snake-nography Design

Gary. K

4/11/2022

Github Link

Initial Ideas

- Possible implementation of GUI using tkinter for practice
- Implementation of a debug mode
- Use argparser for the first time to practice
- Support for lossless image formats like PNG and bmp
- Simple encryption with fixed key or user input
- Prompt if the cover image is not big enough

Cloaking Images

input image can be of any format since all the bits are converted dierectly. i will open the file as binary and read it. subsituting the bits in bytes is done via matching the rows of the hidden image with the cover image for the easier extracting.

at the begining of the script the the file that needs to be hidden is read as bin file and the data inside of is stored into a variable. then that variable is multiplied by 3 (color depth 3 for RGB) to find how many pixels I need to change on the cover image. the number of pixels then are devided by the width of the cover image to see how many rows of pixels I need to change to correctly hide the secret image.

then a new numphy array is created that has values added to it byte by byte containing data from the cover image and single bit flipped from the secret image.

the numpy array is used to create a image in a lossless fromat (PNG, bmp, etc ...) using cv2 library for image processing.

below is the bitmap header sections for reference:

Bitmap File Structure			
Block	**Field**	**Width**	**Description**
BITMAPFILEHEADER **Fields:** 5 **Width:** 14 bytes	FileType	2 bytes	A 2 character string value in ASCII. It must be 'BM' or '0x42 0x4D'
	FileSize	4 bytes	An integer (unsigned) representing entire file size in bytes (number of bytes in a BMP image file)
	Reserved	2 bytes	To be utilized by an image processing application. Initialized to '0' integer (unsigned) value.
	Reserved	2 bytes	To be utilized by an image processing application. Initialized to '0' integer (unsigned) value.
	PixelDataOffset	4 bytes	An integer (unsigned) representing the offset of actual pixel data in bytes.
BITMAPINFOHEADER **Fields:** 11 **Width:** 40 bytes	HeaderSize	4 bytes	An integer (unsigned) representing the size of the header in bytes. It should be '40' in decimal.
	ImageWidth	4 bytes	An integer (signed) representing the width of the final image in pixels.
	ImageHeight	4 bytes	An integer (signed) representing the height of the final image in pixels.
	Planes	2 bytes	An integer (unsigned) representing the number of color planes. Should be '1' in decimal.
	BitsPerPixel	2 bytes	An integer (unsigned) representing the number of bits a pixel takes to represent a color.
	Compression	4 bytes	An integer (unsigned) representing the value of compression to use. Should be '0' in decimal.
	ImageSize	4 bytes	An integer (unsigned) representing the final size of the compressed image. Should be '0' in decimal.
	XpixelsPerMeter	4 bytes	An integer (signed). Should be set to '0' in decimal to indicate no preference of the target device.
	YpixelsPerMeter	4 bytes	An integer (signed). Should be set to '0' in decimal to indicate no preference of the target device.
	TotalColors	4 bytes	An integer (unsigned) representing the number of colors in the color pallet.
	ImportantColors	4 bytes	An integer (unsigned) representing the number of important colors. Ignored by setting '0' in decimal.
COLOR TABLE **Fields:** 4 x entries **Width:** 4 x entries	Red	1 bytes	An integer (unsigned) representing Red color channel intensity.
	Green	1 bytes	An integer (unsigned) representing Green color channel intensity.
	Blue	1 bytes	An integer (unsigned) representing Blue color channel intensity.
	Reserved	1 bytes	An integer (unsigned) reserved for other uses. Should be set to '0' in decimal
PIXEL DATA			An array of pixel values with padding bytes. A pixel value defines the color of the pixel.

semi-optional

Extracting the Image

I will use cv2 library to read the data on a userinput file. then a list is made from those data and schemed through, grabbing the last bit of them and putting them in another list. then I will use a loop to compile these bits one by one and putting them in groups of 8 to create a byte. then bytes are turned into hex fortmat and writted to the new file.

Encryption and Decryption

i will use a simple bit XOR for encryption. user input key charecters ASCII value are added to eachother and then made smaller than a byte using modulus. if user wants to use encryption then each byte is XOR'ed with the key byte to create a new byte. same fucntion adn formula can be used with the same key to decrypt the file.

Refrences

https://medium.com/sysf/ bits-to-bitmaps-a-simple-walkthrough-of-bmp-image-format-765dc6857393

https://en.wikipedia.org/wiki/BMP_file_format

https://www.garykessler.net/library/fsc_stego.html

https://www.section.io/engineering-education/steganography-in-python/
https://samsclass.info/124/proj14/pxor.html

https://docs.python.org/3/library/argparse.html#metavar

https://stackoverflow.com/questions/5603364/how-to-code-argparse-combinational-options-in-python