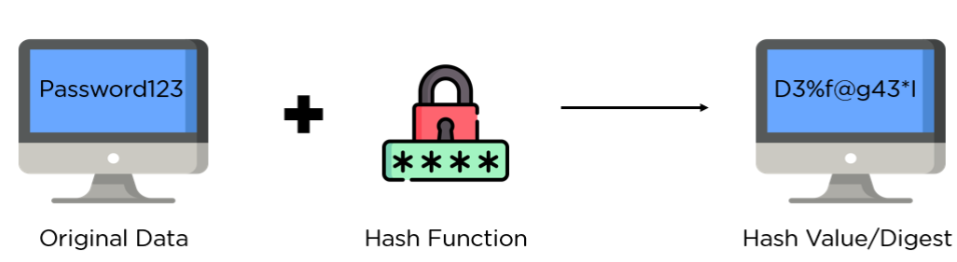
**The cryptographic hash function SHA-256**

**Hashing:**

* Hashing is the process of scrambling raw information to the extent that it cannot reproduce it back to its original form.
* It takes a piece of information and passes it through a function that performs mathematical operations on the plaintext.

This function is called the hash function, and the output is called the hash value/digest.



There are two primary applications of hashing:

* **Password Hashes:** It converts user passwords into a hash value before being stored on the server. It compares the hash value re-calculated during login to the one stored in the database for validation.
* **Integrity Verification:** When it uploads a file to a website, it also shared its hash as a bundle. When a user downloads it, it can recalculate the hash and compare it to establish data integrity.

**SHA-256** (secure hash algorithm, FIPS 182-2)  **:**

* SHA-256 is a cryptographic and key less hash function that is, an MDC (Manipulation Detection Code) with digest length of 256 bits.
* Published in 2001, through the joint effort between the NSA and NIST to introduce a successor to the SHA 1 family, which was slowly losing strength against brute force attacks.
* A message is processed by blocks of 512 = 16 × 32 bits, each block requiring 64 rounds.
* **Cryptographic Hash Function**: SHA-256 is a cryptographic hash function that converts input data into a fixed-length 256-bit string, ensuring data integrity and security.
* **Data Integrity**: SHA-256 ensures that even a minor change in input data generates a completely different hash, making it a reliable tool for verifying data integrity.
* **Applications**: SHA-256 is used in blockchain technology, digital signatures, and password hashing to protect data from tampering and unauthorized access.
* **Security Features**: The algorithm is resistant to collision attacks, meaning it is extremely difficult to find two different inputs that produce the same hash value.
* **Efficiency**: Despite its complexity, SHA-256 is computationally efficient, allowing for quick generation of hash values and faster data processing.

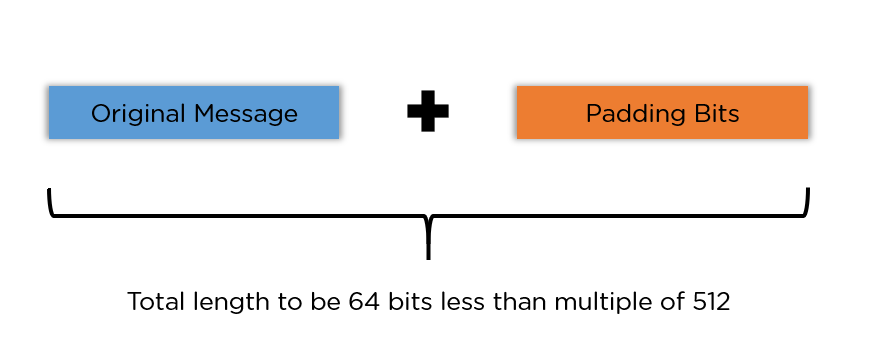
**features of the SHA algorithm**:

* **Message Length:** The length of the cleartext should be less than 264 bits. The size needs to be in the comparison area to keep the digest as random as possible.
* **Digest Length:** The length of the hash digest should be 256 bits in SHA 256 algorithm, 512 bits in SHA-512, and so on. Bigger digests usually suggest significantly more calculations at the cost of speed and space.
* **Irreversible:** By design, all hash functions such as the SHA 256 are irreversible. You should neither get a plaintext when you have the digest beforehand nor should the digest provide its original value when you pass it through the hash function again.

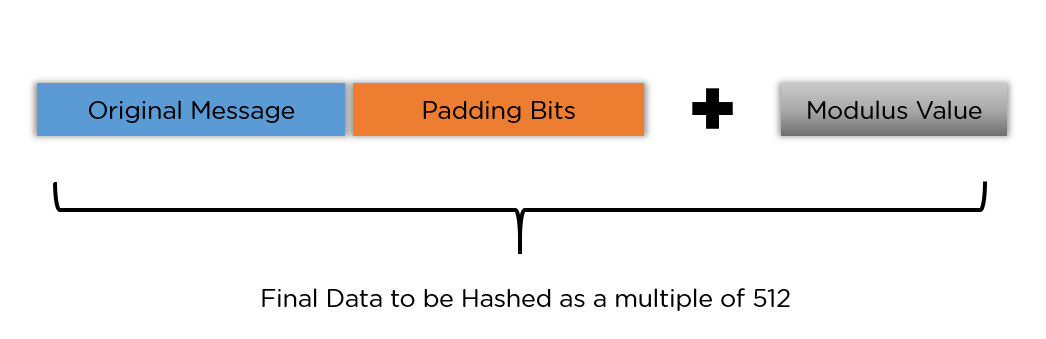
**Steps in SHA-256 Algorithm**

The complete process can be divided into following five different segments

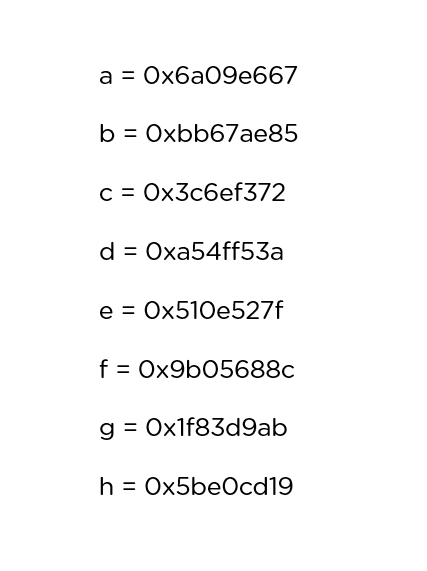
**Padding Bits:** It adds some extra bits to the message, such that the length is exactly 64 bits short of a multiple of 512. During the addition, the first bit should be one, and the rest of it should be filled with zeroes.



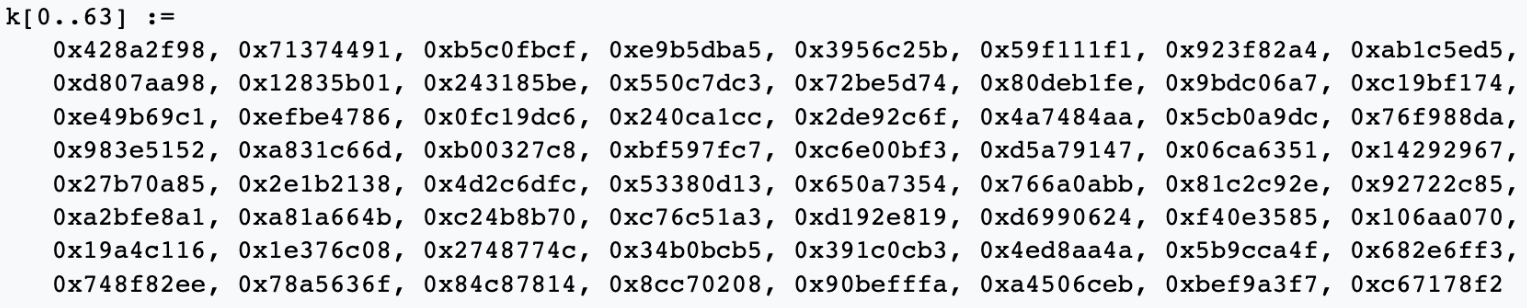
**Padding Length:** You can add 64 bits of data now to make the final plaintext a multiple of 512. You can calculate these 64 bits of characters by applying the modulus to your original cleartext without the padding.



**Initialising the Buffers:** You need to initialize the default values for eight buffers to be used in the rounds as follows:

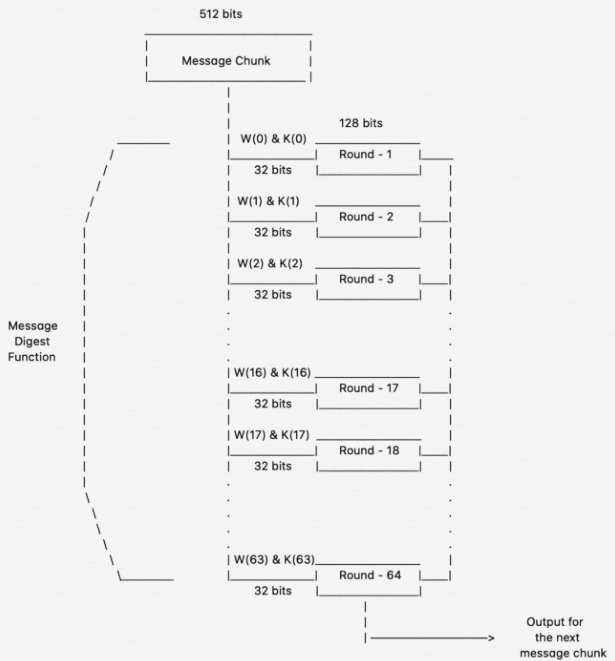


You also need to store 64 different keys in an array, ranging from K[0] to K[63]. They are initialized as follows:



**Compression Functions**

The entire message gets broken down into multiple blocks of 512 bits each. It puts each block through 64 rounds of operation, with the output of each block serving as the input for the following block. The entire process is as follows:



Courtesy: [Medium Article on SHA 256](https://infosecwriteups.com/breaking-down-sha-256-algorithm-2ce61d86f7a3)

While the value of K[i] in all those rounds is pre-initialized, W[i] is another input that is calculated individually for each block, depending on the number of iterations being processed at the moment.

**Output**

With each iteration, the final output of the block serves as the input for the next block. The entire cycle keeps repeating until you reach the last 512-bit block, and you then consider its output the final hash digest. This digest will be of the length 256-bit, as per the name of this algorithm.

With the SHA 256 algorithm being implemented thoroughly since the early 90s, there are specific applications that you can look into. You will see them in the next section.

**Applications of SHA algorithm**

* **Digital Signature Verification:** Digital signatures follow asymmetric encryption methodology to verify the authenticity of a document/file. Hash algorithms like SHA 256 go a long way in ensuring the verification of the signature.
* **Password Hashing:** As discussed above, websites store user passwords in a hashed format for two benefits. It helps foster a sense of privacy, and it lessens the load on the central database since all the digests are of similar size.
* **SSL Handshake:** The SSL handshake is a crucial segment of the web browsing sessions, and it’s done using SHA functions. It consists of your web browsers and the web servers agreeing on encryption keys and hashing authentication to prepare a secure connection.
* **Integrity Checks:** As discussed above, verifying file integrity has been using variants like SHA 256 algorithm and the MD5 algorithm. It helps maintain the full value functionality of files and makes sure they were not altered in transit.

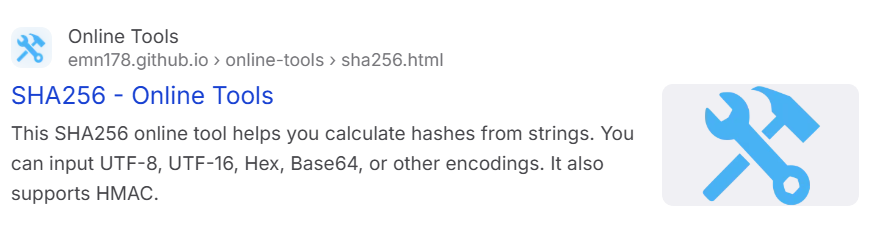
**Conclusion**

* It is widely used in cryptography and data integrity verification due to its security and resistance to collision attacks, making it suitable for applications such as digital signatures, data authentication, and password hashing.
* SHA-256 is known for its high level of security, computational efficiency, and the fact that it is practically impossible to derive the original data from its hash value.

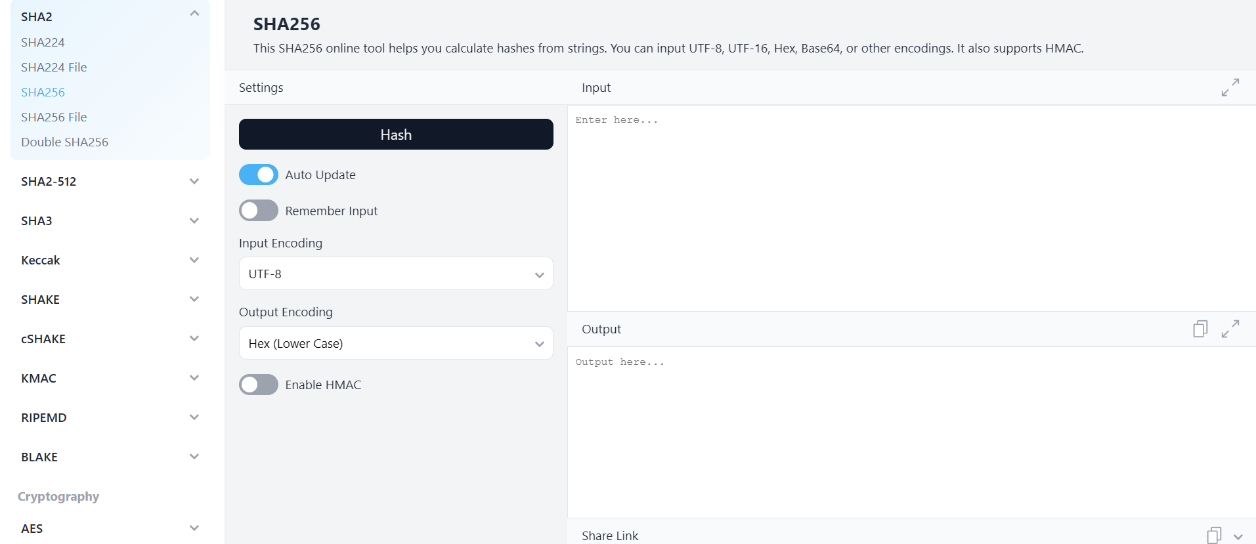
**Manipulation Detection Code**

* Manipulation Detection Codes (MDC) are a class of checksum algorithms designed to detect both accidental and malicious modifications of an electronic message or document.
* These codes do not require the use of secret information such as a cryptographic key, making them useful for implementing encryption and message authentication in different protocol layers without key management difficulties. It is recommended that cryptographic checksums intended to detect fraudulent messages should be around 128 bits in length.

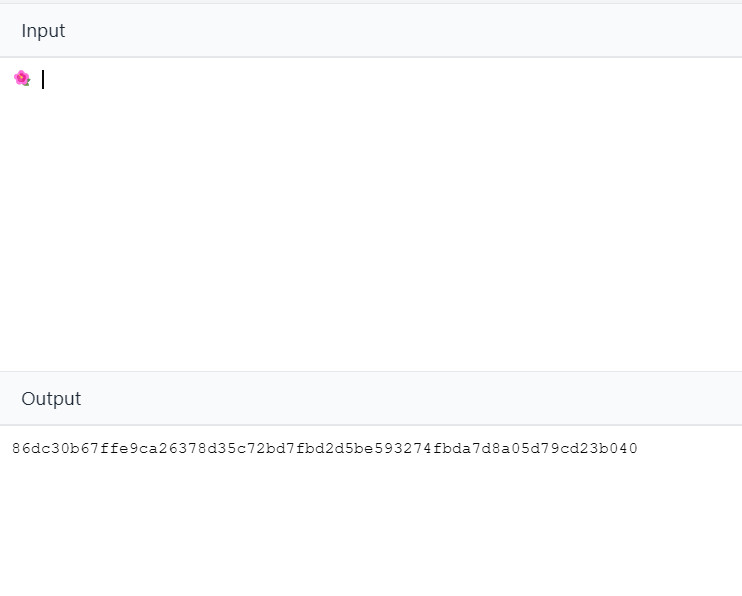
**SHA256 Simulation:**



SHA256 when there is no data:



Change in SHA when there is data:



Change in SHA with change in data:



Change in SHA when data is updated:



---------