# 

# SI430 Project: Modes of Operation

Name: Chris Paris

Alpha: 264845

Submission  
  
Part 2 submission: \_X\_ On time \_\_ Late (-5 pts)

Full submission: \_\_ Early (+5 pts) \_X\_ On time \_\_ Late (-10 pts)

Overall penalty/reward: \_\_\_0\_\_\_\_\_ pts

Honor

I didn’t discuss the project with any other student.

I didn’t help any other student.

I didn’t show my code to any other student.

I didn’t copy another other student’s code.

I didn’t look at the online resources directly related to the project solutions.

Initial: \_\_CPP\_\_\_\_\_

External source   
(E.g., I referred to this site for xxx)

Challenges

(E.g., it was difficult to figure out how to xxx)

What I learned

(E.g., I learned that xxx.)

Feedback to the instructor

(E.g., This was cool. This was too much. It would have been better if …)

# Part 1

Uses Cryptodome.AES’s encrypt and decrypt functionality for ECB.

For fix\_bmp\_header, copies bmp header bytes and overwrites target file’s data at that length.

# Part 2

Same as P1, but for CTR and CBC

# Part 3

Pad uses the block size minus the length of the last block and pads that byte that many times. So I take the total length mod blk\_size, subtract it from blk\_size, then add it that many times. Unpad gets the last byte, checks for that hex digit and converts it to an integer. It then loops through that many times, each time checking the last byte to see if there is an error and, if no error, removes the last byte until the original unpadded message is all that remains.

# Part 4

Solution: b'known plaintext atk CTR\n'

How I found it: XOR C & TGT to get (P ^ goal), then XOR result with P to get the target plaintext.

# Part 5

# Part 6

You can get most of the information back, based on the different encryption schemes. CTR will have minimal change, as by setting 1 byte to 0, the most that will be affected is the R, G, or B value of one pixel. ECB and CBC will be about equal in distortion, which will happen to 16 bytes, so approximately 6 pixels will be affected. CTR minimizes damage by having the XOR happen after the counter has been encrypted, but since the plaintext or the result of (M[i] ^ C[i-1]) go through the encryption function, the entire block is affected.

# Part 7