

YSC2221 Assignment 2

Background

In computer security, passwords are encrypted using a one-way mathematical function called hash functions. Hash functions convert your password strings into human-unreadable form of data called hashes. The security comes from the fact that it is mathematically impossible to reverse the hashes back to their original password strings.

There is however a way to hack this. Rainbow tables have been constructed to precompute large numbers of password combinations into hashes in advance. These tables contain a mapping of hashes to their original passwords.

A security expert who intends to crack these passwords will compare these hashes against the hashes in the tables to determine what is the original password.

What you have to do?

In this assignment, your program will attempt to determine the original image whose every pixel has been hashed.

1. Read in the rainbow table for every possible pixel value
2. Read in every hashed data pixel of hashed file
3. Compare the hash of each pixel against your rainbow table to determine the original pixel value
4. Reconstruct the original pixel values of the image
5. Write the original image back to a file

Provided Files

These files are required for the assignment.

hashed.txt

This file contains the hashed values of all the pixel data of the original image. It is in this format

```
640 360
70234eba 3af5f102 28405eec
70234eba 3af5f102 28405eec
...
```

The first line contains 2 numbers, the first number 640 is the number of columns and the second number 360 is the number of rows of pixels in that image.

For the subsequent lines, each line represents the colour data for each pixel for red, green and blue components respectively. Each colour component value has been individually hashed.

The order of the pixels described in this file is from left-to-right then top-to-down of the image collapsed into linear form.

rainbow-table.txt

Pixel colour values range from 0 to 255.

```
2144df1c 0
99f8b879 1
8b4d1797 2
...
ff41d9ed 255
```

For each line in this file, the value on the left is the hash of the number on the right.

requirements.txt

A list of libraries and their versions that are required for the assignment.

ysc2221_assignment2_xxx.py

This is the provided skeleton code where you will primarily work on.

test_ysc2221_assignment2_xxx.py

Sample tests to get you started on what is expected of each function.

Note that passing all tests in this provided sample does not imply a complete grade. You may add additional test cases for your test purposes. You don't have to submit this file for grading.

Setup and Testing

```
# Install the required libraries
pip3 install -r requirements.txt

# Run the test
pytest test_ysc2221_assignment2_xxx.py

# Run the program to generate the image file for submission
python ysc2221_assignment2_xxx.py
```

Work submissions

You have to submit 2 files for grading.

Code

Submit `ysc2221_assignment2_xxx.py`. Replace `xxx` with your student ID (e...) before submission.

Percentage indicates the grade allocation for each function.

- `read_rainbow_table()` - 30%
 - `read_hashed_file()` - 30%
 - `unhash_file()` - 30%
 - `write_bytes_to_file()` - 10%
1. Fill in your details at the top if this file
 2. Ensure your code passes the style check by flake8
 3. Fill in all the functions according to the requirements specified
 4. Do not modify the function definitions
 5. Your code cannot import any libraries
 6. Feel free to add comments to your code to add clarity for the grader
 7. If you manage to determine what is the original image, write the answer in the designated area at the top of the file. If you are unclear what is the image, approach your peer tutors before submission. Do NOT discuss with your peers!
 8. You may add additional test cases in `test_ysc2221_assignment2_xxx.py` for your own use but you do not have to submit that file.
 9. Final submission must be fully interpretable by Python

Image

Your program will generate an image. Submit the image with the filename `xxx.jpg` where `xxx` is your student ID (e...).