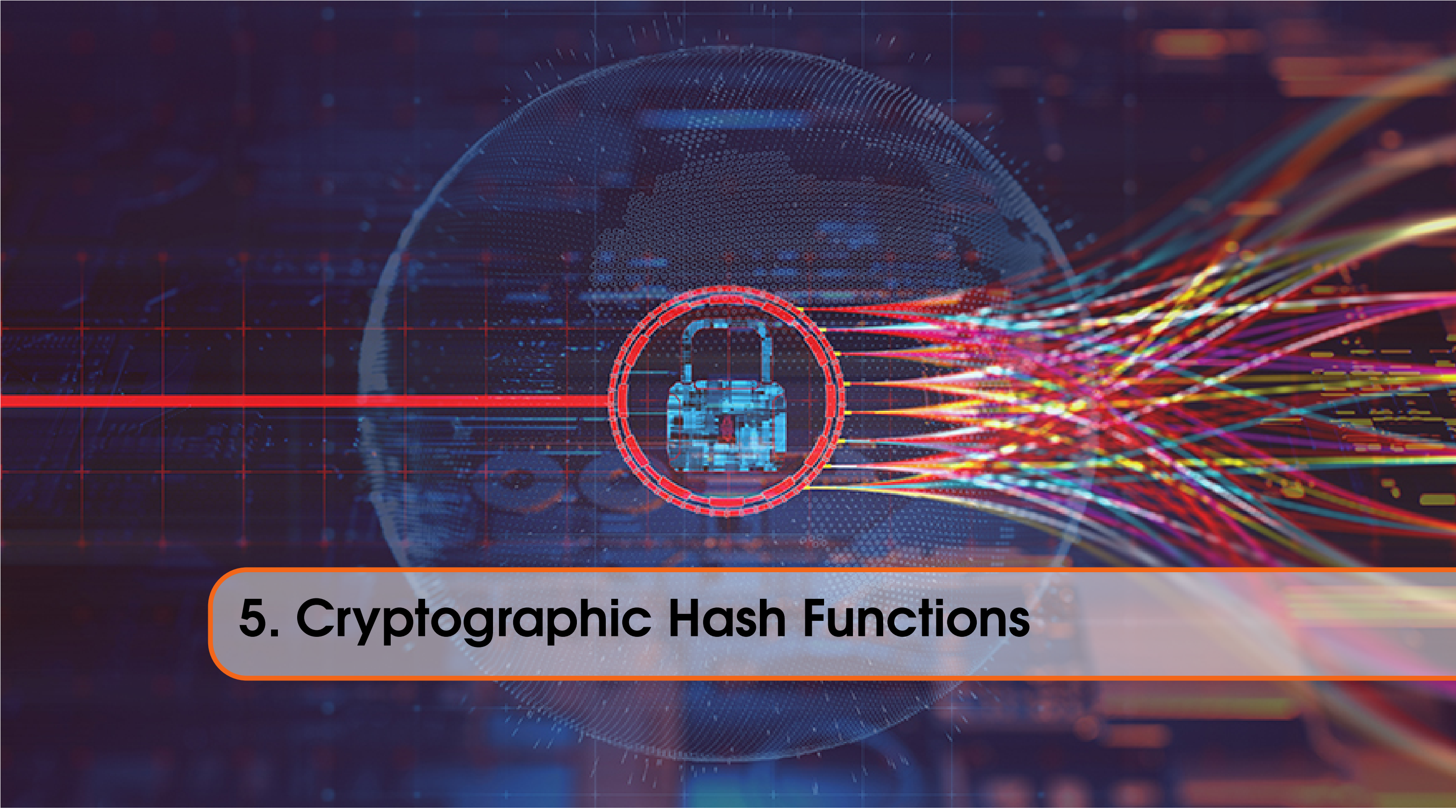
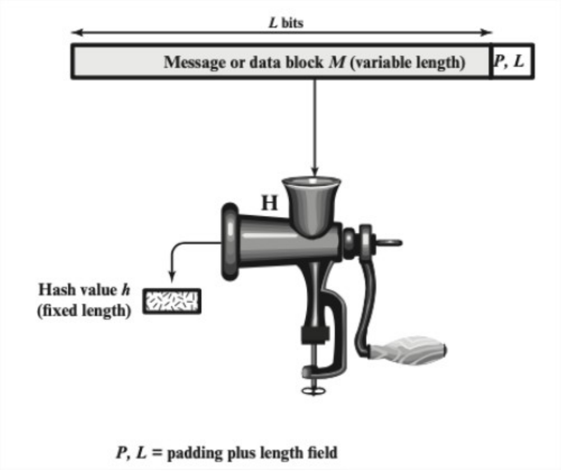


5. Cryptographic Hash Functions

5.1 OVERVIEW

5.1.1 Introduction and learning objective



L bits
Message or data block M (variable length)
P, L
H
Hash value h
(fixed length)
P, L = padding plus length field

Figure 5.1: Hash Function

For cryptographic applications, need one or more of these properties:

1. **The one-way property** : Given **h**, it's infeasible to find **x** such that **H(x)=h**. *(Also called the "Preimage resistance")*
2. **The collision-free property**:
   - *Weak collision resistance*: Given x, it's infeasible to find $y \neq x$ such that $H(x) = H(y)$. *(Also called "Second preimage resistance")*
   - *Strong collision resistance* : It's infeasible to find any two x and y such that $x \neq y$ and $H(x) = H(y)$. *(Also called "Collision resistance")*

R A secure one-way hash function needs to satisfy two properties: the one-way property and the collision-resistance property. The one-way property ensures that given a hash value h, it is computationally infeasible to find an input M, such that $hash(M) = h$. The collision-resistance property ensures that it is computationally infeasible to find two different inputs M1 and M2, such that $hash(M1) = hash(M2)$.

Several widely-used one-way hash functions have trouble maintaining the collision-resistance property. At the rump session of CRYPTO 2004, Xiaoyun Wang and co-authors demonstrated a collision attack against MD5 [3]. In February 2017, CWI Amsterdam and Google Research announced the SHAttered attack, which breaks the collision-resistance property of SHA-1 [4]. While many students do not have trouble understanding the importance of the one-way property, they cannot easily grasp why the collision-resistance property is necessary, and what impact these attacks can cause. *(SEED Labs - Wenliang Du, Syracuse University)*

The learning objective of this lab is for students to get familiar with one-way hash functions and Message Authentication Code (MAC). After finishing the lab, in addition to gaining a deeper understanding of the concepts, students should be able to use tools and write programs to generate hash value and MAC for a given message. Besides, other goal is for student to really understand the impact of collision attacks, and see in first hand what damages can be caused if a widely-used one-way hash function's collision-resistance property is broken (MD5 and SHA-1 collision).

5.1.2 Background

To complete this lab well, you are expected to know how cryptographic hash functions work. If you are not familiar with them, you should find out in more detail in:
Chapter 11: Cryptographic Hash Functions *W. Stallings, CS book - Cryptography and network security: Principles and practice, 7th ed. Boston, MA, United States: Prentice Hall, 2017*

5.1.3 Lab environment and Tools

1. **Operating system:** Windows, Linux (Ubuntu), MacOS
2. **Programing languages and IDE:** Flexible, you are free to choose any programming language you wish (Python, Golang are highly recommended)

5.2 LAB TASKS

5.2.1 Generating message digests (hash values) and HMAC

Task 5.1 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:

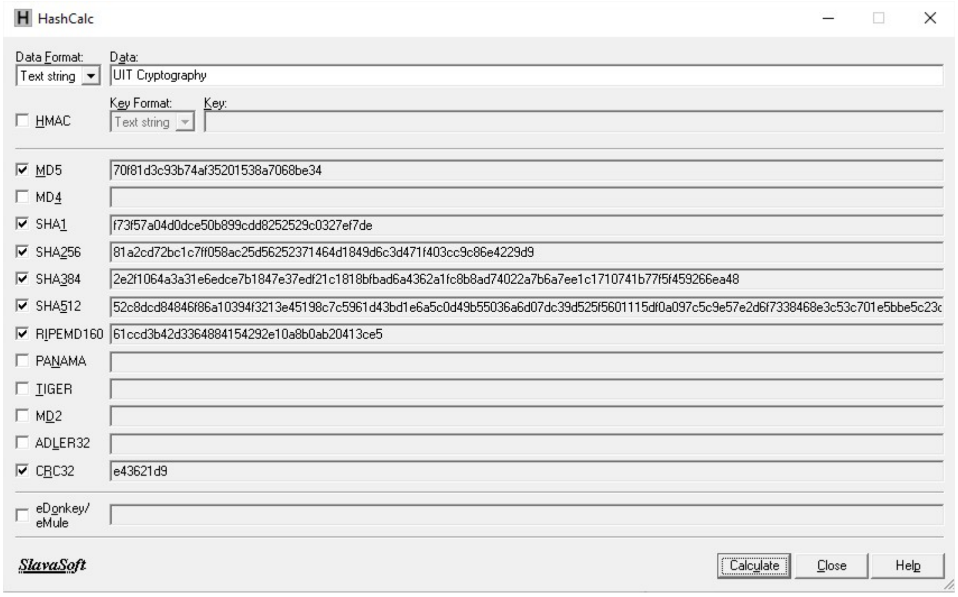
- Text string
- Hex string
- File

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

1. Generate the hash values of **"UIT Cryptography"** in Text string and Hex string format. Then, compare the results with other tools to verify.

2. Create a text file, put your name and student's ID inside. For example, "Nguyen Van An - 19521234". Generate hash values H1 of this files (using both MD5 and SHA-1). Subsequently, send this file to your friend via email or upload to Google Drive and download. Calculate hash values of the downloaded file and compare to those of the original file. Please observe whether these hash values are similar or not.

Tips 5.1. *You can refer to a similar application like HashCalc (`https://www.slavasoft.com/hashcalc/`)*



HashCalc

Data Format: Text string

Data: UIT Cryptography

□ HMAC Key Format: Text string Key:

☑ MD5 70f81d3c93b74af35201538a7068be34

□ MD4

☑ SHA1 f73f57a04d0dce50b899cdd8252529c0327ef7de

☑ SHA256 81a2cd72bc1c7ff058ac25d56252371464d1849d6c3d471f403cc9c86e4229d9

☑ SHA384 2e2f1064a3a31e6edce7b1847e37edf21c1818bfbad6a4362a1fc8b8ad74022a7b6a7ee1c1710741b77f5f459266ea48

☑ SHA512 52c8dcd84846f86a10394f3213e45198c7c5961d43bd1e6a5c0d49b55036a6d07dc39d525f5601115df0a097c5c9e57e2d6f7338468e3c53c701e5bbe5c23c

☑ RIPEMD160 61ccd3b42d3364884154292e10a8b0ab20413ce5

□ PANAMA

□ TIGER

□ MD2

□ ADLER32

☑ CRC32 e43621d9

□ eDonkey/ eMule

Figure 5.2: HashCalc application on Windows OS

5.2.2 Hash properties: One-way vs Collision-free

Task 5.2 It is now well-known that the crytographic 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 follow:
   Message 1
   d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f89
   55ad340609f4b30283e488832571415a085125e8f7cdc99fd91dbdf280373c5b
   d8823e3156348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0
   e99f33420f577ee8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70
   Message 2
   d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f89
   55ad340609f4b30283e4888325f1415a085125e8f7cdc99fd91dbd7280373c5b
   d8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0
   e99f33420f577ee8ce54b67080280d1ec69821bcb6a8839396f965ab6ff72a70
   How many bytes are the different 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.

2. Consider two executable programs named hello and erase.
   - If you are using Windows, you can download these .exe files here here.
   - If you are using Linux, you can download the similar pair: **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
3. Download two PDF files: shattered-1.pdf and shattered-2.pdf. Open these files to check the different. Then generate SHA-1 hash for them, observe the result.

Draw the conclusion base on your observations. Could you explain the reasons for the existence of collision in MD5 and SHA-1? ■

Advanced Task 5.1 Find out how Windows passwords are stored (which hash functions are used). Once you have that hash value, is there any way to find the original password? Suppose that you have 3 password hash values of 3 users **u1, u2, u3** as follow:
u1:1003:NO PASSWORD*********************:8846F7EAEE8FB117AD06BDD830B7586C:::
u2:1004:NO PASSWORD*********************:C705696627D5DE4C57E4E78E01A14EAA:::
u3:1005:NO PASSWORD*********************:03008F60BA0146D0B9E7B21221A722C3:::
Try to find the original passwords of these users. Note that these hash are generated by **pwdump7** [1] tool.

Advanced Task 5.2 Conduct **MD5 Collision Attack Lab** (4 sub-tasks) in SEED Labs. You can find out the full version of this SEED lab here:
https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_MD5_Collision/

5.3 REQUIREMENTS - EVALUATION

5.3.1 Requirements

You are expected to complete all tasks in section **Lab Tasks**, advanced tasks are optional and you could get bonus points for completing those tasks. You can either practice individually or work in team (2 members/team). If you prefer to work in team, please register in the first class with instructor and keep working in your team afterwards.

Your submission must meet the following requirements:

- You need to submit a **detailed lab report in .PDF** format, **using report template** that was provided on the courses website. You need to provide screenshots, to describe what you have done and what you have observed and explanation to the observations that are interesting or surprising.
- **Both of Vietnamese and English report are accepted**, that's up to you. Students in High Quality and Honor program are expected to write lab report in English.
- **Students in Honor program are expected to finish all advanced tasks.**
- When it comes to **programming tasks**, please attach all source-code and executable files (if any) in your submission. Please also list the important code snippets followed by explanation and screenshots when running your application. Simply attaching code without any explanation will not receive points.

[1] https://www.tarasco.org/security/pwdump_7/

- **Submit work you are proud of - don't be sloppy and lazy!**
  Your submissions must be your own work. You are free to discuss with other classmates to find the solution. However, report-copy is prohibited. Both of report owner and copier are received *a special gift - Zero point*. Please remember to clearly cite any source of material (website, book,...) that influences your solution.

> **Notice 5.3.1** Combine your lab report and all related files into a single ZIP file (.zip), name it as follow:
>
> **StudentID1_StudentID2_ReportLabX**
>
> *For example: 19520123_19520234_ReportLab4.zip*. Maximum file size: 10MB.
> If it is larger than 10 MB, then you should upload to Google Drive and submit the link to your report *(remember to share view permission with instructor)*

5.3.2 Evaluation

- Well complete all basic tasks: 70% *or 50% with students in Honor program (.ANTN)*
- Well complete the advanced tasks: 10-100% or bonus points to the next lab.
- In-class activities: + up to 10 points
- Work individually: **+ 2 point** *(updated)*
- Report written in English: + up to 2 points

> **Notice 5.3.2** Assignments are expected to be completed by due date. For every day the assignment is late after the deadline, 10% will be deducted from the lab score. No assignments will be accepted once they are 7 or more days late.
>
> Any part presented in your report may be randomly examine at the next class to verify your work. Absence without any rational reason could result in 30% deduction (or more) of your team's score.

5.4 REFERENCES

[1] William Stallings, *Cryptography and network security: Principles and practice, 7th ed*, Pearson Education, 2017. *Chapter 11: Cryptographic Hash Functions*
[2] Wenliang Du (Syracuse University), *SEED Cryptography Labs*
https://seedsecuritylabs.org/Labs_16.04/Crypto/.
[3] MD5 Collision, Available: http://www.mscs.dal.ca/~selinger/md5collision/.
[4] SHA-1 Collision, Available: https://shattered.io/.

Training platforms and related materials

- ASecuritySite - https://asecuritysite.com
- Cryptopals - https://cryptopals.com

Attention: *Don't share any materials (slides, readings, assignments, labs,...) out of our class without my permission!*