COSC 4343: Computer Security and Privacy

Homework # 2: due on Thurs. 9/24/2020 at 11:59 pm

Remaining late days may be used in this assignment if needed.
Check "Late Days" policy in syllabus for more details.

Write a Python program that implements the Rivest Cipher (RC4), an example of stream cipher; a generalization of the One-Time Pad cipher. Your program should contain the following two main functions:

1- **oneTimePad:**
This function should implement the basic One-Time Pad cipher. Your function should take a plaintext string and a key as input, and should return the ciphertext string as output. It is intuitive that in the One-Time Pad cipher, the length of the key is equal to the length of the plaintext string. Make sure this pre-condition is satisfied before calling the function.

2- RC4:
This function implements the RC4 cipher as explained in class. Your function should take a plaintext string and a short key as input, and should return the ciphertext string as output. Remember that in RC4, the input key is short and therefore it needs to be stretched according to the RC4 cipher in order to generate a keystream equals to the size of the message. You can then send the generated keystream along with the message to the oneTimePad function that you previously implemented to encrypt your message.

After you create the above two functions, your program should display the following menu:

1- One-Time Pad
2- RC4
3- Exit

In Option 1, you are to test the encryption and decryption of a message using your oneTimePad function. (1) Ask the user for a message and key, (2) Check that the size of the message matches the size of the key, (3) Encrypt the message, (4) Decrypt the message using the same key and make sure the decrypted message matches the original message.

In Option 2, you are to test the encryption and decryption of a message using your RC4 function. (1) Ask the user for a message and a short key, (2) Encrypt the message, (3) Decrypt the message using the same key and make sure the decrypted message matches the original message.

Option 3 should exit the program. Note that this is the only way you can exit the program, and the menu should be inserted in a loop that will always be displayed to the user until option 3 is chosen.

Grading: (Total points: 10 pts.)

- (6 pts.): Program satisfies requirements and program components are functioning as expected.
- (2 pts.): Program is error free (no crashing), runs without any errors or warnings, and handles incorrect input appropriately without crashing the program.
- (1 pt.): Program is user friendly and easy to exchange input and output with the user.
- (1 pt.): Program contains comments explaining various sections of the source code.
- (-5 pts.): Program does not compile or run successfully.
- (-2 pts.): Program is not bulletproofed, i.e. does not handle input errors properly.
- (-1 pt.): Submitted file is not named according to instructions.
- (-1 pt.): Missing comments in program source code.
- (-1 pt.): Program menu does not loop or exits before instructed.

Submission instructions:

1- Make sure that your Python program is written in Python 3.X version and runs well on Mac OSX terminal (command line) using python3 command prompt.
2- Make sure that your Python program is well tested and runs without any errors or warnings. Include in your submission the Python source code and any other files you may have used.
3- Make sure your program is not using built-in functions that solve the major required components of the assignment, in which it may defeat the purpose of the assignment. When in doubt, ask me.
4- Clean up your code and remove any unnecessary code segments.
5- Add necessary comments to explain some of your code segments.
6- Your program should be submitted through Canvas.
7- Your submission filename should be named in the format of HW02-FirstLastname.py. For example, if your name is John Smith, then your python program file should be "HW02-JohnSmith.py".

Submission Checklist:

Use this checklist to make sure your submission contains the following:

- □ Source code with comments
- □ Any other files (like .txt or .dat files) that are necessary to run the program
- □ Program is bulletproof, that is, handles input errors without crashing
- □ Program produces no errors or warnings when compiled or interpreted
- □ Submission file follows the format described in the submission instructions
- □ Submission files downloaded and checked for missing files

Always let me know if you have any questions or need more clarification on the assignment or submission instructions.

Good Luck ☺