Quiz. 2

(Deadline March 14, 2024)

Problem 1

You are expect to write a Python 3 program that breaks SHA1 hashes in a **brute force** manner. Please use the password list below, and copy them locally for ease of use.
https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt

For each hash value, your program should output the actual clear text **password**, count **the number of tries** before reaching a solution, and time **how long it takes** to break the hash, if found. For example:

```
$ python problem1.py
Hash: db3ae03df555104cd021c6308d5d11cfa40aac41
Password: hotmom
Took 30568 attempts to crack input hash. Time Taken: 0:00:00.073000
... and so on
```

Here are the provided SHA1 hashes you need to break:

a) Easy hash: ef0ebbb77298e1fbd81f756a4efc35b977c93dae

b) Medium hash: 0bc2f4f2e1f8944866c2e952a5b59acabd1cebf2

c) Leet hacker hash: 9d6b628c1f81b4795c0266c0f12123c1e09a7ad3
Hint: The salt term here is: dfc3e4f0b9b5fb047e9be9fb89016f290d2abb06
This is concatenated before hashing with another word to produce the salted hash.

d) Extra Credit: 44ac8049dd677cb5bc0ee2aac622a0f42838b34d
Hint: This hash constitutes two terms separated by one space

Problem 2

Checksums are crucial for ensuring data integrity in digital communications and storage. By generating a small, fixed-size data snippet or "hash" from a block of digital data using specific algorithms, checksums allow the verification of the integrity without requiring the original data.

You need to download this video file: https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/BigBuckBunny.mp4

Please calculate the checksums of the downloaded video file by using various hash functions, including MD5, SHA1, SHA-2(sha224, sha256 and sha512), and SHA-3(sha3-224, sha3-256 and sha3-512), and answer the following questions.

a) Write a Python 3 program to compare the speed of the hash algorithms.
Hint: You can use hashlib or time library

b) Which one is the fastest?

c) Rank the speed of each hash function.

Problem 3

Given the transposition cipher:

UONCS VAIHG EPAAH IGIRL BIECS

TECSW PNITE TIENO IEEFD OWECX

TRSRX STTAR TLODY FSOVN EOECO

HENIO DAARQ NAELA FSGNO PTE

Please decrypt this ciphertext.

Hint: How to determine the dimension of the rectangle?

1) Vowel Frequencies can help us to determine the dimensions of the rectangle. In English, approximately **40%** of plaintext consists of vowels. Therefore, for the correct dimension, each row of the rectangle should be approximately 40% vowels.

2) For example: "ASAIR ITFNM IMTKL SOIEE M". There are 21 letters.
Because we know that the message completely fills the rectangle, this suggests either a 3X7 or a 7X3 rectangle.
Consider our choice between 3X7 and 7X3 as an example. For a 3X7 rectangle, each row should contain approximately 2.8 vowels.
Let us note the difference between this estimate and the actual count.

For a 3X7 rectangle:

							Number of vowels	Difference
A	I	T	M	T	S	E	3	0.2
S	R	F	I	K	O	E	3	0.2
A	I	N	M	L	I	M	3	0.2

The average difference of each row is 0.2.

For a 7X3 rectangle:

			Number of vowels	Difference
A	F	L	1	0.2
S	N	S	0	1.2
A	M	O	2	0.8
I	I	I	3	1.8
R	M	E	1	0.2
I	T	E	2	0.8
T	K	M	0	1.2

The average difference of each row is 0.88.

So in this case, 3X7 rectangle is more likely.

What to turn in:

1) The file you need to upload is structured as follows:

- <student_id>.zip
  - problem1.py
  - problem2.py
  - problem3.py (If any)
  - <student_id>.pdf

2) The <student_id>.pdf file should contain instructions on how to run your code and the solution to the provided exercises.