Catherine Yeo, Sharvani Jha
5/25/2016
Number Theory

Visual Cryptography

ABSTRACT

This report briefly describes a 2-bit encryption program. A photo consisting of only black and white pixels (two bits) is inputted into the program, and two images of seemingly random noise will be generated. The two images could be layered to create ("decrypt") the original inputted image using XOR.

PROCEDURE

There are two images used in this process of visual cryptography. The first image is made up of random pixels that are the same size and shape of the original image (see Image 1).

Then we must create the second image, in which the pixels are still the same size and shape of the original image. Using exclusive-or ("XOR"), we can get the appropriate second encrypted image. Specifically, call the sum of the RGB values of the original image pixel A, and the sum of the RGB values of the random noise image pixel B. If A = B, then A XOR B = 0 no matter what, so the new pixel will be black. If A is not equal to B, then A XOR B = 255*3, so the new pixel will be set to white.

In reverse, the two images can then be also combined using an XOR to decrypt the image as the original image.

EXAMPLE 1: 2-Bit Encryption

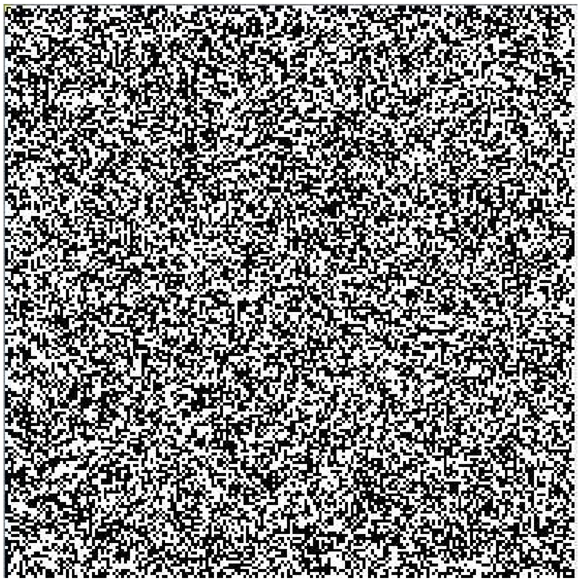

Image 1: First encrypted image (random noise)

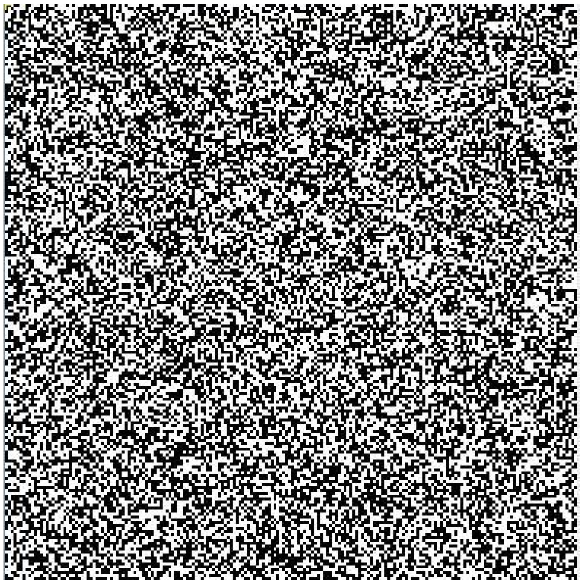

Image 2: Second encrypted image

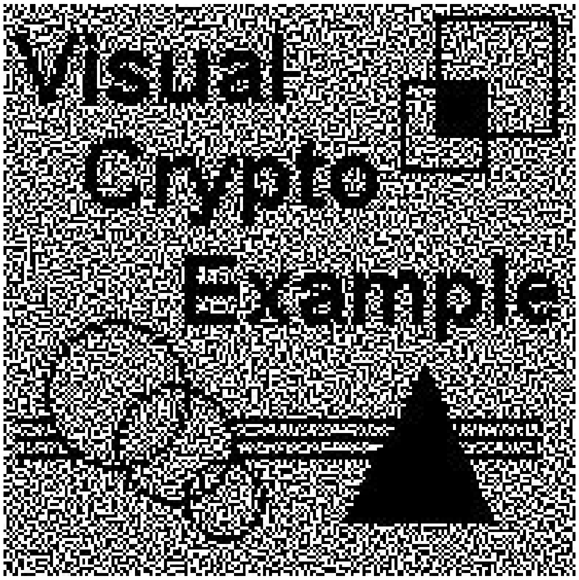

Image 3: Original Image[1]

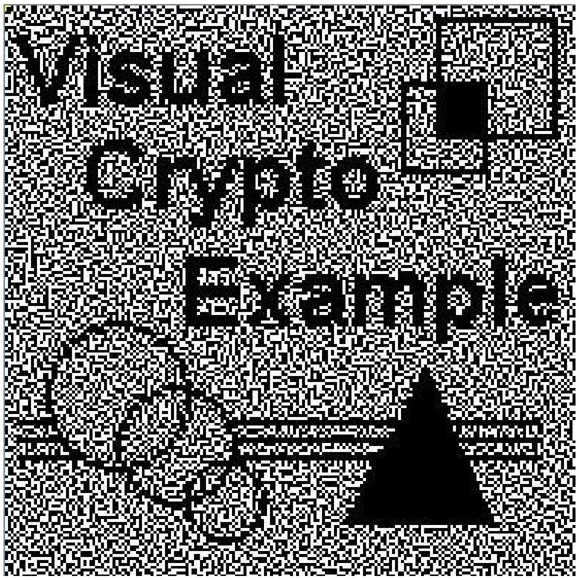

Image 4: Decrypted image

[1] Source: https://antoniuscpilkom.wordpress.com/lecture-notes/cryptography/visual-cryptography/

EXAMPLE 2: 2-Bit Encryption

Another example of this process is the encryption implemented on an image of Mickey. However, this method did not work as well due to Mickey's distinct curves and thick black outline. A faint outline of Mickey could be discerned in Image 6.

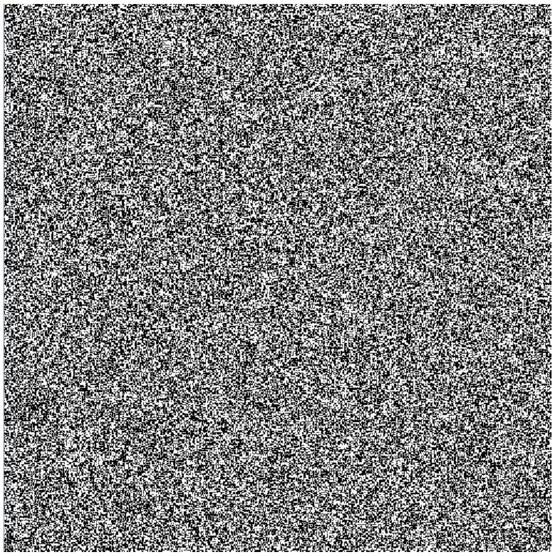

Image 5: First Encrypted Image

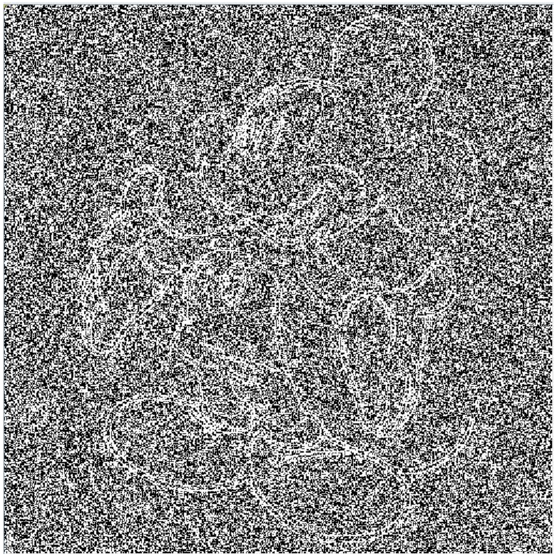

Image 6: Second Encrypted Image

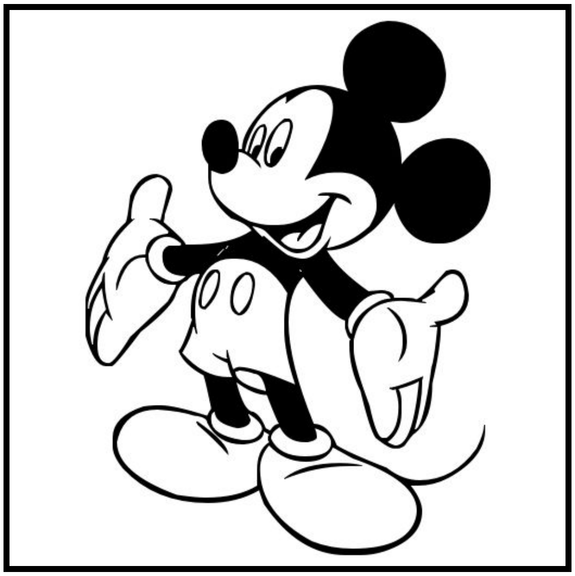

Image 7: Original Image[2] (with a frame here)

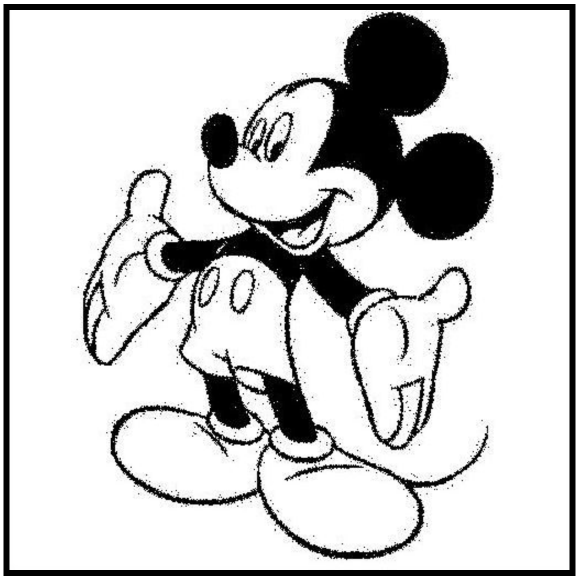

Image 8: Decrypted Image

[2] Source: http://images.clipartpanda.com/mickey-mouse-black-and-white-mickey-mouse-37-512.jpg

EXAMPLE 3: 3-Bit Encryption

For more effective results, we can use three encrypted images instead of just two. Images 9 and 10 are both randomized, while Image 11 is retrieved through multiple uses of XOR. This method can also be used to make a better 2-bit encryption by taking the XOR of Images 9 and 10 as one encrypted image, and Image 11 as the second.

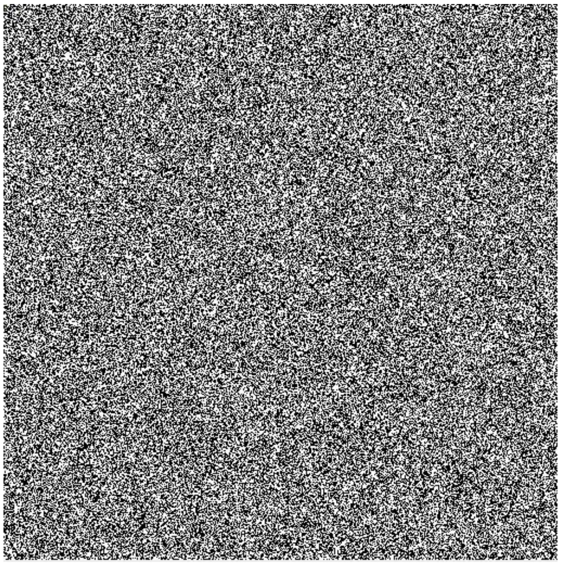

Image 9: First Encrypted Image

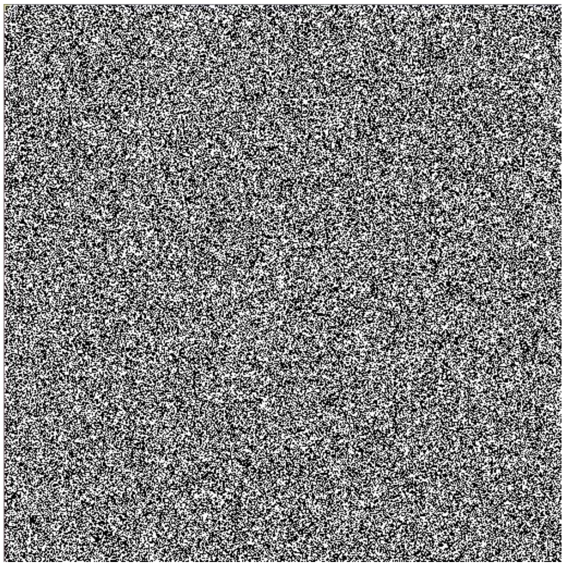

Image 10: Second Encrypted Image

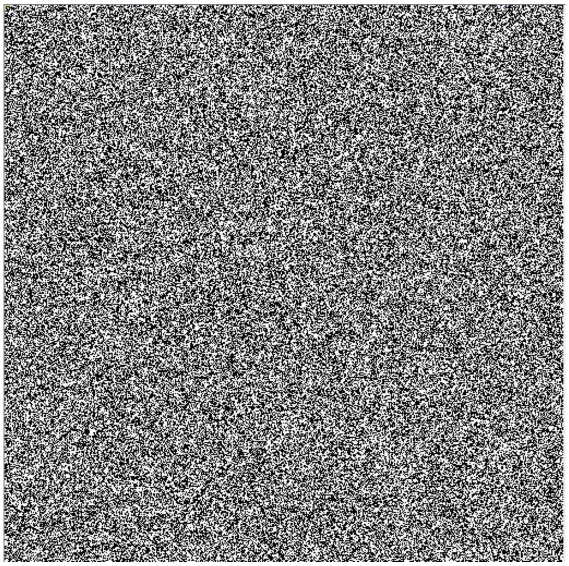

Image 11: Third Encrypted Image

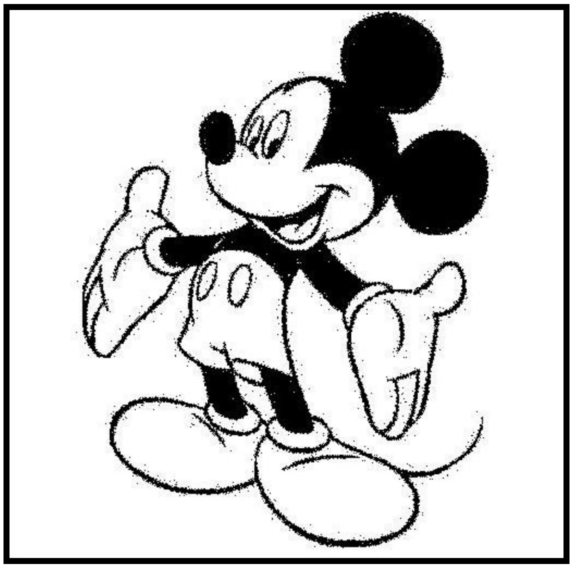

Image 12: Decrypted Image

CONCLUSION

This method could be generalized to be used for 3-bit, 4-bit, 8-bit, etc. Instead of only generating one image consisting of random noise, more could be generated and a complicated XOR process would be implemented in order to generate the correct image. For 3-bit encryption (see *Example 3*), instead of just taking the XOR of two images, conduct it on three. Furthermore, we observed that the greater the number of bits there are, the higher the image resolution is and the more effective the encryption process is.