Homework Number: 2

Name: Steven Rutledge

ECN Login: rutleds

Due Date: 1/27/22

\*Note: My code runs with “python3” followed by the CL script.

Problem 1:

I have a main function that reads the input and calls either a encrypt or decryption function. These functions use the get\_key, get\_round\_key, and substitution helper functions from the lecture and sample, however they required some changing to work for this program. The encypt pulls out each block, then cycles them through a loop of round keys then prints the final string in hex to the output file. Next the decryption was different because I had to read in hex then index 64 bits out of that in order to decrypt. Also the round keys are revered as the model in the notes says. Lastly I had to strip the padded ‘0’s that we added in the encrypt before writing to file.

Encrypted:



Decrypted:

Smartphone devices from the likes of Google, LG, OnePlus, Samsung and Xiaomi are in danger of compromise by cyber criminals after 400 vulnerable code sections were uncovered on Qualcomm's Snapdragon digital signal processor (DSP) chip, which runs on over 40% of the global Android estate. The vulnerabilities were uncovered by Check Point, which said that to exploit the vulnerabilities, a malicious actor would merely need to convince their target to install a simple, benign application with no permissions at all.The vulnerabilities leave affected smartphones at risk of being taken over and used to spy on and track their users, having malware and other malicious code installed and hidden, and even being bricked outright, said Yaniv Balmas, Check Point's head of cyber research. Although they have been responsibly disclosed to Qualcomm, which has acknowledged them, informed the relevant suppliers and issued a number of alerts - CVE-2020-11201, CVE-2020-11202, CVE-2020-11206, CVE-2020-11207, CVE-2020-11208 and CVE-2020-11209 - Balmas warned that the sheer scale of the problem could take months or even years to fix.

Problem 2:

My code uses the same DES algorithm as problem 1 however it is set to read and write the file differently because it is an image. I set up a Boolean called check to just identify if the header has been pulled yet. I found the header to be 128 bits so I read those and stored and wrote them separately. Then in a bv if more to read look I used the DES algorithm described above on the image bits. I then converted it to hex then bits to write to file.

