Lab 5 - One-Way Hash Functions and MAC

Deadline: May 1, 2020

This lab is to be done individually.

Overview

The learning objective of this lab is for students to get familiar with one-way hash functions and Message Authentication Codes (MAC). After finishing the lab, in addition to gaining a deeper understanding of the concepts, students should be able to use tools to generate one-way hash value and MAC for a given message.

Reading Material

Hash Functions:

Please review this week's lectures and reading material.

HMAC:

We haven't covered HMAC in the lecture. A brief introduction to HMAC from Ross Anderson's Security Engineering book is as follows:

"Hash functions have many other uses. One of them is to compute MACs. A naive method would be to hash the message with a key: MACk(M) = h(k,M). However the accepted way of doing this, called HMAC, uses an extra step in which the result of this computation is hashed again. The two hashing operations are done using variants of the key, derived by exclusive-or'ing them with two different constants. Thus HMACk(M) = h(k⊕B,h(k⊕A,M)). A is constructed by repeating the byte 0x36 as often as necessary, and B similarly from the byte 0x5C. If a hash function is on the weak side, this construction can make exploitable collisions harder to find [888]. HMAC is now FIPS 198-1."

You are encouraged (but not required) to read further on HMACs.

Lab Setup

Note. This uses the same environment as the previous lab.

This lab requires the following tools:

1. Openssl
2. A hex editor (Ghex or Bless. You can also use an online hex editor)

You can either install these tools on your own or download the prepared VM from here or here. (If you have trouble setting up the VM please contact Mannan or your primary TA).

Note. The VM file you download is a virtual hard disk.
Note. VM username: seed , password: dees

Note. For those that cannot set up their VMs, ubuntu and macOS has openssl installed by default. You would only need to install a hex editor for the lab.

Lab Tasks

Task 1: Generating Message Digest and MAC

In this task, we will play with various one-way hash algorithms. To see the manuals, you can type "***man openssl***" and "***man dgst***". You can use the following openssl command to generate the hash value for a file:

"openssl dgst dgsttype filename"

Please replace the dgsttype with a specific one-way hash algorithm, such as "***-md5***", "***-sha1***", "***-sha256***", etc. In this task, you should try at least 3 different algorithms, and describe your observations. You can find the supported one-way hash algorithms by typing "***man openssl***".

Submit the following:

1. For a given file, what is the digest size **in bits** for each of the hash functions you tried?
2. Is the digest size different for files of different lengths?

Task 2: Keyed Hash and HMAC

In this task, we would like to generate a keyed hash (i.e. MAC) for a file. We can use the "***-hmac***" option (this option is currently undocumented, but it is supported byopenssl). The following example generates a keyed hash for a file using the HMAC-MD5 algorithm. The string following the "***-hmac***" option is the key.

"openssl dgst -md5 -hmac "abcdefg" filename"

Please generate a keyed hash using HMAC-MD5, HMAC-SHA256, and HMAC-SHA1 for any file that you choose. Please try several keys with different lengths.

1. Do we have to use a key with a fixed size in HMAC?
2. If so, what is the key size?
3. If not, why?

Task 3: The Randomness of One-way Hash

To understand the properties of one-way hash functions, we would like to do the following exercise for MD5 and SHA256:

1. Create a text file of any length.
2. Generate the hash value ***H1*** for this file using a specific hash algorithm.
3. Flip one bit of the input file. You can achieve this modification using ghex or Bless.
4. Generate the hash value ***H2*** for the modified file.
5. Please observe whether ***H1*** and ***H2*** are similar or not.
6. Please describe your observations in the lab report. You can write a short script to count how many bits are the same between ***H1*** and ***H2***.

Optional (Just for Fun):

Try out the MD5 Collisions Demo: https://www.mathstat.dal.ca/~selinger/md5collision/

This part is only for your own learning and requires no submission.

Submission

Take a screenshot of each completed part along with the required observations/findings in a .pdf file.
Submit the file as 2x100xxx.pdf on LMS by May 01, 2020.