ELEC-C7420 - Basic principles in networking

Part-II Security

Assignment II: Implementing Hash functions for Digital Signatures

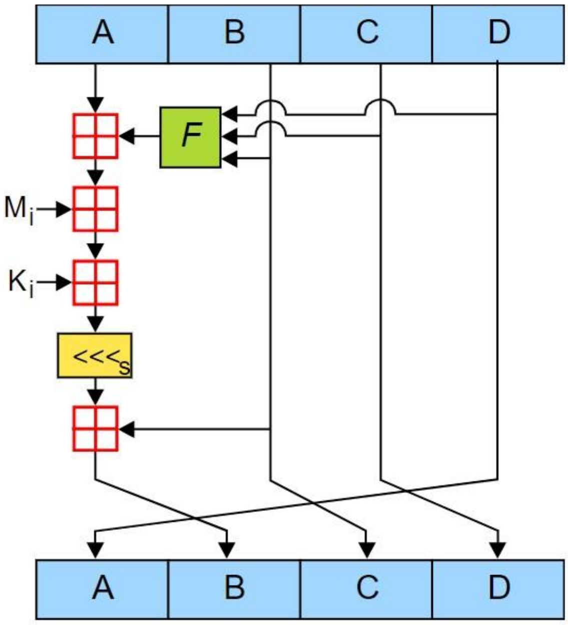
Digital Signatures

Hash functions and techniques may be used to perform digital signatures, in this experiment we will see in detail the implementation of two of these hashing techniques: MD5 and SHA-1.

1. Implement MD5

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits fewer than a multiple of 512. The remaining bits are filled up with 64 bits representing the length of the original message, modulo 264.

The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted A, B, C, and D. These are initialized to certain fixed constants. The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation.



MD5 operation
(source: https://en.wikipedia.org/wiki/MD5)

STEP-1: Read the 128-bit plain text.

STEP-2: Divide into four blocks of 32-bits named as A, B, C and D.

STEP-3: Compute the functions f, g, h and i with operations such as, rotations, permutations, etc.

STEP-4: The output of these functions are combined together as F and performed circular shifting and then given to key round.

STEP-5: Finally, right shift of 's' times are performed and the results are combined together to produce the final output.

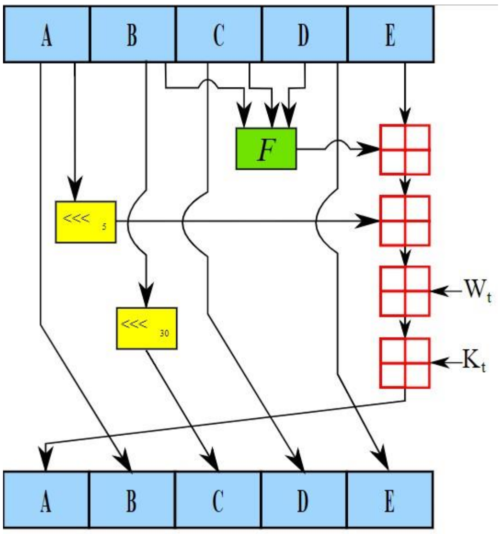
2. Implement SHA-1

[source: https://brilliant.org/wiki/secure-hashing-algorithms/#sha-1]

Secure Hash Algorithm 1, or SHA-1, was developed in 1993 by the U.S. government's standards agency National Institute of Standards and Technology (NIST). It is widely used in security applications and protocols, including TLS, SSL, PGP, SSH, IPsec, and S/MIME.

SHA-1 works by feeding a message as a bit string of length less than bits and producing a 160-bit hash value known as a message digest. Note that the message below is represented in hexadecimal notation for compactness.

There are two methods to encrypt messages using SHA-1. Although one of the methods saves the processing of sixty-four 32-bit words, it is more complex and time-consuming to execute, so the simple method is shown in the example below. At the end of the execution, the algorithm outputs blocks of 16 words, where each word is made up of 16 bits, for a total of 256 bits.



SHA-1 operation
[source: https://en.wikipedia.org/wiki/SHA-1]

STEP-1: Read the 256-bit key values.

STEP-2: Divide into five equal-sized blocks named A, B, C, D and E.

STEP-3: The blocks B, C and D are passed to the function F.

STEP-4: The resultant value is permuted with block E.

STEP-5: The block A is shifted right by 's' times and permuted with the result of step-4.

STEP-6: Then it is permuted with a weight value and then with some other key pair and taken as the first block.

STEP-7: Block A is taken as the second block and the block B is shifted by 's' times and taken as the **t**hird block.

STEP-8: The blocks C and D are taken as the block D and E for the final output.

3. Please answer the following questions in detail in the report

- Of the two mentioned hash function, would you use one for Security Application? Why? If not, provide an alternative.
- Please explain in brief what makes hash functions resistant to attacks. Provide an exemplary brief case study.
- Provide a comparison between MD5 and SHA-1. Overall, which one do you think performs better than the other one?
- What does it mean for a hash algorithm to be broken?

Report Structure

Please submit a written report. Suggested template for the report-

- Section 1: Goals of the experiment (What is the purpose and motivation behind the experiment).
- Section 2: Experimental Setup (Details of the experimental setup step by step)
- Section 3: Results & Conclusion (Please include snaps for each step containing proof of the successful implementation)
- Section 4: Answer of the given questions.
- Section 5: Annex (Please paste your sketches in this section)

Assessment Criteria (Total 10 Points)

- Implementing MD5 (3 Points)
- Implementing SHA-1 (3 Points)
- Answers of given questions (4 Points)

*****[Please, note that, for Arduino there are built in modules to implement hash functions. You can use these modules to ease up the pain.]**