Cybersecurity
Homework Assignment 2

COSC 3371
2019 Fall

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 section "*Keeping a message confidential.*"

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 make sure that the uploaded source file can be compiled and executed without any unhandled exceptions and that you have not used any non-standard libraries.

In each problem, your goal is to recover a plaintext (or at least some information about its contents). Please note that you will need a working Internet connection to solve this assignment. Each problem builds on the preceding one, so you have to solve them in order.

Problem 1 (2 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 (2 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, which means that it consists of only three AES blocks, each being 16 bytes (128 bits) long. You can just try to rearrange the three blocks in different ways (there are only 5 possibilities) to restore the ciphertext.

Problem 3 (1.5 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 header of the file is encrypted, so no image-viewer program will be able to figure out how to display it (e.g., without the header, a program will not know what the image width and height are). 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 (1.5 points): Two Plaintexts, Two Ciphertexts, and One Mistake

It seems that 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 cannot 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, which is essentially a stream cipher. Since you have one of the plaintexts (`plain4A.txt`) and both ciphertexts (`cipher4A.bin` and `cipher4B.bin`) were XORed to the same pseudorandom sequence, you should be able to easily recover the other plaintext!

Problem 5 (2 points): End of the Line

Dr. Watson looks puzzled. How could you decrypt the ciphertext without knowing the mysterious Professor Moriarty's birthdate? To be honest, you do not have a clue about those three numbers either. However, there are not that 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 same dimensions as the other ones, which means that the third, fourth, fifth, and sixth bytes should be the same as in the other bitmap files.[1]

[1] Note that when you try to decrypt the ciphertext with an incorrect key, the decryption function might throw a `BadPaddingException` exception (due to the malformed plaintext). You can simply ignore these by catching them and then continuing with the next key candidate (i.e., surround the decryption function call with `try-catch`).