

Report on SHA256

Understanding the SHA-256 Cryptographic Hash Function

Overview of Hashing

Hashing refers to the process of transforming readable data into a scrambled, irreversible format, making it impossible to reconstruct the original input.

It works by passing the data through a mathematical function that performs specific operations on the plaintext.

This function is called a hash function, and its output is referred to as the hash value or digest.

Main Uses of Hashing

- **Password Storage:** User passwords are converted into hash values before being stored. During login, the system hashes the entered password and compares it with the stored hash for authentication.
- **Data Integrity Checks:** When a file is provided for download, its hash value is often shared alongside it. The recipient can recompute the hash after download and compare it to verify the file's integrity.

SHA-256 (Secure Hash Algorithm, FIPS 182-2)

- SHA-256 is a keyless cryptographic hash function, categorized as a Manipulation Detection Code (MDC), producing a 256-bit digest.
- Introduced in 2001 by NIST and the NSA as a stronger replacement for SHA-1, which was losing resistance against brute-force attacks.
- It processes data in 512-bit blocks (16×32 -bit words), executing 64 computational rounds per block.
- Designed to generate a fixed-length 256-bit output from any size input, ensuring data integrity and security.

- Even the smallest change in input drastically changes the output hash (avalanche effect).
- Applications: Commonly used in blockchain, password hashing, digital signatures, and data authentication.
- Security: Strong resistance to collisions (two different inputs producing the same hash).
- Efficiency: Capable of fast computation without sacrificing security.

Core Characteristics of SHA Family

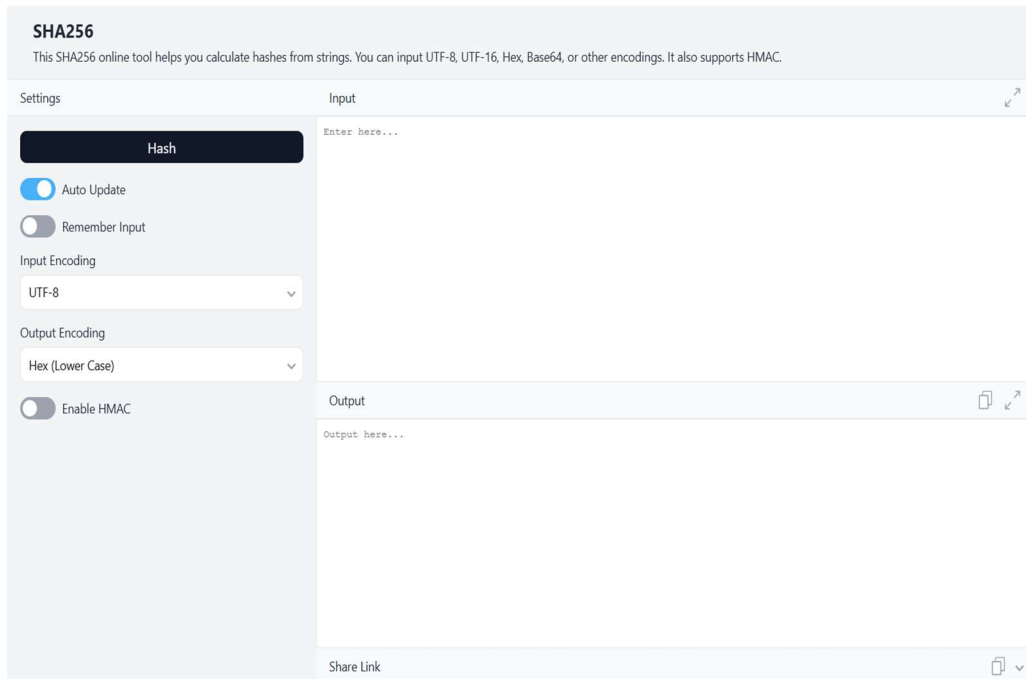
- Message Size: The original data must be less than 2^{64} bits for SHA-256.
- Digest Size: Output length is fixed (256 bits for SHA-256, 512 bits for SHA-512, etc.).
- Irreversibility: Once hashed, it's computationally infeasible to retrieve the original data from the digest.

Steps in the SHA-256 Algorithm

- Bit Padding: Extend the message length so that it is exactly 64 bits short of a multiple of 512. The first added bit is '1', followed by enough '0's to meet the length requirement.
- Length Padding: Append 64 bits representing the original message length (before padding) so the total size is a multiple of 512.
- Buffer Initialization: Initialize eight constant 32-bit buffers to predefined values.
- Round Keys Setup: Prepare an array of 64 constants ($K[0]$ to $K[63]$) used in each round of processing.
- Compression Function:
 - Split the padded message into 512-bit chunks.
 - Process each chunk through 64 rounds of operations, combining the data with buffer values and round keys.
 - The output of each chunk becomes the input for the next chunk.
- Final Output: Once all chunks are processed, the final concatenated values of the buffers form the 256-bit hash.

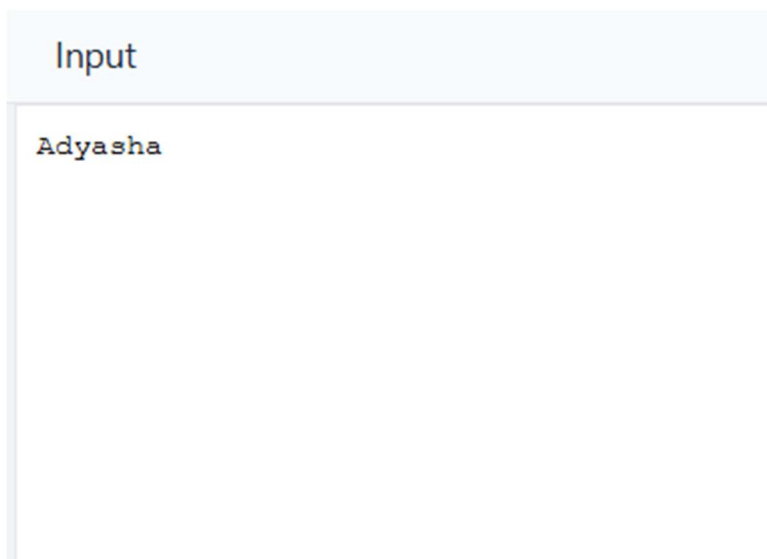
SHA256 Simulation

1.The interface is displayed before any input is entered. The tool allows the user to type text into an “Input” field, which will then be hashed in real-time when the “Auto Update” feature is enabled.



The screenshot shows the SHA256 online tool interface. At the top, the title "SHA256" is displayed, followed by a description: "This SHA256 online tool helps you calculate hashes from strings. You can input UTF-8, UTF-16, Hex, Base64, or other encodings. It also supports HMAC." Below this, the interface is divided into two main sections: "Settings" on the left and "Input" on the right. The "Settings" section includes a "Hash" button, a toggle for "Auto Update" (which is turned on), a toggle for "Remember Input" (which is turned off), a dropdown for "Input Encoding" set to "UTF-8", a dropdown for "Output Encoding" set to "Hex (Lower Case)", and a toggle for "Enable HMAC" (which is turned off). The "Input" section has a text field labeled "Enter here..." and a "Share Link" button at the bottom right. The "Output" section, located below the input field, has a text field labeled "Output here..." and a "Share Link" button at the bottom right.

2.A user-provided input, "Adyasha", is entered into the input field.



The screenshot shows a close-up of the input field. The text "Adyasha" is entered into the field. The field is labeled "Input" at the top.

3. The tool instantly processes this text through the SHA-256 algorithm and produces a fixed-length 256-bit hash value, represented in hexadecimal form

Output

```
63e76be23f16758c26096aba18a6fdffb1d420204f56b7ae2f18f249ae313cfc
```

Conclusion

SHA-256 remains one of the most secure and widely adopted hashing algorithms due to its resistance to collisions, computational efficiency, and irreversibility. It is an essential tool in cryptography, ensuring data authenticity and protection in various security protocols.