ECE 404 HW 5

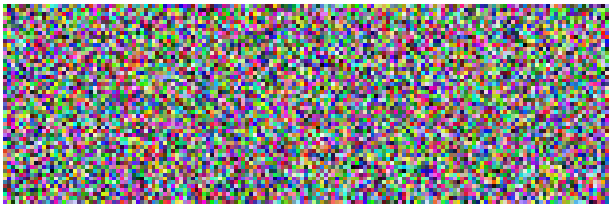
Nimal Padmanabhan

February 21, 2023

1 Programming Problem

For this programming problem, we were tasked to extend our implementation of the AES-256 bit encryption and decryption algorithms from Homework 4 to create the ANSI X.931 pseudorandom number generator and encrypting a ppm image file using the counter (CTR) mode in AES. For part 1, the X.931 function takes in three parameters: a 128-bit date-and-time `BitVector` object denoted by `dt`, a key file name denoted by the string variable `key_file`, and 128-bit seed value `BitVector` object denoted by `v0`. We first obtain the key from the key file through simple file reading operations. Next, we generate the substitution tables that will help us generate the keywords and ultimately the round keys for the AES algorithm. As noted in the diagram in Section 10.6, there are 3 EDE blocks, which stands for encrypt-decrypt-encrypt for the 3DES algorithm. We replace the DES components with AES encryption since the round keys will be generated once, but the XORing of components remains the same. Since we are generating three numbers, the loop will run 3 times and the seed value `v0` will change each after each iteration.

For encrypting the ppm file in Part 2, the process is very similar except we now have read and write in binary mode since it is an image file and we are encrypting the image using the counter (CTR) mode. The function `ctr_aes_image` takes in `iv`, a 128-bit initialization vector that is a `BitVector` object, the input image file `image.ppm`, the encrypted image file `enc_image.ppm`, and the name of the key file `keyCTR.txt`. After obtaining the ppm headers from the input image file, we create our initialization vectors as a list of integers, which makes it easier to add 1 to each vector as shown in the diagram in Section 9.5.5 in Lecture 9. Using the list of initialization vectors, we go through each block (128-bit chunks) to get the plaintext block, encrypt the initialization vector as a `BitVector` object with the key, and XOR with the plaintext block to produce our ciphertext block, which will be written to the output ppm file each iteration. The main difference between encrypting the helicopter image in AES as opposed to DES is that you could still see the outline of the helicopter in DES unlike AES, where there is no outline of the helicopter.



The encrypted ppm file after AES as a png file.