CSIT 495/595 - Introduction to Cryptography

Course Project

Total Points: 20

Basic Details

- Groups: You will do this project in groups. Each group should be of size 3. I recommend that you talk to your fellow classmates and form groups by yourself. Once you form a group, please email me your group details.
- Collaborative Work: Within each group, it is expected that all the team members work in a collaborative manner (it is up to the group members on how they distribute the work among themselves).

Project Topic

In this project you will implement a File Integrity Management System using HMAC. More specifically, given a set of files, you need to generate their hashes and use them to verify the integrity of this files at a later point. If a file has been modified in any manner, your program should be able to detect this. As part of this project, you can use any of the standard HMACs, such as SHA-2 or MD5 (we will also discuss about these schemes in the course). Note that MD5 is cryptographically broken, but many legacy systems are still using them; therefore, it is fine in using MD5 in this project. However, you also have the option of using SHA-2 or SHA-3. Please refer to the papers given in the reference section.

Project Deliverables

The first step for you is to understand the problem and how to use HMAC to solve this problem. You need to implement two programs as follows:

(a). The first program should take two inputs as arguments: *directory1* and *directory2*. Your program should compute a hash value (using HMAC) for each file in the folder *directory1* and store that hash value in a new file that will be saved under the folder *directory2*.

(b). The second program should perform the verification process. It also takes the same two input arguments as the first program. It should generate hashes again (you can reuse some code from the first program here) and check whether they are matching with the corresponding values stored in *directory2*. For each file, this program should output two strings: the filename and YES/NO (denoting whether the hashes matched or not).

Please note that you can use any programming language (C, C++, Java, etc.) to implement your project as per your convenience. However, it is your responsibility to check whether the programming language you select supports proper crypto libraries that you may want to use in your project. C, C++, JAVA provide support for various crypto libraries.
The evaluation plan is provided below.

- I will arrange a meeting (sometime in November) with each group to review the project progress and provide some suggestions.

- Each group need to give a project demo to me towards the end of the course (demo schedule will be released later).
- Points will be assigned to each group based on the correctness and robustness of their programs.

If you have any questions related to this project, please post them on the Canvas or contact me directly.

References

[1] http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

[2] http://dl.acm.org/citation.cfm?id=RFC2104

[3] http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf