**Use of MD5 Algorithm**

It was developed with the main motive of security as it takes an input of any size and produces an output if a 128-bit hash value. To be considered cryptographically secure MD5 should meet two requirements:

1. It is impossible to generate two inputs that cannot produce the same hash function.
2. It is impossible to generate a message having the same hash value.

Initially, MD5 was developed to store one way hash of a password and some file servers also provide pre-computed MD5 checksum of a file so that the user can compare the checksum of the downloaded file to it. Most Unix based Operating Systems include MD5 checksum utilities in their distribution packages.

**How do the MD5 Algorithm works?**

As we all know that MD5 produces an output of 128-bit hash value. This [encryption](https://www.educba.com/what-is-encryption/) of input of any size into hash values undergoes 5 steps and each step has its a predefined task.

**Step1: Append Padding Bits**

* Padding means adding extra bits to the original message. So in MD5 original message is padded such that its length in bits is congruent to 448 modulo 512. Padding is done such that the total bits are 64 less being a multiple of 512 bits length.
* Padding is done even if the length of the original message is already congruent to 448 modulo 512. In padding bits, the only first bit is 1 and the rest of the bits are 0.

**Step 2: Append Length**

After padding, 64 bits are inserted at the end which is used to record the length of the original input. Modulo 2^64. At this point, the resulting message has a length multiple of 512 bits.

**Step 3: Initialize MD buffer**

A four-word buffer (A, B, C, D) is used to compute the values for the message digest. Here A, B, C, D are 32- bit registers and are initialized in the following way

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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: Processing message in 16-word block**

MD5 uses the auxiliary functions which take the input as three 32-bit number and produces a 32-bit output. These functions use logical operators like OR, XOR, NOR.

|  |  |
| --- | --- |
| 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)) |

The content of four buffers are mixed with the input using this auxiliary buffer and 16 rounds are performed using 16 basic operations.

**Output-**

After all, rounds have performed the buffer A, B, C, D contains the MD5 output starting with lower bit A and ending with higher bit D.