or message into G.
3) Convert the images F and G into binary format using dec2bin.
4) Now take the each bit of a pixel from G and store it in the LSB of F.
5) Now convert the binary values of the resulting image(Stego-image) to decimal using bin2dec.
6) The output of bin2dec would be column matrix and so it has to be converted to the size of the image F.
7) Convert the image to Uint8 and write it to some folder. The Stego-image is obtained.

Note: This algorithm stores the secret image into the last bit of the cover image along the Column pixels.

Decoding:

1) Read the Stego-Image into F.
2) Convert F into binary values using dec2bin.
3) Take the LSB of each pixel column wise, and append these bits depending on the depth of the secret message to form pixels of the Secret image.
4) Convert this into Decimal using bin2dec.
5) Convert the column matrix into the size of the secret image.

Note: Step5 can be done successfully only by the prior knowledge of the size of the secret image.

The Algorithms are coded in MATAB and find the files Steg.m for encoding and stegdec.m for decoding, attached. Note that stegdec.m needs the prior knowledge of the size of the secret image to get the correct output. Infact its an advantage, as only the authorized viewer who knows the size of the secret image can view the secret image embedded in Stego-image.

Note: As we are using LSB based Stegano-graphy, the size of Cover image should be atleast 8 times more than the Secret Image.

Results:

The algorithm explained above is coded in MATLAB and the results are shown below:

1-a) Cover Image

1-b) Secret Image

1-c) Stego- Image

1-d) Stego-Image

1-e) Secret Image decoded from Stego-image

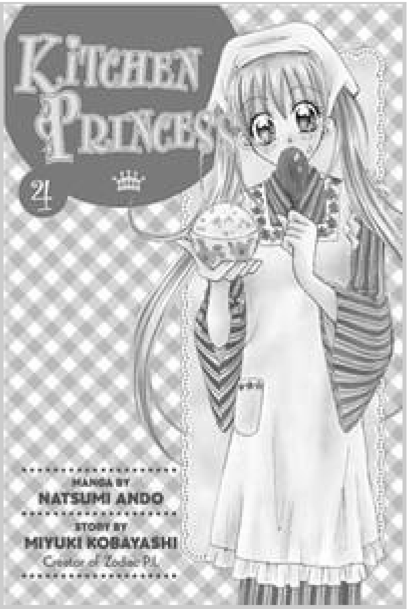
2-a)

2-b)

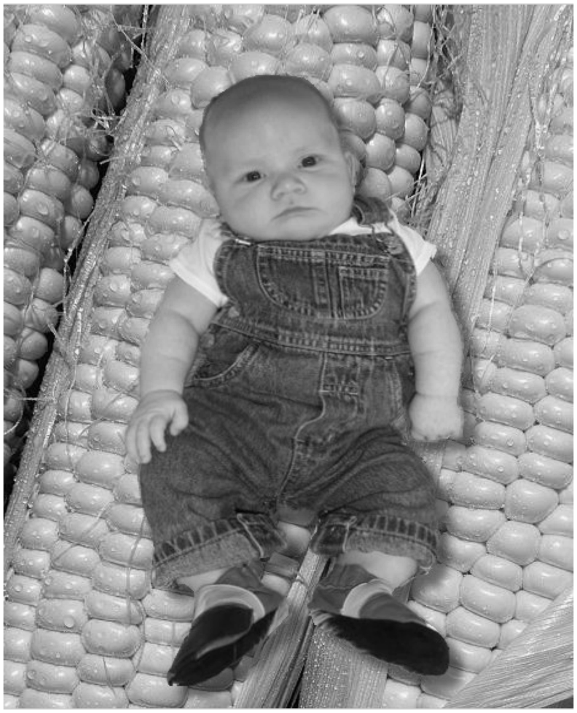
2-c)

2-a), Cover Image 2-b) Secret Image 2-c) Stego - Image

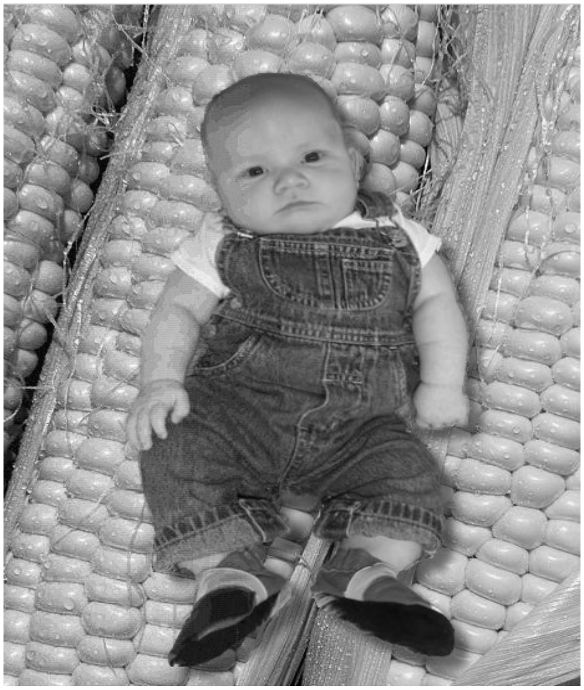
The same algorithm is implemented by using 4 LSB of the Cover Image and the results are shown below:



Secret Image



Cover Image

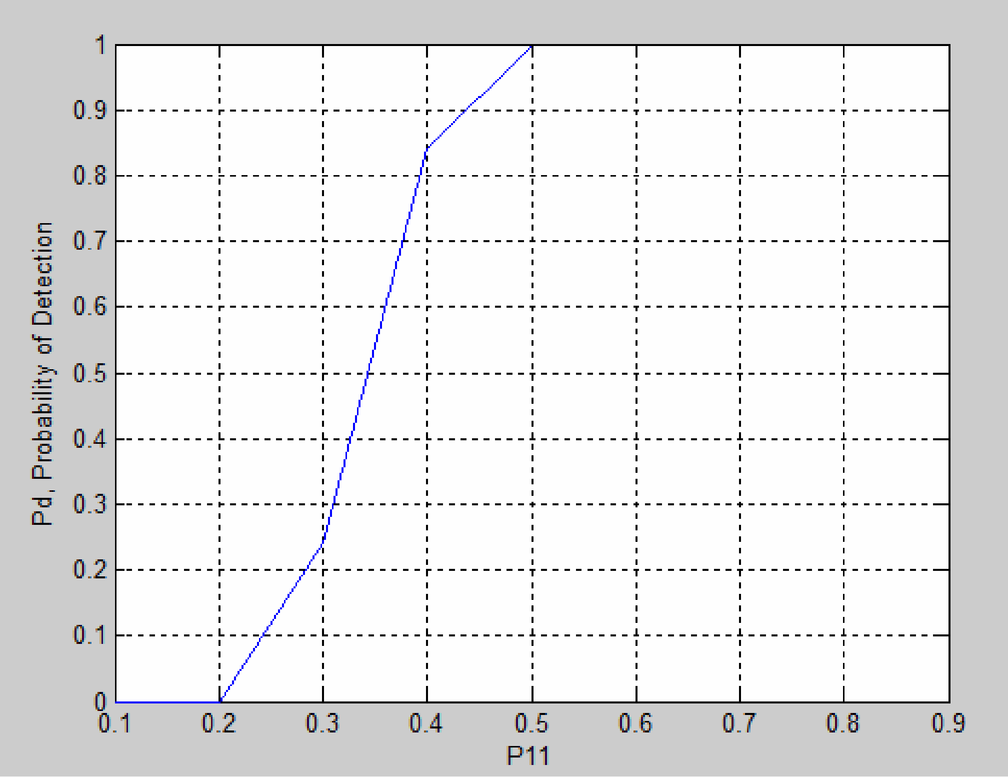


Stego Image

Note: We can see that though 4 bits are replaced the cover image is almost retained but with small stripes indicating the presence of some hidden message.

Now we consider a numerical example when M=64, P12=0.05 and P13=1-(P11+P12) and J= (M+1)/2 and we plot Pd, the probability of correct detection vs P11 by using the relation:

$$P_d = \sum_{k=J}^{M} \sum_{r=0}^{M-k} \frac{M!}{k!r!(n-k-r)!} p_{11}^{k} p_{12}^{r} p_{13}^{M-k-r}$$



In this numerical example, Pd=0.5 when P11=0.33. This means that the steganalyst is forced to make a random guess regarding the presence of hidden data only when P11=0.33. So, in this case, 44 bits can be reliably hidden (assuming half the number of hidden data do not require replacing a LSB) if the detector in the first stage is forced to achieve P11=0.33.

Thus the image property and the strategy of the steganalyst play a role in determining the data hiding capacity for LSB based schemes.

Conclusion:

In this report we have gone through a LSB based technique, implemented the algorithm in MATLAB and derived expressions for arriving at the steganographic capacity of LSB based image data hiding techniques.

List of Tasks performed:

Task	Performed by:	Status
Study	Kalyan Karnati	Completed
MATLAB Code	Kalyan Karnati	Completed
Experiments	Kalyan Karnati	Completed
Documentation	Kalyan Karnati	Completed

Challenges faced during the Study:

- Briefly gone through "R. Chandramouli and N.D. Memon, "A distributed detection framework for watermark analysis" to understand the concepts of Steganography.
- Derivation for finding the Capacity, includes many Random processes concepts and I've gone through "Probability, Statistics and Random Processes for Electrical Engineering by Alberto Leon-Garcia" and "H. Poor, An introduction to signal detection and estimation" for understanding MAP estimate and other concepts.
- Have gone through "N.F. Johnson, Z. Duric, and S. Jajodia, "Information hiding: Steganography and watermarking - attacks and countermeasures" to learn more about the LSB algorithm and the attacks.

References:

"Analysis of LSB based Image steganographic techniques" by R. Chandramouli and Nasir Memon

N.F. Johnson, Z. Duric, and S. Jajodia, "Information hiding: Steganography and watermarking - attacks and countermeasures," Kluwer Academic Publishers, 2000.

R. Chandramouli and N.D. Memon, "A distributed detection nframework for watermark analysis," Proc. ACM Multimedia and Security Workshop, 2000.

"Probability, Statistics, and Random Processes for Electrical Engineering" by Alberto Leon-Garcia.

H. Poor, An introduction to signal detection and estimation, Springer Verlag, 1994.