جامعة قطر
QATAR UNIVERSITY

College of Engineering-Dept. of Computer Science and Engineering

Masters of computing program

Network security (CMPT-612)

Assignment 2

Spring, 2023

Final mark: **20**

Submission due: March 18th, 2023 midnight

- **Provide your report as a pdf file (any other format will be rejected), use the name "Assignment2-{student name}-{student no}.pdf" with a cover page containing your name, student number, and e-mail.**
- **Include modified code, descriptions, and all results with clear comments on each result into the pdf report.**
- **Attach all other relevant files, including all python files, figures, and all codes used.**
- **In addition to the pdf report, submit all assignment attachments in one archive file "Assignment2-{student name}-{student no}.zip" to blackboard by the due date shown above.**

Introduction

The objectives of this assignment are as follows:

1- To get acquainted with using developed simulator for network security modeling.

2- To study and perform comparison between the most common block and stream ciphers over an error channel.

Refer to the model shown in figure 1.

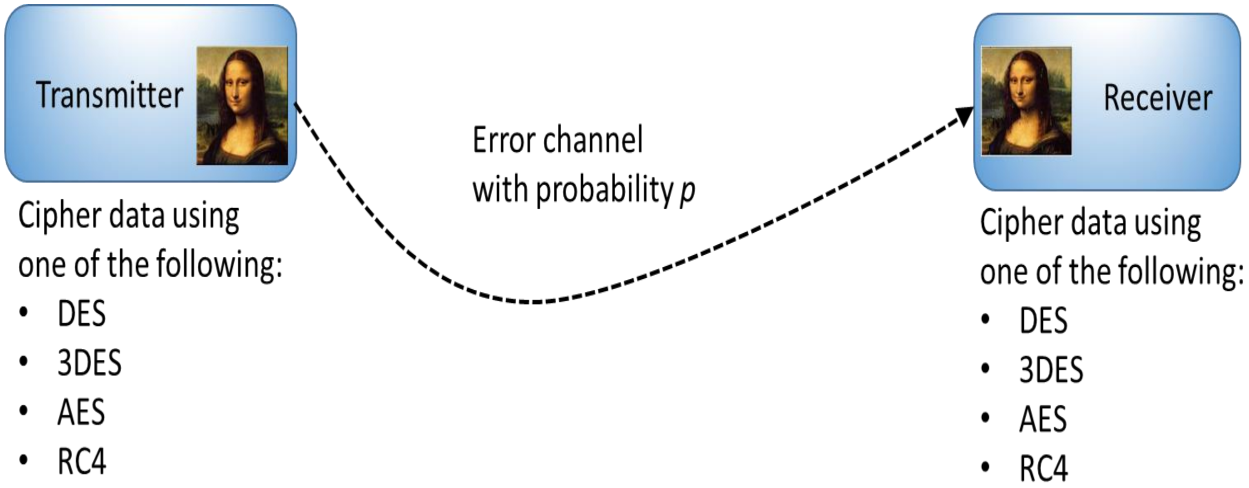

Figure 1 – Crypto image transfer over an error channel

Your task is to write/modify the python code provided to build this model, according to the instructions below and perform a series of measurements as described.

Consider the python code provided for simulating the crypto image transfer over an error channel. Please, familiarize yourself with the files, documentation inside each python file, and refer to the stub at the beginning of each script for a high level description of each file:

Tasks:

1. Modify the files DES_Demo.py and AES_Demo.py to change the text input, and run both files to test both ciphers with different input. Report the output of the text/images you test with.
2. Using the DES,py and DES_Demo.py files, create two files DES3.py and DES3_Demo.py to create the triple DES cipher/decipher, according to figure 2.3 in the textbook. Test the implemented algorithm with arbitrary text similar to what you did for DES and AES. Report your results.
3. Modify the RC4.py file to fill the encrypt function as explained in the function. Create a file RC4_Demo.py file and test the cipher with different input. Describe how the decrypt function can be done? Report your output.
4. Modify the utils.py file to fill the function channelError, to generate a random noise with the same length as the input, then add the noise to the input to simulate an error according to the probability of error prb parameter. Test your code for an arbitrary input text. Report your results.
5. Modify the main.py file by filling the compareCHERR_DES_AESx() function, to simulate the transfer of the monalisa200.jpg image over the error channel using the DES, AES-128, AES-192, and AES-256 cipher. Calculate the pixel error rate (see the PixelErrorRate function in the utils file) of the recovered image for channel error rates of 10e-4, 10e-5, and 10e-6. Also, calculate the time complexity for the encryption in each case. Report the recovered images for each case, a figure showing the pixel error rate for all four ciphers vs the channel error rates, and a figure showing the time complexity vs the channel error rates.

6. Modify the main.py file by filling the compareIMGQLTY_DES_AES_RC4() function, to simulate the transfer of all four monalisa images with different resolutions (i.e. 100 x 100, 200 x 200, 300 x 300, and 400 x 400) images using the DES, DES3, AES-128, and RC4 cipher. Calculate the time complexity for each case. Report the recovered images for each case, and a figure showing the time complexity vs the channel error rates.
7. (Only for the AES cipher) Modify the AES.py file, to implement the encrypt and decrypt functions code for the two extra modes of operations CFB, and CTR modes of operation. Test the code using the AES_Demo.py to trigger the CFB and CTR modes of operation. Report the results.
8. Repeat 5, fill the compareCHERR_AESmodes_RC4() function using the ciphers AES, AES with CBC, AES with OFB, AES with CFB, AES with CTR modes, and RC4.
9. Repeat 6, fill the compareIMGQLTY_AESmodes_RC4() using the ciphers AES, AES with CBC, AES with OFB, AES with CFB, AES with CTR modes, and RC4.
10. Comment on all the results achieved, including the recovered images, the error rates, and time durations incurred by each cipher.