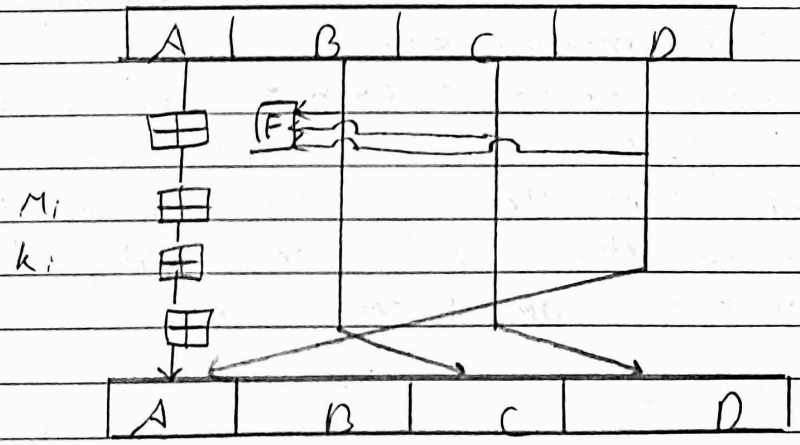
CSS lab 5

Aim: For varying message size, test integrity of message using MD-5, SHA-1 and analyze the performance of 2 protocols. Use Crypt APIs.

Theory:

1) MD5 (Message digest 5) is widely used cryptographic hash function with a 128 bit hash value.
2) MD5 hash is typically expressed as a 32 digit hexadecimal number.
3) MD5 processes a variable length message into a fixed length output of 128 bits.
4) 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 end of message.
5) This is followed by many zeros as are required to bring the length of the message upto 64 bits less than a multiple of 512.



6) One MD5 operation is shown above.

7) MD5 contains 64 of these operation, grouped in 4 rounds of 16 operations. F is a non linear function, one function is used to divide in each round.

8) There are 4 possible functions F; a different one is used in each round:

$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$

$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$

$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$

Algorithm.

1) Append padding bits, the message is "padded" (extended) so that its length is congruent to 448 modulo 512. That is the message is extended so that it is just 64 bits shy of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448 modulo 512.

2) Append length, A 64 bit representation of b is appended to the result of the previous step. In the unlikely event that b is greater than 2^{64}, then only lower order 64 bits of b are used.

A(0)

e) ONE OF 't' Consonants

3) Initialize the MD buffer. A 4 word buffer (A,B,C,D) is used to compute the message digest. Here each of A,B,C,D is a 32 bit register.

4) Process message on 16 word block. We first define four auxiliary functions that each take as input 3 32 bit words and produce as output one 32 bit word.

5) Output: The message digest. produced as output is A,B,C,D that is, we begin with the lower order byte of A and end with higher order byte of D.

→ SHA 1:

1) Secured hash algorithm
2) It works for any input message that is less than 2^{64} bits.
3) The o/p of SHA 1 is a message digest of 160 bits in length.

Algorithm:

1) Padding: Length of message is 64 bits short of multiple of 512 after padding.

2) Append a 64 bit length value of original message is taken.

3) Divide the I/P block into 512 bit blocks.

FOR EDUCATIONAL USE

4) Initialize CV 5-word (160-bit) buffer (A, B, C, D, E)

5) Process block: copy chaining variables A-E into variables a-e. Divide the current 512 bit block into 16 subblocks, each consisting of 32 bit. No of rounds is 4 and each contains 20 iterations (Total 80 iterations).

6) Output hash value in final buffer value

* Conclusion:

Thus, I have successfully studied and tested integrity of message using MD-5 and SHA-1 algorithm using python.