Lab-1 LSB based Image Steganography

Himanshu Goyal 17CS02011

Aim:

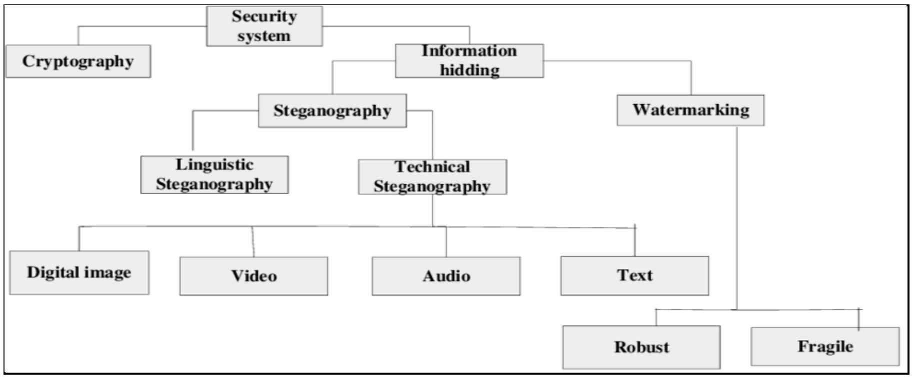
- Design and implementation of an algorithm to hide secret messages into the cover image by only using LSB bits of pixels.

Theory:

Cryptography: is a method to hide information, (or: authenticate information, authenticate users, exchange keys, etc.). It is based on keys and applying keyed transformation to data and the data changes to encrypted data so that with the key one can reverse the transformation (or check authenticity, etc). There are many primitives in cryptography: symmetric encryption where key for encryption and decryption is the same, public-key cryptography where the keys above are different. There are digital signatures, MAC-ing information, and so on many other methods. RSA is a primary example of public key, Diffie-Hellman is a primary example of key exchange, AES is a primary example of symmetric key ciphers etc.

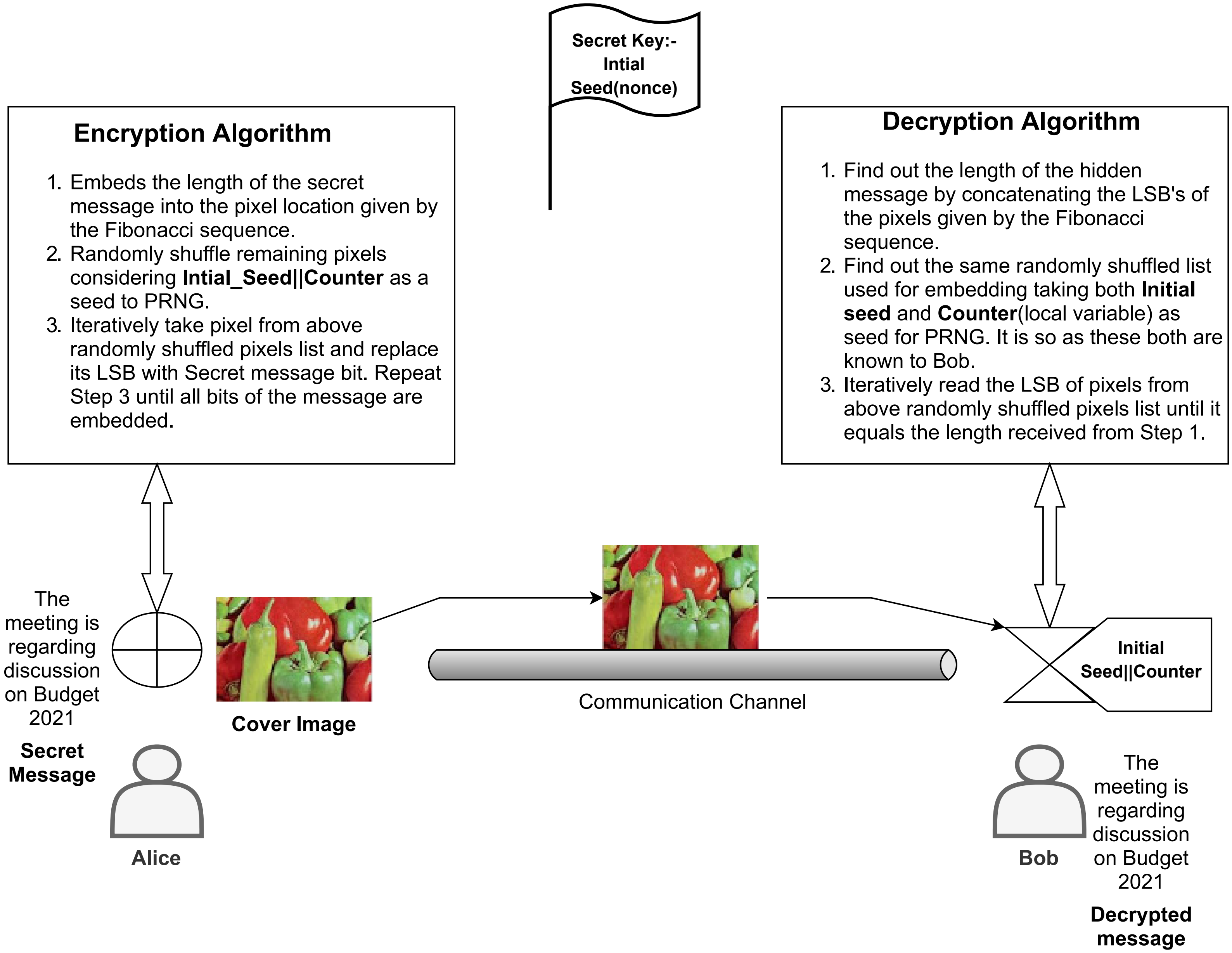
Steganography: is a method to modify text or other form of information transforming the source to a target file (say text) which encodes a message but without knowing the key the text simply looks like a normal text (while secretly embedding a message). Say the key tells you to take every 5th word and look at its third letter and this is the hidden message.

Watermarking: is a method to embed an identifier in some hidden way in a file (typically in non text files) so that if you have the key you can authenticate the existence of the watermark. The method is typically hidden to the naked eye, but may be detectable. It's goal is authentication of the source of the data.



Self Proposed Algorithm: Hiding Secret Message into a Cover Image

Assumption: Only LSB should be used for manipulation of any information



Secret Key:-
Intial
Seed(nonce)

Encryption Algorithm

1. Embeds the length of the secret message into the pixel location given by the Fibonacci sequence.
2. Randomly shuffle remaining pixels considering **Intial_Seed||Counter** as a seed to PRNG.
3. Iteratively take pixel from above randomly shuffled pixels list and replace its LSB with Secret message bit. Repeat Step 3 until all bits of the message are embedded.

Decryption Algorithm

1. Find out the length of the hidden message by concatenating the LSB's of the pixels given by the Fibonacci sequence.
2. Find out the same randomly shuffled list used for embedding taking both **Initial seed** and **Counter**(local variable) as seed for PRNG. It is so as these both are known to Bob.
3. Iteratively read the LSB of pixels from above randomly shuffled pixels list until it equals the length received from Step 1.

The meeting is regarding discussion on Budget 2021

Secret Message

Cover Image

Communication Channel

Initial Seed||Counter

Alice

Bob

The meeting is regarding discussion on Budget 2021

Decrypted message

Counter: It denotes the step of the meesage transmssion. i.e. Counter=k(can be 1 also) when Alice is sending message first time and then increments sequentially on next transmissions.

Experiment(s) and Result(s):

Cover_1.png

Secret Message Length	PSNR (**in dB**)
1 byte	86.3
5 byte	81.8
500 byte	63.3
1 KB	60.3
4 KB	54.2
6 KB	52.5

Cover_2.png

Secret Message Length	PSNR (**in dB**)
1 byte	90.3
5 byte	85.8
500 byte	66.8
1 KB	63.8
4 KB	57.7
6 KB	56.1

Conclusion:

The practical difference between steganography, cryptography and watermarking has become quite meaningful after performing this experiment. The use of PRNG, a well known tool in the security domain, has helped here to randomly select the pixels for hiding secret messages into the cover image. The randomisation introduced in the self-proposed algorithm makes the job of adversary difficult to decipher in comparison to its counterpart available deterministic algorithms.

In Future, I would like to extend this proof-of-work algorithm in order to get higher PSNR and would also test it on other multimedia objects.