Bonus Homework A: Padding Oracle Attack on CBC

Submission policy. Submit your answers on Blackboard by 11:59pm Friday, **Dec. 3**, 2021. Your submission should include three files: a README with instructions to execute your program (including type of encoding used, i.e. Base64 or Hex) and the decryption of the challenge ciphertext, your code (preferable single file) and at least one example file .txt with ciphertext. Your code **has to be** very well documented to explain each step of the attack. No late submissions will be accepted.

Absolutely no online search for relevant code or libraries that might directly implement the attack.

The homework will be graded by the class TA Anuj Pokhrel (apokhre@gmu.edu).

Exercise 1. Padding Oracle Attack on AES-CBS mode.
In this assignment you will have to implement the Padding Oracle Attack for the CBC mode of operation when using PKCS#7 padding. (You can find the description of the attack in [KL] 3.7.2 and in the lecture slides).

You can write your code in any programming language you like (but we will only be able to help with questions on C, C++, Python, Ruby and Java).

The goal is to decrypt messages that were encrypted using AES-CBC (128 bits) using the key: `Crypto is cool!!`
You have to use this specific encryption key and you will have to hardcode it in your code – but of course not use it to decrypt! – should only be used as part of your CheckPadding oracle.

Your code should include 2 functions: *CheckPadding()* and *Padding Oracle()* and should work as follows:

1. Your code will take as input a file that contains an encrypted message using AES-CBS mode (for AES 128) under the key `Crypto is cool!!` and using PKCS#7 padding.

2. The function *CheckPadding()* will take as input a ciphertext, decrypt it with the hardcoded key and return true or false indicating whether padding is correct or not (this is your padding oracle).

3. The function *Padding Oracle()* will take as input a file that includes a cipherext (encrypted under AES-CBC using the hardcoded key). The size ciphertext will *have to be* a multiple of 16 bytes, else you can terminate.

   (a) Your function should now perform the padding oracle attack using the *CheckPadding()* method described above in order to decrypt the given ciphertext.

   (b) You will have to keep counter on how many times you had to call the the *CheckPadding()* function.

4. Your code should return the decryption (i.e. plaintext) and the counter (i.e. how many times you called the *CheckPadding()* function).

Here is a sample ciphertext for your to crack:

```
encryptionIntVecnZ7iCcAs0szUmzJSp4730y5Kkqorz2/3AnQ6uUatqDrDkh66y1ZVY+woF42MYWd3
```

Recall that the first 16 bytes is the IV.