Sheet #6

1- Suppose Alice signs M Alice sends $S = [M]_{Alice}$ to Bob and Bob computes $M = \{S\}_{Alice}$ with Alice public key , Is it OK to just send S for verifying M integrity ? justify.

No, because there is no technique can be used in Bob's side for verifying integrity i.e. he must have the original message M to compare it with the signed message $[M]_{Alice}$.

2- Suppose that Alice signs M sends M and $S = [M]_{Alice}$ to Bob. State the advantages and disadvantages of this approach. Suggest a solution for this technique shortage.

Advantages: Integrity was achieved as M and S are now available to Bob.

Disadvantages: If M is big, $[M]_{Alice}$ is costly to compute and waste bandwidth to send M and $[M]_{Alice}$.

Solution: Alice signs h(M), where h(M) is much smaller than M

Alice sends M and $S = [h(M)]_{Alice}$ to Bob

Bob verifies that $h(M) = \{S\}_{Alice}$

3- State the main characteristics must be provided by any hash function.

Crypto hash function h(x) must provide

a. **Compression** — output length is small
b. **Efficiency** — h(x) easy to compute for any x
c. **One-way** — given a value y it is infeasible to find an x such that h(x) = y
d. **Weak collision resistance** — given x and h(x), infeasible to find $y \neq x$ such that h(y) = h(x)
e. **Strong collision resistance** — infeasible to find any x and y, with $x \neq y$ such that h(x) = h(y)
f. Lots of collisions exist, but hard to find one

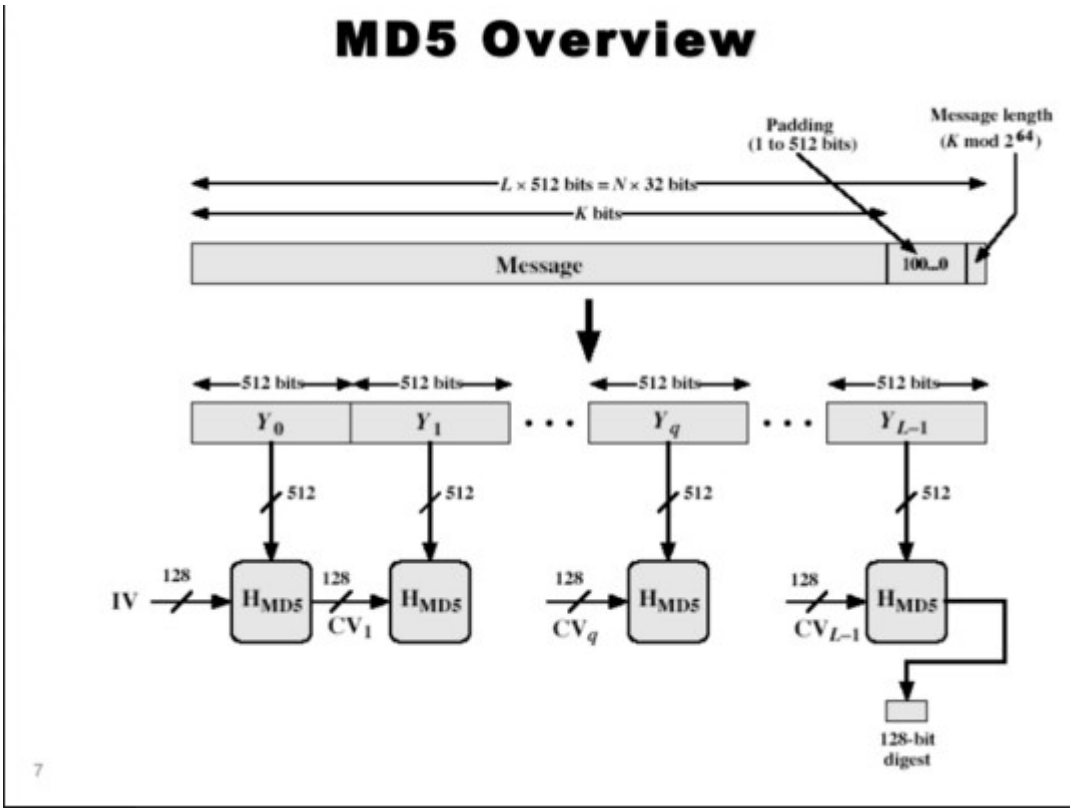
4- Why a hash function must finally collide?

Because it has a restricted output space and no restriction on the number of usage.

5- In terms of output space compare between **MD5 and SHA** ?

MD5 □ 128 bit output , SHA □160

6- Draw a general diagram for MD5 operation and write down the steps required to get Message digest using this algorithm.



Step 1. Append Padding Bits

The message is "padded" (extended) so that its length (in bits) 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.

Step 2. Append Length

A 64-bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step.

Step 3. Initialize MD Buffer

A four-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. These registers are initialized to the following values in hexadecimal, low-order bytes first):

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

Step 4. Process Message in 16-Word Blocks

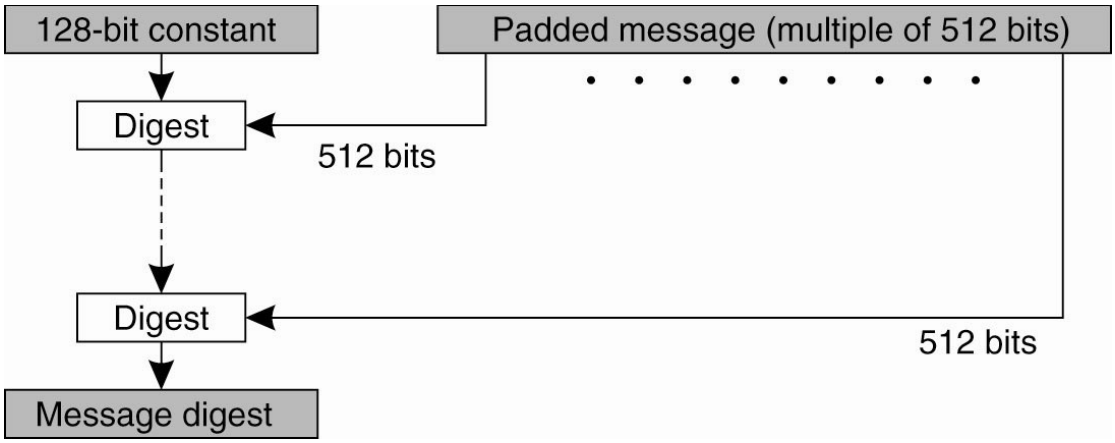
We first define four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word.

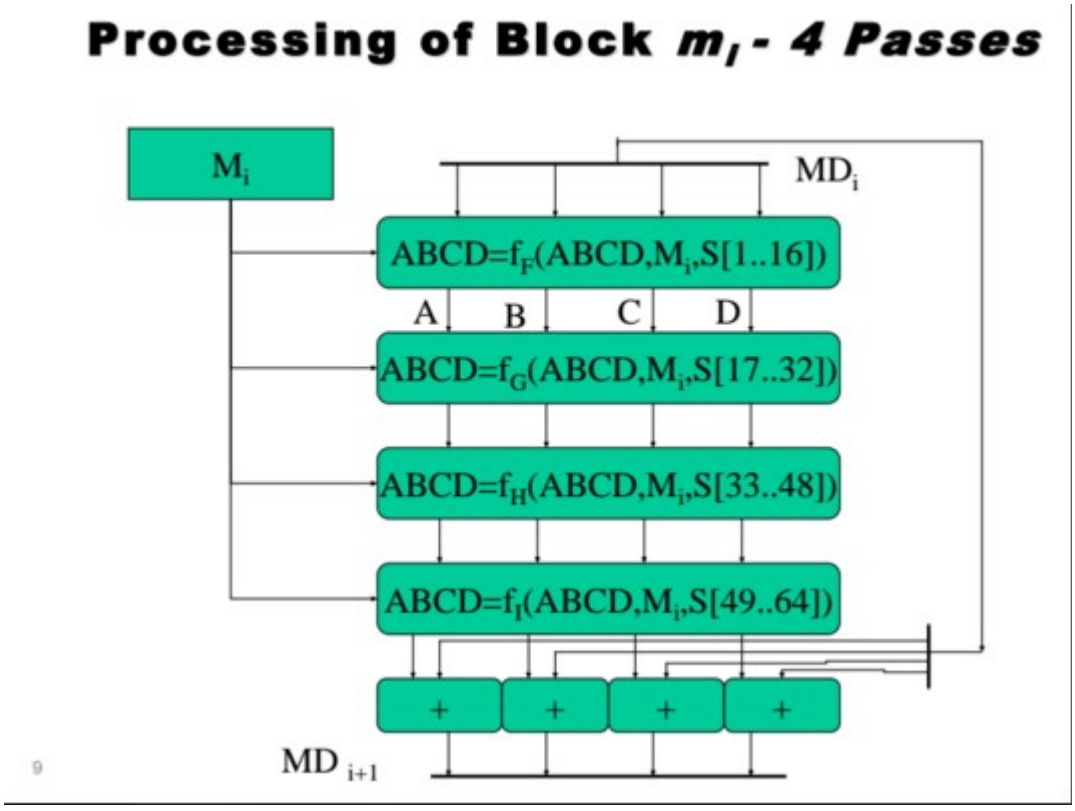
F(X,Y,Z) = XY v not(X) Z

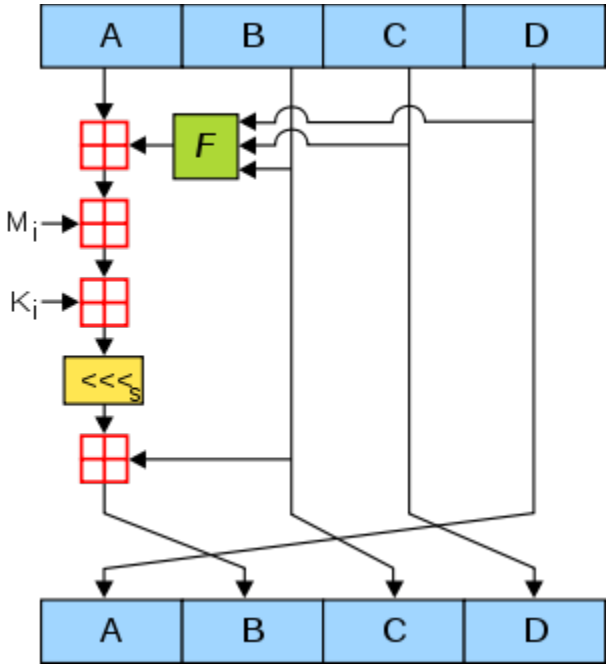
G(X,Y,Z) = XZ v Y not(Z)

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X v not(Z))







7- write down the steps required to get Message digest using SHA

SHA Overview

1. pad message so its length is 448 mod 512
2. append a 64-bit length value to message
3. initialise 5-word (160-bit) buffer (A,B,C,D,E)
   1. (67452301,efcdab89,98badcfe,10325476,c3d2e1f0)
4. process message in 16-word (512-bit) chunks:
   1. expand 16 words into 80 words by mixing & shifting
   2. use 4 rounds of 20 bit operations on message block & buffer
   3. add output to input to form new buffer value
5. output hash value is the final buffer value
6. Note that the SHA-1 Overview is very similar to that of MD5.

12

Best Regards