Project 2
Computer Security
Subhash Rajapaksha
Zhou Shen

1. Initial attempt

We modified the original text in main.py and xorcode.py to see the difference; in details, we changed 4900 to 5900 in the text field. Then we encrypt the text with those two different methods. Next, we used WinHex to see the difference. Based on the difference we saw in WinHex, we can then modify the .hex file directly. It is a speculative way but not good in practice use.

2. Wise solution

For stream cipher encryption,
We have 88 characters including punctuation mark in the plain text. Because 4 of 4900 in the text is the last 5th character, we decided to check the last 5th element in datax.hex files with WinHex.
we found 9D is the last 5th character in datax.hex file, we xored 9D and 4, the result we get is the key which is 99 in hex value. Then we xored 99 and 5 to get the cipher text for 5 and the answer is 9C. Therefore, once we replace 9D with 9C and use xorcode.py to decrypt the message, we can get 5900 instead of 4900. After decoding, we got below plain text.

“Thomas Schwarz, Jesuit Community, Marquette University, Milwaukee, Wisconsin, USA, 5900.”

For 8B block-code DES with CBC encryption,
We have 88 characters including punctuation mark in the plain text. So there is no need to add paddings to ciphertext. Because 4 of 4900 in the text is the last 5th character in the last block, we decided to change the 4th byte in the last second block in data.hex files with WinHex.

To change 4 into 5 in “4900”, only the 8th bit is needed to flip (0000 1000 -> 0000 1001). Therefore corresponding bit of the C_{n-1} is flipped, so the byte in the data.hex changed from BC to BD. After decoding, we got below plain text. The 8 bytes block previous to the targeted block in the plain text is giving unreadable set of characters since we changed it in the ciphertext.

“Thomas Schwarz, Jesuit Community, Marquette University, Milwaukee, Wisco n÷ kMº³A, 5900.”