

# LSB Steganography

# Description

- Encode a user given secret file inside a .bmp image file to get a steganographed file.
- Also encode a pre defined magic pattern and user given password.
- For decoding / extracting the secret file, get magic pattern and password from user

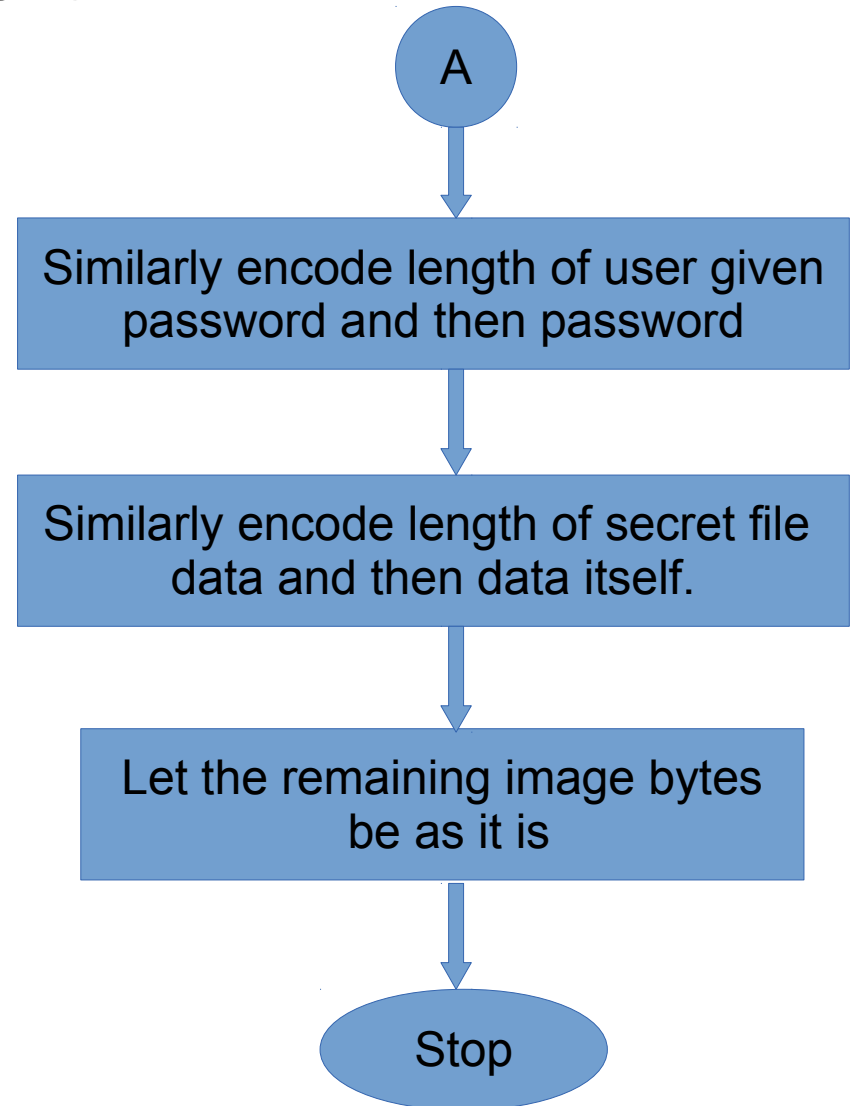
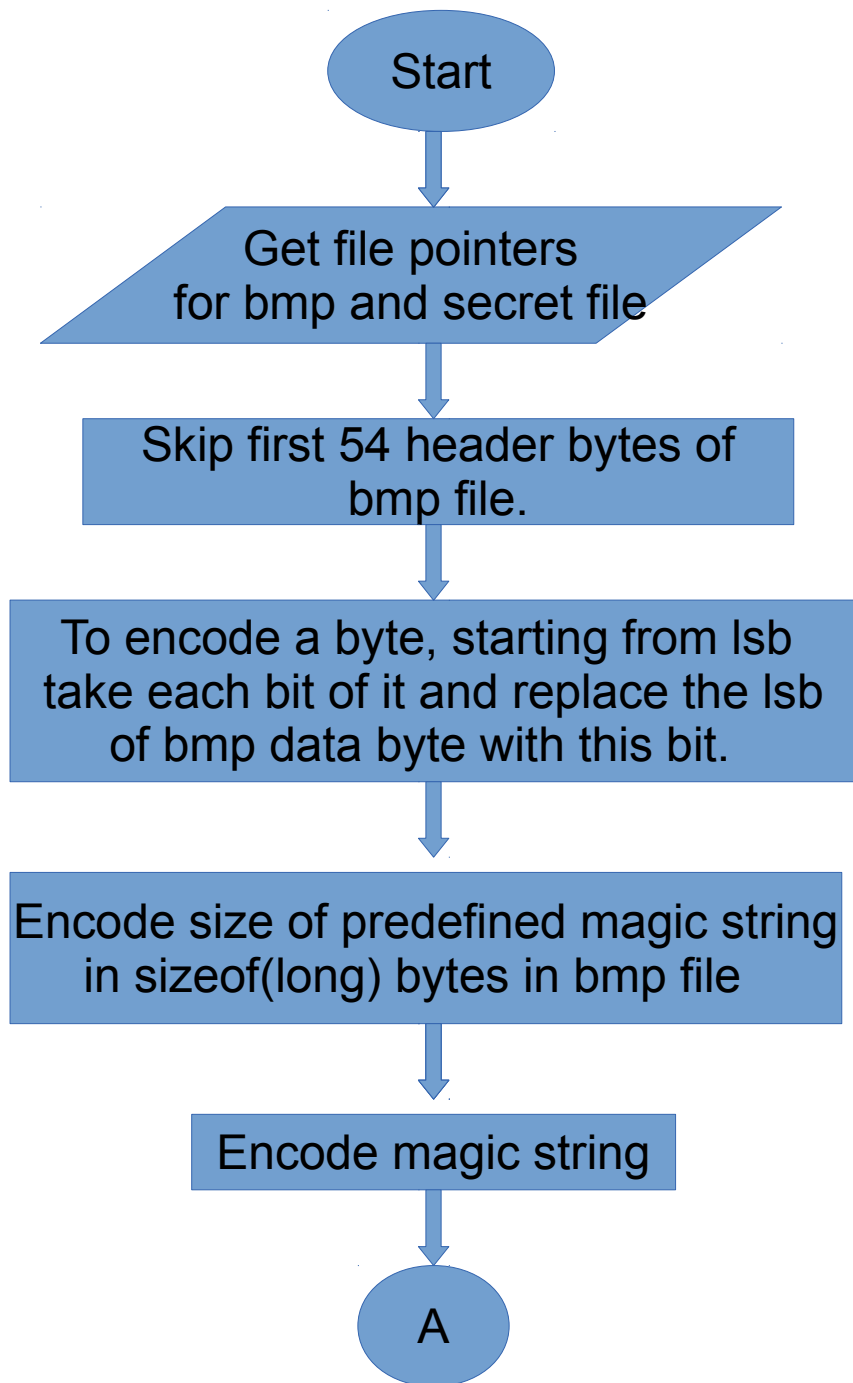
# Algorithm

- Step 1: Get file pointer for .bmp image file and source secret file.
- Step 2: Skip first 54 header bytes of image file.
- Step 3: To encode a data byte, take one bit at a time starting from lsb.
- Step 4: Replace the lsb of image byte (starting from 55<sup>th</sup> byte) with the bit obtained.
- Step 5: First encode size of predefined magic pattern in sizeof(long) bytes.
- Step 6: Then encode the predefined magic pattern.
- Step 7: Encode size of user entered password
- Step 8: Encode password.

# Algorithm

- ▯ Step 9: Encode size of extension of secret file.
- ▯
- ▯ Step 10: Encode extension of secret file.
- ▯ Step 11: Calculate and encode size of secret file data.
- ▯ Step 12: Encode secret file data.
- ▯ Step 13: Keep the remaining bytes of .bmp image as it is.

# Flowchart



# Advantages

- Encrypting Secret data.

# Disadvantages

- LSB of each data byte of .bmp image is lost and cannot be recovered back.