IX

Lab

**Practical Report IX**

**Hash Functions and Digital Certificates**

**Subject: Network Security**

**Class: CS4243.O11. CTTT**

|  |  |
| --- | --- |
| **Instructor** | Nguyễn Xuân Hà |
| **Student** | Đỗ Công Trình (20522058) |
| Link code: <https://github.com/hell-angle/NS_Lab4> |
| **Percentage done** | 80% done |
| **Thời gian thực hiện** | 15/11/2023 – 22/1/2023 |
| **Tự chấm điểm** | 9/10 |

1. Lab Task.
2. Generating message digests (hash values) and HMAC

Task 1.1 Your task is to write an application to calculate hash values (at least 3 different types: MD5, SHA-1, SHA-2) for an input, which could be:

* Text string
* Hex string
* File

You are able to use the hash library for your own programming language. Then, test your application with the following exercise:

* Generate the hash values of "UIT Cryptography" in Text string and Hex string format. Then, compare the results with other tools to verify.
* Create a text file and put your name and student’s ID inside. For example, "Nguyen Van An - 19521234". Generate hash values H1 of these files (using both MD5 and SHA-1). Subsequently, send this file to your friend via email or upload it to Google Drive and download it. Calculate the hash values of the downloaded file and compare them to those of the original file. Please observe whether these hash values are similar or not.

Solution:

* Application Programming:

A screen shot of a computer program

Description automatically generated

Image 1:

* Test Application:

1. Text and hex
2. text

A computer screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Hex format

A screenshot of a computer

Description automatically generated

We can use the encoded string to hash: 5549542043727970746f677261706879

MD5:

A screenshot of a computer

Description automatically generated

SHA-1:

A screenshot of a computer

Description automatically generated

SHA-256:

A screenshot of a computer

Description automatically generated

Conclusion: Application and website testing algorithms is similar.

1. Upload File

A computer screen shot of a program code

Description automatically generated

Image:

In terminal screen, you can see 2 times run file B1.py. I want to mention here is first time is running file not upload in google drive and second time run file after downloading file context. So, no thing is change before and after uploading file in google drive.

1. Hash properties: One-way vs Collision-free

Task 2.1

It is now well-known that the cryptographic hash function MD5 and SHA-1 has been clearly broken (in terms of collision-resistance property). We will find out about MD5 and SHA-1 collision in this task by doing the following exercises:

1. Consider two HEX messages as follows:

Message 1 d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f89 55ad340609f4b30283e488832571415a085125e8f7cdc99fd91dbdf280373c5bd8 823e3156348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0e99f 33420f577ee8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70 Message 2 d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f8 955ad340609f4b30283e4888325f1415a085125e8f7cdc99fd91dbd728037 3c5bd8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d 248cda0e99f33420f577ee8ce54b67080280d1ec69821bcb6a8839396f965a b6ff72a70

How many bytes are the difference between two messages?

Let’s generate MD5 hash values for each message. Please observe whether these MD5 are similar or not and describe your observations in the lab report.

1. Consider two executable programs named hello and erase:
   * If you are using Windows, you can download these .exe files here.
   * If you are using Linux, you can download the similar: hello and erase. Run these programs and observe what happens. Note these programs must be run from the console. Let’s generate MD5 hash values for these programs and report your observations
2. Download two PDF files: shattered-1.pdf and shattered-2.pdf. Open these files to check the difference. Then generate SHA-1 hash for them, and observe the result.

***Draw the conclusion based on your observations, explain the reasons for the existence of collision in MD5 and SHA-1.***

Solution:

1. Consider two HEX messages as follows:

Code python programming to calculate difference bytes of 2 hex messages:

A computer screen shot of a program

Description automatically generated

Image :

Result: after compiled this program and here is result:

A screenshot of a computer code

Description automatically generated

Image :

Describe observations:   
Observing different plaintexts resulting in the same MD5 hash value indicates a significant vulnerability in the MD5 algorithm known as a collision. This undermines the fundamental security properties of hash functions, particularly uniqueness. Applications relying on MD5, such as data integrity verification and password hashing, become susceptible to attacks. Security best practices advise transitioning to more secure hash functions like SHA-256 to mitigate these risks and ensure robust cryptographic practices.

1. Consider two executable programs named hello and erase:

• If you are using Windows, you can download these .exe files here.

• If you are using Linux, you can download the similar: hello and erase.

Solution:

A screenshot of a computer

Description automatically generated

Image :

A screenshot of a computer

Description automatically generated

Image:

1. Download two PDF files: shattered-1.pdf and shattered-2.pdf. Open these files to check the difference. Then generate SHA-1 hash for them, and observe the result

Solution:

A screenshot of a computer

Description automatically generated

Image :

A screenshot of a computer program

Description automatically generated

Image:

Draw the conclusion based on your observations, explain the reasons for the existence of collision in MD5 and SHA-1.