Cybersecurity
Homework Assignment 2
COSC 3371, Spring 2024

Version 1, February 17, 2024
Version 2, February 20, 2024

One of the features needed for this assignment was deprecated in Python. The description assumes that you are using Java.

In this homework assignment, you will use the `javax.crypto` package. To get familiar with the most important classes and interfaces, read the "*Java security: Java security, Part 1: Crypto basics*" article at http://www.ibm.com/developerworks/java/tutorials/j-sec1/j-sec1.html, focusing on sections "*Keeping a message confidential*" and "*Ensuring the integrity of a message*."

Please solve the following problems by completing the attached Java source file. For each problem, replace the code between `// BEGIN SOLUTION` and `// END SOLUTION` with your solution (you can also import any standard Java library). The submission uploaded to Blackboard should include the completed Java source file. Please ensure that the uploaded source file can be compiled and executed without unhandled exceptions and that you have not used non-standard libraries.

In each problem, you aim to recover a plaintext (or at least some information about its contents). You will need a working Internet connection to solve this assignment. Each problem builds on the preceding one, so you must solve them in order.

About the file types. All text files can be viewed with a simple text editor. Files with the name msgX.txt (X = 1, ... 5) are message files with instructions on what to do and how to do it in Step X. There are some plainX.txt containing paragraphs in plain text. Files with .bmp are image files that can be viewed if not messed up. You will have to modify the ones that cannot be viewed. The images may provide more information for you to use. All ciphertexts are stored in binary files with an extension of .bin. They are not ASCII files. It is better not to open these files.

What to submit:

- Java code in one file.
- The files your program outputted (plain1.bmp, plain2.txt, Cipher3_modified.bmp, plain4B.txt, and plain5.bmp).

Problem 1 (15 points): The Game is Afoot

You are at 221B Baker Street in the company of Dr. Watson when the following e-mail arrives:

> "*Dear Mr. Sherlock Holmes,*
>
> *I must once again ask you to help us as a consulting detective. Three days ago, the invaluable Koh-i-Noor diamond was stolen from the Tower of London. We fear that the thieves are planning to sell the diamond on the black market, where it may be lost forever. Fortunately, the thieves acted hastily and they accidentally left a disk drive at the scene of the crime. We recovered two files from this drive (please find them attached), but our detectives at Scotland Yard were not able to make sense of them. We believe that the infamous Professor Moriarty is behind this spiteful act, but our detectives have no leads to follow. Sherlock, you are our only hope!*
>
> *Sincerely,*
>
> *Inspector Lestrade*"

The two files (`cipher1.bmp` and `msg1.txt`) are attached to the homework assignment. See the solution template for help.

Problem 2 (20 points): Out of Order

You look at Dr. Watson… he has fallen asleep while you were busy decrypting the message. You suspect that he would not be much help anyway, so you decide not to wake him up. Instead, you look at the ciphertext and see that it is 48 bytes (384 bits) long, meaning it consists of only three AES blocks, each 16 bytes (128 bits) long. You can try rearranging the three blocks in different ways (there are only five possibilities) to restore the ciphertext.

Out of the five possibilities, one will be the correct answer. You can visually check the solutions and decide which one is right. You have two choices. The simple (recommended) way is to code the right one only. The more challenging one is to write the code to choose the right one out of the five.

Problem 3 (20 points): Phantom Clue

Dr. Watson wakes up, looks at the ciphertext, and scratches his head. Not a good sign, obviously. It appears that you are again on your own. You look at the ciphertext: it is a bitmap image (BMP file) that has been encrypted using ECB block-cipher mode, so you should be able to see the patterns of the plaintext. However, you cannot open the image since the file header is encrypted, so no image-viewer program can figure out how to display it (e.g., without the header, a program will not know the image width and height). Suddenly, you get an idea: what if this image has the same format as the first one? You could restore the header by copying the first few thousand bytes of the first plaintext (`plain1.bmp`) to overwrite the first few thousand bytes of the ciphertext (`cipher3.bmp`) and then open the modified ciphertext in an image viewer!

Problem 4 (20 points): Two Plaintexts, Two Ciphertexts, and One Mistake

It seems your luck is running out: the cunning Professor Moriarty used a secure cipher, a secure mode of operation, and a secure key. Dr. Watson is about to call Inspector Lestrade to tell him that you could not discover the location of the meeting when you suddenly realize that Moriarty made a mistake. He used the same key twice with CTR block-cipher mode, essentially stream cipher. Since you have one of the plaintexts (`plain4A.txt`) and both ciphertexts (`cipher4A.txt` and `cipher4B.txt`) were XORed to the same pseudorandom sequence, you should be able to recover the other plaintext easily!

Problem 5 (15 points): End of the Line

Dr. Watson looks puzzled. How could you decrypt the ciphertext without knowing the mysterious Professor Moriarty's birthdate? Honestly, you do not know about those three numbers, either. However, there are not many combinations, so you could brute-force the key. But how will you know which key is the correct one? Well, the plaintext is a BMP file, which means that the value of the first byte is 66 (character 'B' in ASCII), and the value of the second byte is 77 (character 'M'). Further, you suspect that this bitmap file has the exact dimensions as the others, meaning the third, fourth, fifth, and sixth bytes should be the same as in the other bitmap files.