Raymond Yuan

Mr. Friedman

Linear Algebra

My independent study project involved an individual inputting text and receiving an encrypted version of the text with an image with a key hidden inside of it. I tried multiple variations in which the key was printed directly, surrounded by noise, and finally hidden within a plane of the image (I chose to hide it in the blue plane in the RGB array of the image). I began by printing the key and directly giving it to my friend who would have the decryption function, then hiding the key in a larger matrix of noise, and then finally hiding the key in the image. I have other variations in which I hide both message and key within image as well as giving the user the option to choose what image to encrypt. However, the one I have provided is the cleanest and fastest version.

The method through which I ended up hiding the encryption key was in the image because if anyone were to intercept it, every blue value would be similarly distorted with noise, thereby rendering the key completely invisible. However, a downside to this key storage method, hides the key beginning on the same pixel every time, so if an attacker were to know where the key began it would be easier to extract. Yet, this method still works fairly and would difficult for an attacker to break. Fortunately, in this case, what I was trying to do coincided with what I actually did, and I believe it worked fairly well. I actually have created multiple version of this encryption as tests and for fun (encryption where both text and key are both hidden in the image, different pictures, and completely white images with a distorted blue value). A limitation, however, is the size of the image (limits the volume of text able to be translated and the size of the key for the variation in which the message is hidden in the image). To remedy this one can import a much larger image, but increase encryption time.