

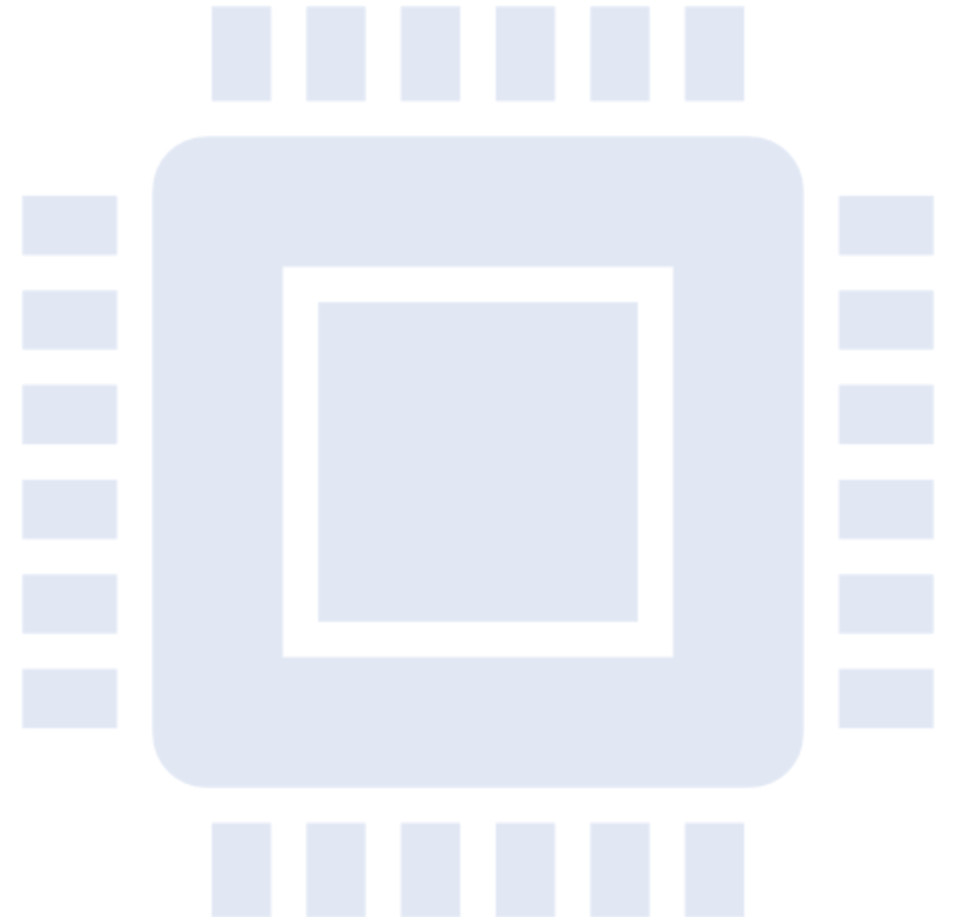
Least Bit Steganography (LSB) With Higher Data Volume Transmission Inside Images

Anish Sachdeva

DTU/2K16/MC/013

Cryptography and Network Security (MC-407)

Dr. Rohit Kumar



Details

- Project
Link: <https://github.com/anishLearnsToCode/lb-image-steganography>
- Student Name: Anish Sachdeva
- Supervisor: Dr. Rohit Kumar
- Subject: Cryptography and Network Security (MC-407)
- Project Title: Least Bit Steganography (LSB) With Higher Data Volume Transmission Inside Images

Steganography

- Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. The word steganography comes from Greek steganographia, which combines the words steganós, meaning "covered or concealed", and -graphia meaning "writing".

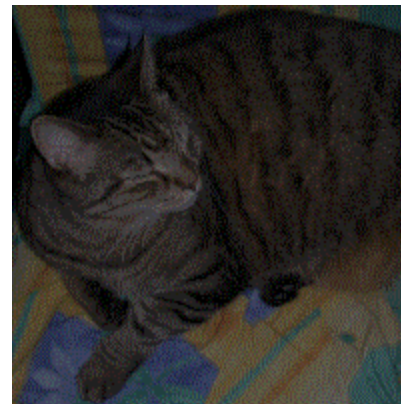
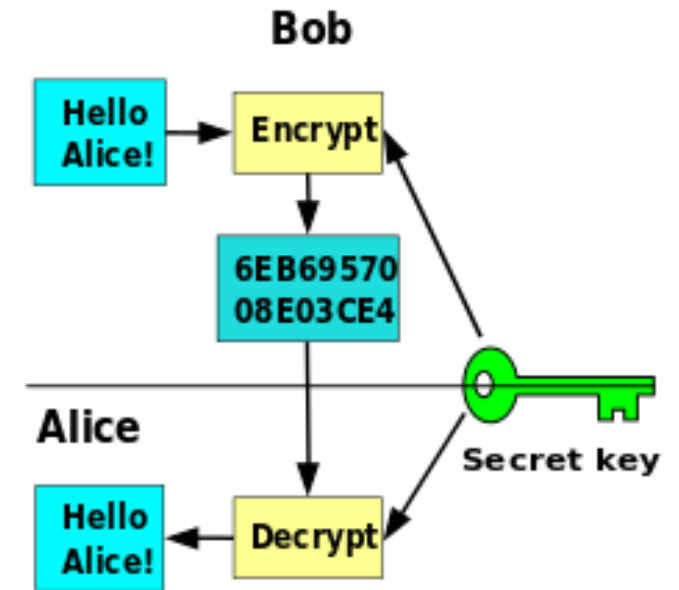


Image of a tree where another second image is hidden in the least 2 significant bits of the image.

Cryptography

Cryptography, or cryptology is the practice and study of techniques for secure communication in the presence of third parties called adversaries. More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography.

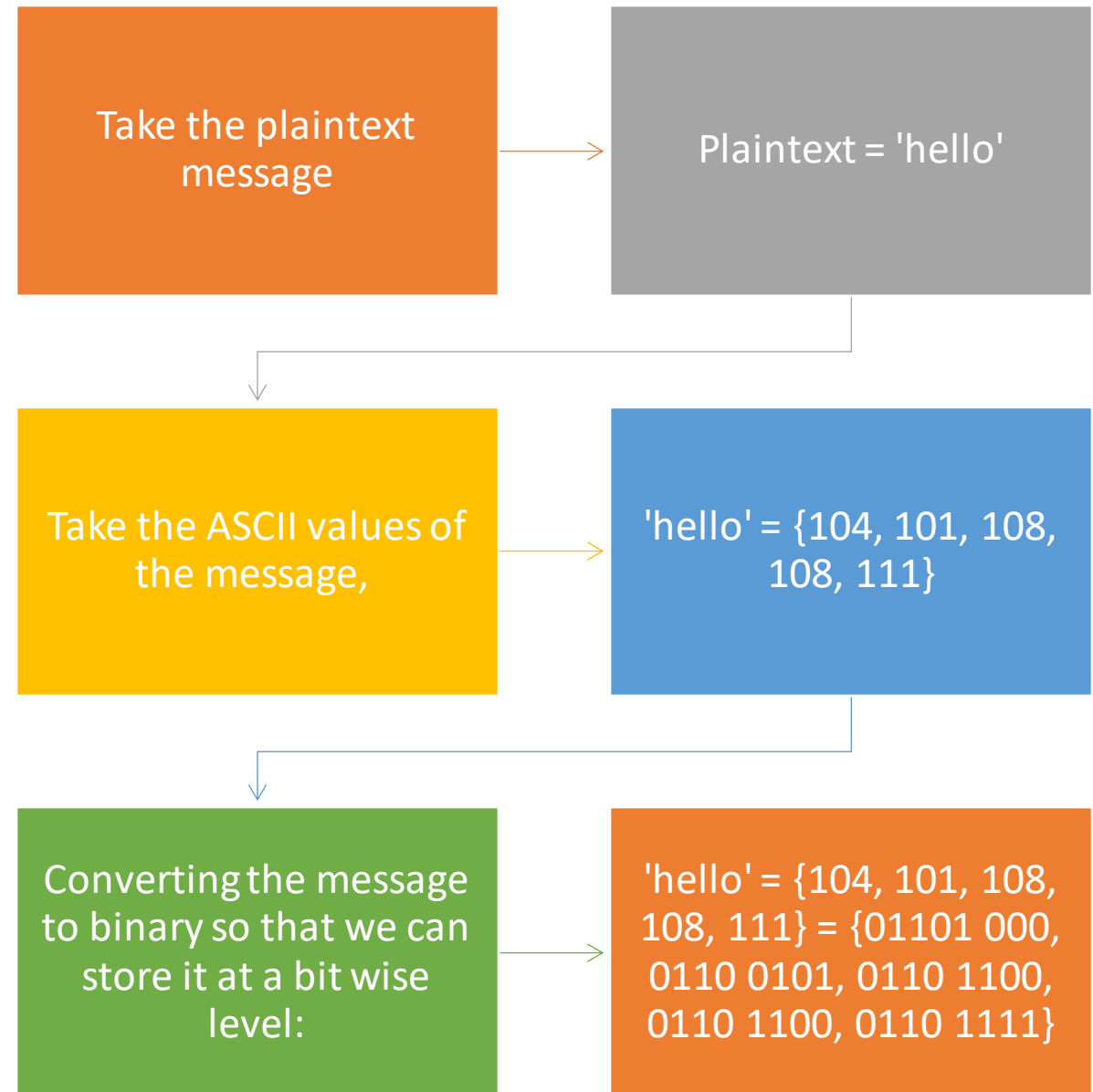


Kirchoff's Law

- The system must be practically, if not mathematically, indecipherable;
- It should not require secrecy, and it should not be a problem if it falls into enemy hands;
- It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will;
- It must be applicable to telegraph communications;
- It must be portable, and should not require several persons to handle or operate;
- Lastly, given the circumstances in which it is to be used, the system must be easy to use and should not be stressful to use or require its users to know and comply with a long list of rules.



Least Significant Bit (LSB) Steganography In Images



- We will now create a bit stream which we need to embed in the image
- $S = \{01101\ 000, 0110\ 0101, 0110\ 1100, 0110\ 1100, 0110\ 1111\}$
- We will now take a RGB Image of dimensions $(w, h, 3)$ and extract the 3 channels R, G and B
- We will convert each channel which has dimension (w, h) into a row vector by unravelling the values and obtain a single vector of dimension $(wh, 1)$
- We will then store the bit stream S into the channels B, R and G by changing the least significant bit in the channels.



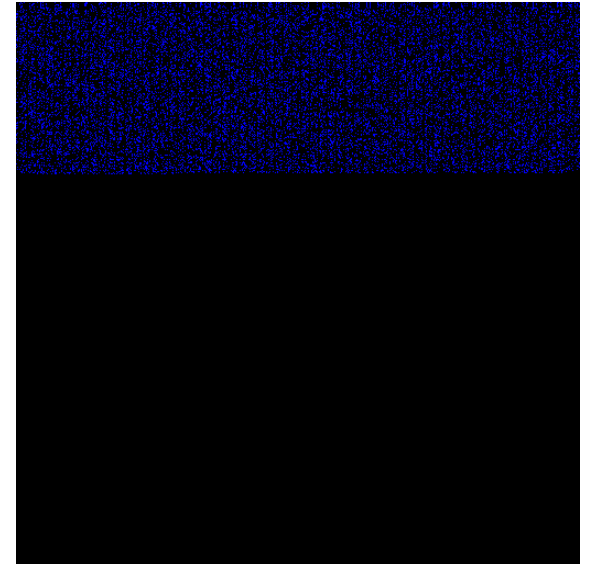
Results of LSB Steganography



Standard Lenna Image



Lenna Image with hidden Plaintext of length 10,000 characters



Difference Between the original and Steganographic Image, where the pixels modified are the Blue Channel pixels only.

An abstract composition of various geometric shapes. In the top left, a green-outlined triangle points towards the top right. To its right is a solid blue circle. Below the triangle is a blue-outlined circle. In the center is a large orange semi-circle. To the right of the semi-circle is a vertical yellow dashed line. In the bottom left is a large solid orange circle. To its right are four short, curved yellow dashed lines. In the bottom right is a green-outlined square.

- Current number of bits offered by the LSB Method
- $\#bits(LSB) = 3wh$
- Number of characters we can store in the Image (or Data Volume of Image) given that a single character will require a byte (8 bits) of space
- $\#characters(LSB) = \text{floor}(3wh/8)$
- So, maximum length of plaintext message we can store is $|P| = \text{floor}(3wh/8)$
- We will need $\lg(\text{floor}(3wh/8))$ bits to store the length of the message with the message.
- So total message length $|M| = |P| + \lg(\text{floor}(3wh/8))$

Image To Grayscale

- There are several techniques of converting an image to grayscale. Simplest is to take average of 3 RGB channels.
- $G = (R + G + B) / 3$



Better Method For Grayscale

- Our eyes do not perceive the channels Red, Green and Blue equally, so the standard method of taking grayscale is a weighted average
- $G = 0.2126 * R + 0.7152 * G + 0.0722 * B$



Original Color Image



Grayscale Image using
averaging




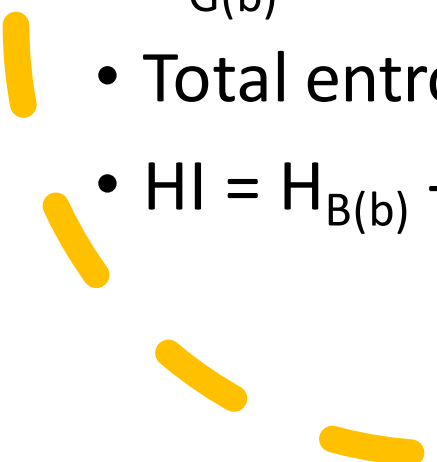
Grayscale Image Using
Weighted Average

Entropy For Higher Bit Image Hiding

- Using the weighted average gives considerably better results
- This shows that not all channels contribute equally to an image perception and contribute different amounts ~ 21.26%R, 71.52%G and 7.22% B
- If we store b bits of data in either the Red, Blue or green channels we will have different information entropy
- Shannon's Entropy is defined as :

$$H = - \sum_i p_i \log_2(p_i)$$

- When storing the least significant bit the probability that we will cause any change is:
- $P_b = 2^0 / 2^8 - 1 = 3.92 \times 10^{-3}$, but the probability weights for perception are not the same, so probability for changing one bit per channel becomes
- $P_B = 0.0722 * P_b$
- $P_R = 0.2126 * P_b$
- $P_G = 0.7152 * P_b$
- The Entropies of a single bit represented in these channels (LSB) is:
- $H_B = -P_B \log_2(P_B) = 3.33 \times 10^{-3}$
- $H_R = -P_R \log_2(P_R) = 8.52 \times 10^{-3}$
- $H_G = -P_G \log_2(P_G) = 2.37 \times 10^{-2}$

- 
- So entropy if we store data in the entire image in the least significant bits will be:
 - $H_{B(b)} = wh\ 3.33 \times 10^{-3}$
 - $H_{R(b)} = wh\ 8.52 \times 10^{-3}$
 - $H_{G(b)} = wh\ 2.37 \times 10^{-2}$
 - Total entropy with LSB Steganography for entire Image
 - $H_I = H_{B(b)} + H_{R(b)} + H_{G(b)} = wh\ 3.533 \times 10^{-2}$
- 

- If we store data in the 2nd least significant bit, the bit change probability for that will be: $P_b = 2^1 / 2^8 - 1 = 2/255$.
- Respectively the probabilities will be:
- $P_B = 0.0722 * P_b = 5.66 \times 10^{-4}$
- $P_R = 0.2126 * P_b = 1.66 \times 10^{-3}$
- $P_G = 0.7152 * P_b = 5.60 \times 10^{-3}$
- The entropies for a single secondleast significant bit will be:
- $H_B = - P_B \log_2(P_B) = 6.10 \times 10^{-3}$
- $H_R = - P_R \log_2(P_R) = 1.53 \times 10^{-2}$
- $H_G = - P_G \log_2(P_G) = 4.19 \times 10^{-2}$

- The total Entropy of the Image if we store in the second Least Significant bit will be:
- $H_I(b_2) = wh (H_B + H_R + H_G) = wh 6.33 \times 10^{-2}$
- If we use both least significant bits to store the data, the message entropy will be:
- $H_I(b_1 + b_2) = H_I(b_2) + H_I(\text{lsb})$
- $= wh (6.33 \times 10^{-2} + 3.533 \times 10^{-2})$
- $= wh 9.863 \times 10^{-2}$

Conclusion

- We can easily store messages inside Images using the last 2 significant bits without changing the entropy of the Image drastically.
- Storing an encrypted message inside the Image makes it very secure as then the attacker will first need to figure out that:
 - A) There is a message stored inside the image
 - B) The message bits are stored using which algorithm and
 - C) Then decrypt the extracted message using the secret key

Encoding the channels in a particular sequence can minimize entropy and store more data with lower entropy.