ECS 153 Homework 4

Overview

For this homework, you will be using OpenSSL and your programming language of choice. This is fairly open ended and you may choose what programming language to use. I suggest to use Python, but if you are more familiar with another programming language or have one you would prefer to use, feel free to use it instead. Regardless of language choice, make sure your code is concise, commented, and easy to read. You may be docked points if your scripts are needlessly complex or hard to read. You are to write your own scripts and report for this assignment but you may work with others to discuss the assignment or your code.

Cracking Hashed Passwords

There is a csv file in this zipped assignment called "passwords.csv". This csv has 20 passwords hashed using SHA-256 with no salt (one hash per line). Your assignment is to find the plain text of each of these twenty passwords. These passwords are some easy-to-guess passwords; they are English words with small permutations. These passwords plaintext are encoded using utf-8.

You are provided a file "dictionary.txt" which is a list of words that the passwords are based on. You are also provided with a list of permutations "permutations.txt" which contains the possible permutations applied to the English words; one example of a permutation is changing all "o" to a "0". All of the passwords in the passwords file are a single English word with at most one of the permutations in permutations.txt applied to it.

As a sanity check to make sure your hashing algorithm is correct, the first password in passwords.csv is "password".

1. Save your password cracking script to pw_cracker.py (use the file extension for the language you are writing in)
2. Write the twenty plain text passwords in order. If you were not able to obtain the plain text for a given password leave a blank line.

Writing Symmetric Key Encryption

For this section, you will be writing scripts to encrypt and decrypt a message using a symmetric key. You are to generate this symmetric key in OpenSSL, **not** in your script. Your scripts should be implemented as follows.

In your directory you will have a plain text message in plain.txt and your key file. Your encryption script will load the plain text message and the key file and use the key to encrypt the message and write the encrypted message to a file. Your decrypter script will load the encrypted message file and the key and use the key to decrypt the encrypted message; the script will then print the decrypted text to the console. This should match the plain text that was encrypted.

1. Save your encryption script to symmetric_encryption.py and the decryption script to symmetric_decryption.py (use the file extension for the language you are writing in)

Writing Public/Private Key Encryption

For this section, you will be modifying your previous scripts to use public/private key encryption. **Make sure to save the previous versions of your scripts before starting this part as you will need to turn them in as well.** You will need to generate a public key / private key pair; these should be put into separate files. This time, the encryption script will encrypt the plain.txt using the public key and save the encrypted version to a file. The decryption script will use the private key to decrypt the encrypted message and print the decrypted text to the console.

1. Save your encryption script to pubkey_encryption.py and the decryption script to pubkey_decryption.py (use the file extension for the language you are writing in)

Handin

Please submit your **answers to the questions** listed above in a single PDF document, your password cracking script, your symmetric key encryption and decryption scripts, and your public/private key encryption and decryption scripts.

Good luck!