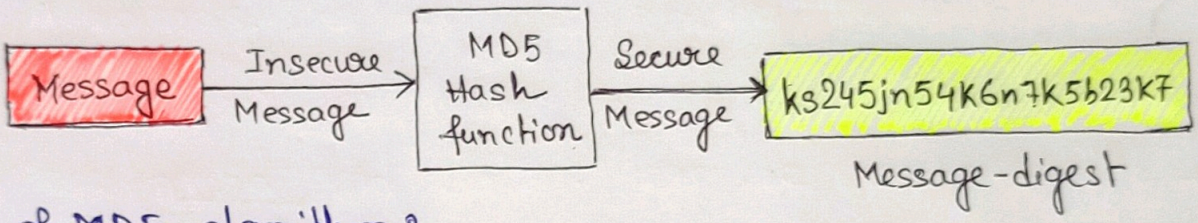
Message-digest algorithm (MD5):

MD5 is a cryptographic hash function algorithm that takes the message as input of any length and changes it into a fixed-length message of 16 bytes. MD5 algorithm stands for message-digest algorithm. MD5 was developed by Ronald Rivest as an improved version of MD4. The output of MD5 is always 128 bits.



Use of MD5 algorithm:

The use of MD5 algorithm is given as follows,

i) It is used for file authentication.

ii) In a web application, it is used for security purpose.

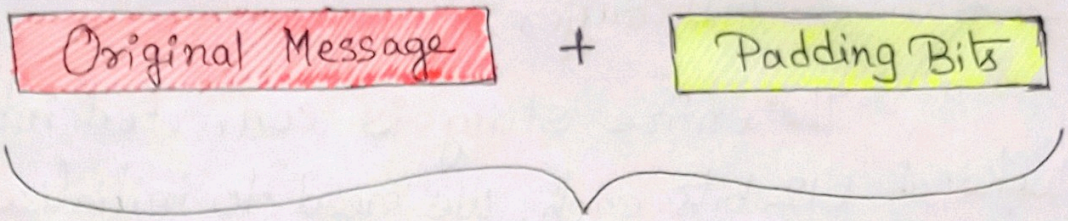
iii) Using this algorithm, password can be stored in 128 bits.

Working of MD5 Algorithm:

The working of MD5 algorithm follows the following steps,

1. Append Padding Bits:

In the first step, padding bits are added in the original message in such a way that the total length of the message is 64 bits less than the exact multiple of 512



Total length to be 64 bits less than multiple of 512

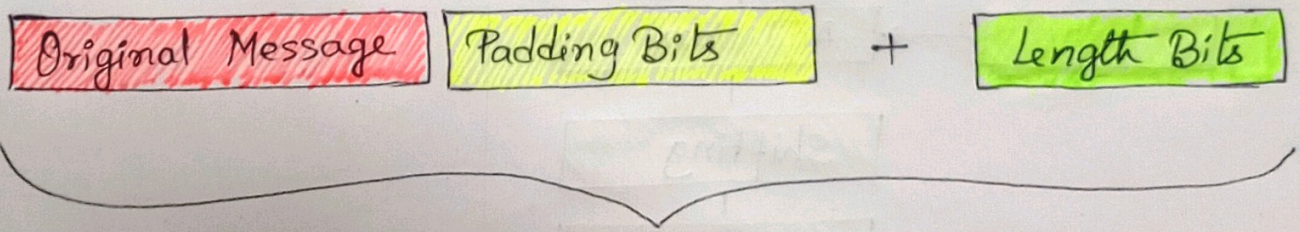
Suppose we have a message of 1000 bits. Now, here we will add 472 bits as padding bits to the original message. After adding the padding bits the size of the original message becomes 1472 bits, which is 64 bits less than an exact multiple of 512 (i.e $512 \times 3 = 1536$)

Length (original message + padding bits) = $512 * i - 64$

2. Append length bits:

In this step, we add the length bit in the output of the first step in such a way that the total number of the bits is the perfect multiple of 512.

Simply we add the 64-bits as a length bit in the output of the first step.



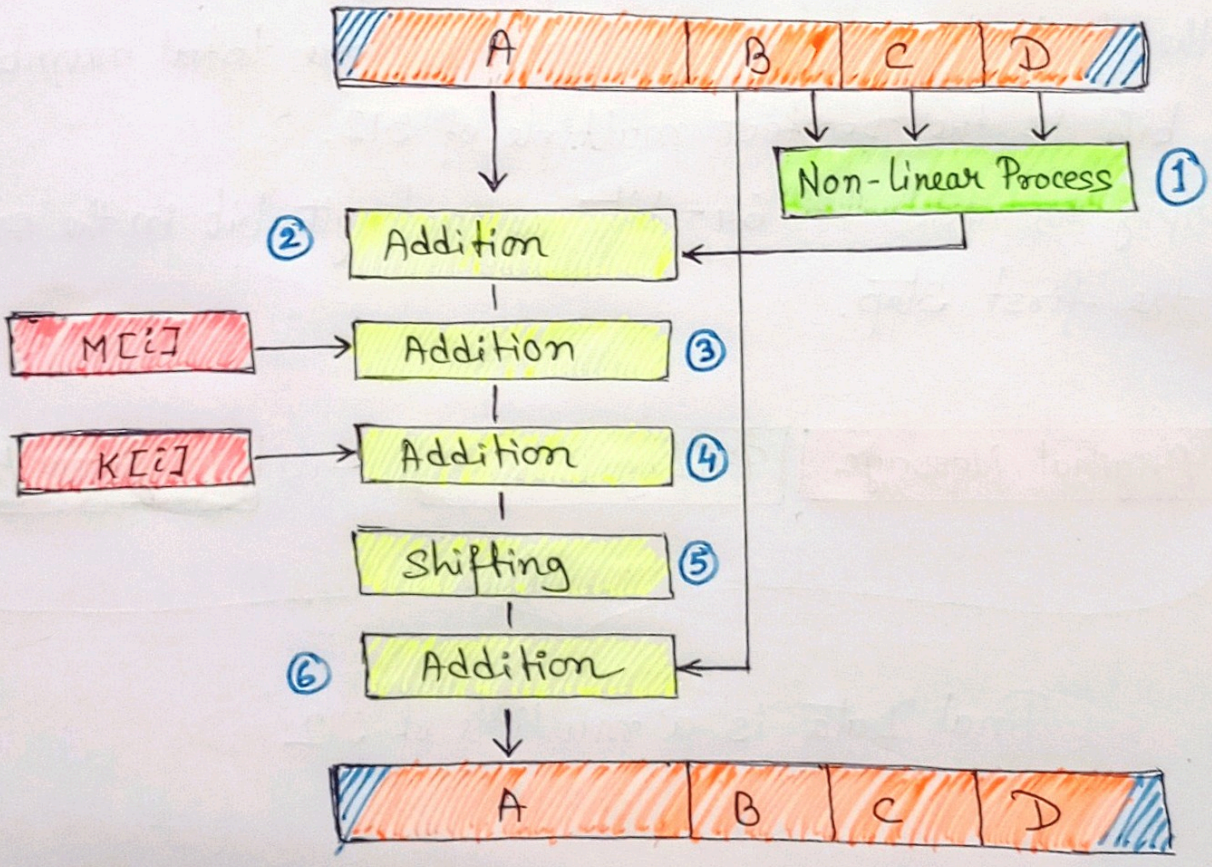
Final Data is a multiple of 512

3. Initialize MD Buffer:

The entire sting is converted into multiple blocks of 512 bits each. We need to initialize four different buffers (namely A, B, C and D). These four buffers are 32 bits each.

4. Process each 512-bits Block:

This is the most important step of the MD5 algorithm. Here a total of 64 operations are performed in 4 rounds. In the first round 16 operations will be performed, in the Second round 16 operations will be performed and so on. Here we consider M[i] as a 32 bits message and K[i] as a 32 bits constant.



According to the image we can define the processing of each buffer as,

1. The values of B, C and D is passed onto a non-linear process
2. The result is added with the value present at A
3. It adds the sub-block value (message) to the result above
4. Then it adds the constant value to the output of the above
5. A circular shift (left shift by n bits) is applied to the string.
6. As a final step the output of step 5 is added with the value of B and is stored in buffer A.

The non-linear process above is different for each round.

Round 1: (B AND C) OR ((NOT B) AND D)

Round 2: (B AND D) OR (C AND (NOT D))

Round 3: B XOR C XOR D

Round 4: C XOR (B OR (NOT D))