UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF COMPUTER NETWORK AND COMMUNICATION

Logo, company name

Description automatically generated

**REPORT**

Subject: Cryptography

Semester II (2023 – 2024)

**Report Lab 4, 5**

Student: Võ Ngọc Bảo

Student ID Number: 22520127

Class: NT219.O22.ANTT

University of Information Technology

Lecturer: Nguyễn Ngọc Tự

**Hồ Chí Minh City, July 2024**

Report Lab 4, 5

**Student Information**

Full Name: Võ Ngọc Bảo

Student ID Number: 22520127

Class: ATTT2022.1

**Device Information**

CPU: AMD Ryzen 5 5600H 3.30 GHz

Ram: 8 GB DDR4 3200MHz

SSD PCle: 512GB

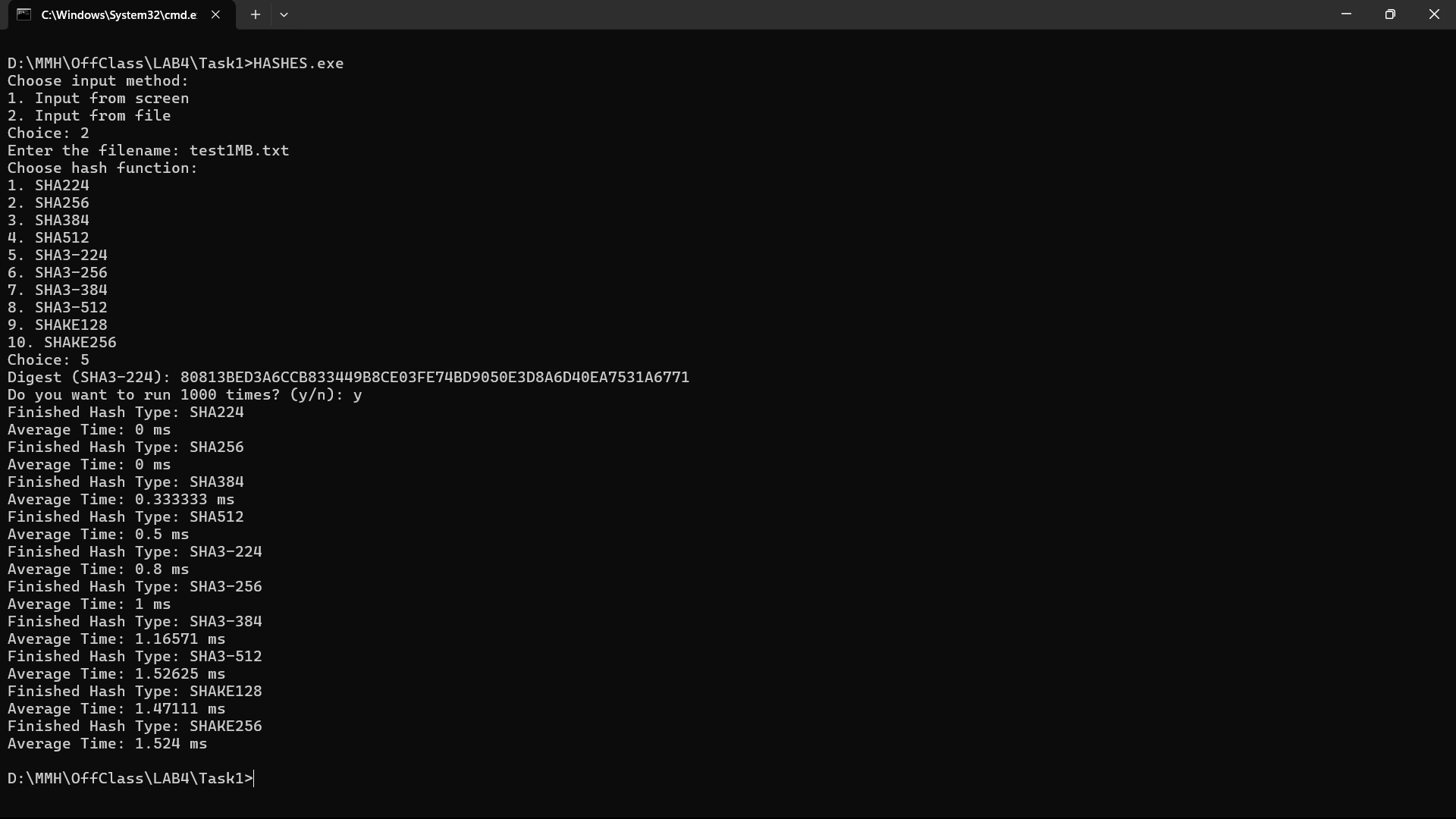
**LAB4:**

**Task 1:**

Time unit : miliseconds

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Window | SHA224 | SHA256 | SHA384 | SHA512 | SHA3-224 | SHA3-256 | SHA3-384 | SHA3-512 | SHAKE128 | SHAKE256 |
| Test1MB.txt | 0 | 0 | 0.3333 | 0.5 | 0.8 | 1 | 1.16571 | 1.52625 | 1.47111 | 1.524 |
| Test25MB.exe | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Test100MB.exe | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Test300MB.msi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Test1GB.exe | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Test1MB.txt



Test25MB.exe

A screenshot of a computer

Description automatically generated

Test100MB.exe

A screenshot of a computer

Description automatically generated

Test300MB.msi

A screenshot of a computer

Description automatically generated

Test1GB.exe

A screenshot of a computer

Description automatically generated

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ubuntu | SHA224 | SHA256 | SHA384 | SHA512 | SHA3-224 | SHA3-256 | SHA3-384 | SHA3-512 | SHAKE128 | SHAKE256 |
| Test1MB.txt | 1 | 1 | 1.66667 | 2 | 2.4 | 2.66667 | 3.00714 | 3.65625 | 3.62667 | 3.671 |
| Test25MB.exe | 24 | 24 | 38.1467 | 38.1575 | 40.586 | 42.5333 | 46.4629 | 53.475 | 52.6289 | 52.824 |
| Test100MB.exe | 97.02 | 74.405 | 101.207 | 114.582 | 130.588 | 144.31 | 162.996 | 192.004 | 190.301 | 192.33 |
| Test300MB.msi | 194.15 | 170.815 | 266.83 | 313.885 | 368.35 | 410.187 | 466.637 | 553.135 | 547.003 | 552.961 |
| Test1GB.exe | 471.29 | 468.45 | 813.78 | 986.343 | 1165.52 | 1299.6 | 1486.66 | 1769.78 | 1753.08 | 1776.93 |

Test1MB.txt

A computer screen shot of a person

Description automatically generated

Test25MB.exe

A computer screen shot of a person

Description automatically generated

Test100MB.exe

A computer screen shot of a person

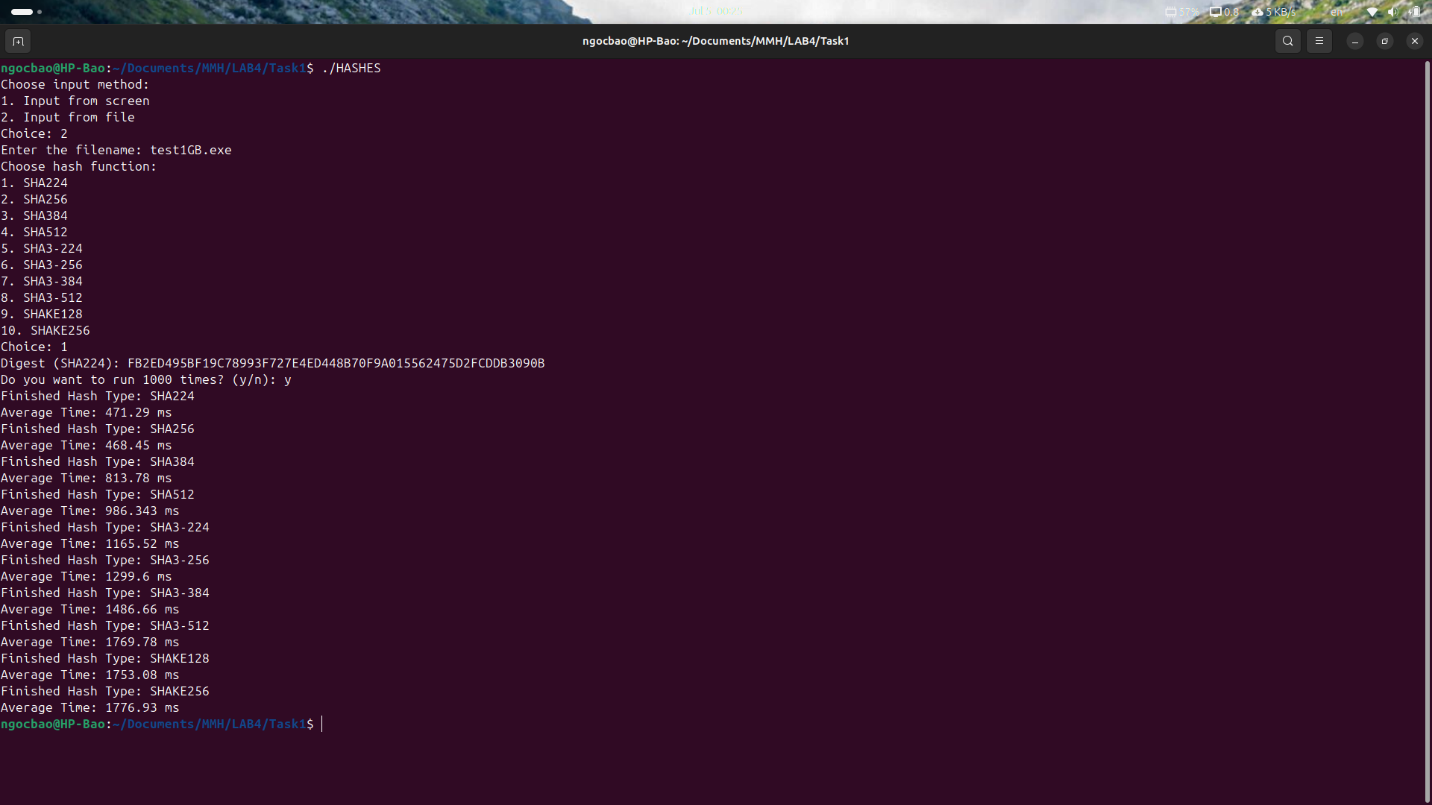
Description automatically generated

Test300MB.msi

A computer screen shot of a person

Description automatically generated

Test1GB.exe



**Task2 :**

A screenshot of a computer

Description automatically generated

**Task3 : Collision and length extension attacks on Hash functions**

Task3.1: Two collision messages have the same prefix string (Generate 2 different files with the same MD5 hash)

Prefix.txt:

A computer screen shot of a computer error

Description automatically generated

Solution: Use md5\_fastcoll

Step 1: Run md5\_fastcoll to create 2 different files with the same prefix and the same MD5 hash digest.

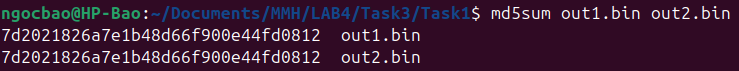
A computer screen shot of a program code

Description automatically generated

Step 2: Compare the two out files.



Step 3: Use md5sum and compare hash.



That is collision!

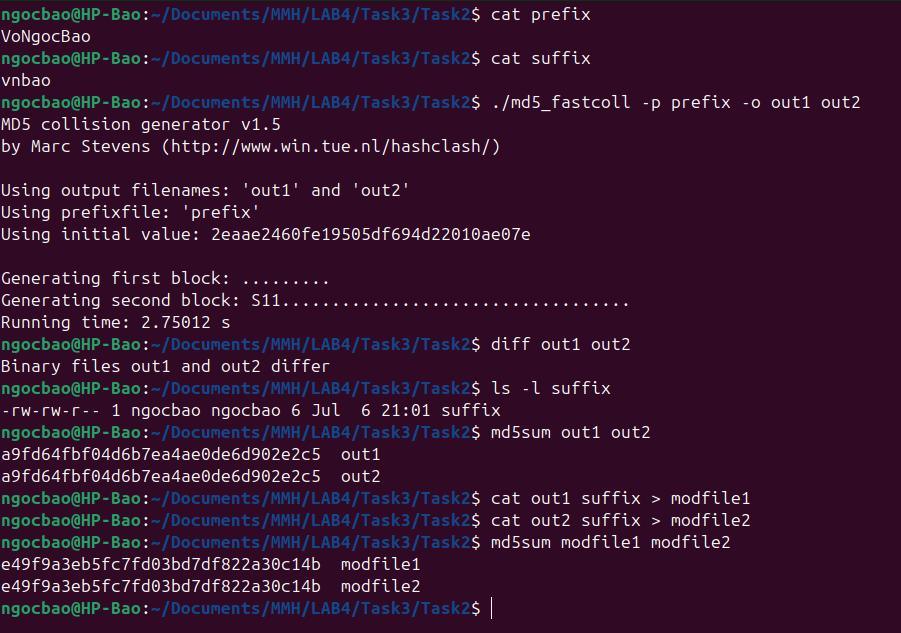
Full step:

A screenshot of a computer program

Description automatically generated

Task3.2: Understanding MD5's property

Solution: use hashclash (md5\_fastcoll)



Task3.3: Two different C++ programs but have the same MD5 (Generating two executable files with the same MD5 hash)

Solution: md5\_fastcoll

Step 1: implement the program

A screenshot of a computer

Description automatically generated

Step 2: Build the \*.o file with the gcc



Step 3: Check the \*.o file with bless and spot the location of a continuous block of A's, the byte offset is 1040 (4160). Slipt the \*.o into 3 parts: prefix, suffix, and the var. The var is required that MD5(prefix || variant1 || suffix) = MD5(prefix || variant2 || suffix).

The prefix is chosen to be a multiple of 64. The offset is 4224.



The suffix is kept with 10FF because of ending byte offset as 10FF in files



Step 4: Using the md5\_fastcoll to get the same hash transfer to file1 and file2

A computer screen shot of a computer code

Description automatically generated

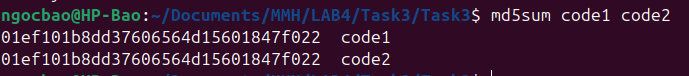
Step 5: Create 2 binaries transfer to code1 and code2



Step 6: Check the binary



Step 7: Use md5sum and compare hash



That is collision!

**Task4: Length extension attacks on MAC in form**

A computer screen with white text

Description automatically generated

A close up of a screen

Description automatically generated

**LAB5:**

ECDSA:

A screenshot of a computer

Description automatically generated