Jason Park

Park1036

2/4/2021

**<u>Problem 1</u>**

Encrypt and decrypt are essentially the same, and I'll be explaining just one. The main differences are selecting the amount of bits to read at a time in decrypt and then saving the output as ascii or hex. The most important part was reversing the roundkeys in decryption. First of all, I would read the input into a bitvector of 64 bits, padding 0's as necessary. The bitvector is divided into two halves, and then the main part of the feistel function occurs. (In order: expansion permutation, xoring with a roundkey, substitution, sbox permutation, and finally xoring with the original left. The original right becomes the new left, and the modified half becomes the new right. This continues for 15 more rounds, and then the left and right are flipped one more time. Finally, the 64 bits are written to the encrypted/decrypted.txt file.

<u>Encrypted message:</u>

```
D04a94419ec6556c20029c83a277790c5c6380595291ecc23a40b90d60ae5b114dcefad2a3765
2e80dbed6bea5ab59d92b8f043c65e1ced023bfe2aa6c4e162de19db8a75ff0f779baa3629395da
7d740e784fd3150db010f0054cd9f70a13ad6a553f954a79ca990dadc1a697ed8821548099cade
db2d6572810b06df68150cd16af08948628fab087c8577826ee1e0ca728ea3def08044613e608e
9ee27cf91a7052f2d11e6a42b371c5216e296a5486887d331794502300e42cfe9b228da863420c
7a9d2eb3797bf08185451fc5948a61890e2fec008abe98af6a313ba886300a3041f4ca3f273f177f
dd95fb97cfd7724c196421848826c105892bfbbe47e64551e146fc6d7130d00a2dd01fa6b14a6fb
6fb054f843710ddd9a311d54882db94802ceac4fd454332747d76b4e6be9e614545db3e6a8517c
413628268c07aa64f7175ce8cd40a00a86fb279fed136146b2f863a0f54bb9407a21418641ba55b
6641e1acb69bdf816a2cf41479f80ddde5a4a43da9f53758f152f58bb5919b65d4a80250c259b38f
```

498f354223cbbcbe14e5408aadc581eb0d5b19ef8219fbe42edc4e9f0467826a5c1a8141af67f0a8

97b4f212e5ab49417b576aba488381be68fc72080ee3ed00b56152e2d7da477b92c98379b694d

4f466eb0d93d083fd62d36ef1ca7f3b4399af80559ffbe0bd48b6eb441a569d479f94a54cb9ca816

990971e229831db528e70972cae2f82df38026db9db5b118ff17df3a7621911b51626ab948dd95a

777b4219b0ad0ab6180def71f24b42b23444d03b974681d583e07040d443d9365241e1fa77e1b4

684da92913a6ea9a2af407d586ddf8b242706e8775ced9fa520291bbafe441dfab3c4b5d93cfe16

54202d0b7ff5c381a6a2c489e2c756eb40b6b98482a49878d04f4422fcf43605826dd6dc32cd867

9e51bc800e3ae48673c19c5890c7eec8fc58775299ea756be20afff89395ac6b021f5bb37c36e30f

5948979c96b76537d8785721f1b9789e325d2e779c4e0859c093ba756c8998219cfc497f0b7f66e

259eebea3fba7a9ceed545ed833506d558c2dd9a8812ed9bdc69e9b0bdfbd514399d2a43be6bf5

0a2ddab68b3c3b449a430efdd46755871a8697737a7fd251de37390186a0c701ef7839a2b2ee99

a8d6aaf540aefd111c8507120fbc296ade7e0a30846b4ad461f2af4db654e8e0008fa5a2fa42381d

8350bd431d714c42f478ca43e3f31d4e2b77c4fea7b5fc92b55c18fd29ac06b78797333758a54e7

aab3439dc079d168b7c416e23cd49084a57ff1c0974dd36102983521b30ecd7fd1931201daab50

59b8139f5a3017cd7fd1931201daab744fae27721dad95cd7fd1931201daababcce6cc5c4ddacec

d7fd1931201daab462cde434ec9b646cd7fd1931201daab0013da3321192b13b1bcd34158bc587

8e3dbd126d0b7edf4cb345a27fa36e8df45ed30ec1b4cf954fd1fb2eb165e3fde33aab34a81ef30b

95855c1917ea1826f0093dfae7a9e6124ac8036677dc75ddb8cc79548b5f6b673e2aaa2e74de55

9d17d4c3597eb793828ce2eba373130b87f5201f6e90c06758f

Decrypted message:

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**

For this, I reused my des encryption function with a few modifications. I opened the file in binary first. Next, I read the next three lines to save the header for later. After that, I read the file in hex, and then used the hex to create bitvectors (padded as necessary), and finally the bitvector is passed into the des encryption.