CSCE4853/5853 Homework 5 (Programming)

Due date: March 26, 2019

Full Grade: 80 pts

I. Task Description

In this assignment, you will implement a password cracking tool, and explore its performance. Your code should have two separate programs, a password generator and a password cracker.

The generator program should take a manually input username and password from the command line, generate a random 32-bit salt for it, and then store the salted, hashed password in a file pwd.txt in the following format: *[username, salt, H(password||salt)]*, where H() is the SHA-256 function, and "||" means concatenation. Suppose the password has N characters, where N is a parameter with value between 2 and 5 in our demo tests. (Note that too large values of N might take a very long time to crack on a common desktop/laptop computer.)

Then the cracker program reads the record *[username, salt, H(password||salt)]* from pwd.txt, and tries to crack the password. It will launch a brute-force attack, trying all possible passwords until finding one that matches the record.

Part 1: Implement a cracker program that considers passwords with lower-case letters only. Measure and output the number of trials and the time needed to find the password.

Part 2: Implement a cracker program that considers passwords with lower-case and upper-case letters. Measure and output the number of trials and the time needed to find the password.

Part 3: Implement a cracker program that considers passwords with lower-case letters, upper-case letters, and digits. Measure and output the number of trials and the time needed to find the password.

Part 4: Implement a cracker program that considers passwords with lower-case letters, upper-case letters, digits, and special symbols including $, #, %, &, *, (, and). Measure and output the number of trials and the time needed to find the password.

Hint: You don't need to implement four separate cracker programs for the four parts; actually one program could handle all the four cases. For example, the generator program could write into the file a #part number, and the cracker program can read that number to know which type of password is considered.

II. Tests

You need to demo your program to the Grader. A demo sign-up sheet will be distributed online. During the demo, your program will be tested in the following ways.

Test of Part 1: For a manually input password, output the number of trials and the time needed to crack the password.

Test of Part 2: Same as Part 1.

Test of Part 3: Same as Part 1.

Test of Part 4: Same as Part 1.

III. Other Instructions

Any programming language is fine.

Results must be printed to the command line (not to a file).

Submit your code to Blackboard as a .zip file named in this format:
HW5.YourLastName.YourFirstName.zip.