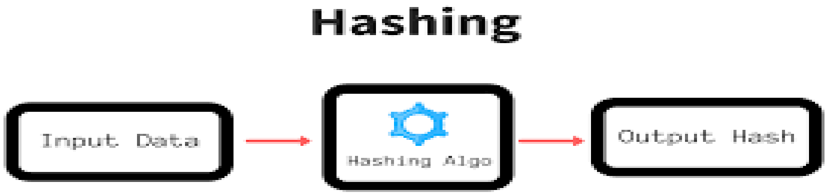
What is Hashing?

Hashing is a method of cryptography that converts any form of data into a unique string of text. Any piece of data can be hashed, no matter its size or type. In traditional hashing, regardless of the data's size, type, or length, the hash that any data produces is always the same length. A hash is designed to act as a one-way function—you can put data into a hashing algorithm and get a unique string, but if you come upon a new hash, you cannot decipher the input data it represents. A unique piece of data will always produce the same hash.

Hashing is mapping data of any length to a fixed-length output using an **algorithm**. Typically, the hashing algorithm most people know of is SHA-2 or SHA-256. That's because it's the current standard for SSL encryption.



What is an MD5 hash?

An MD5 hash is created by taking a string of an any length and encoding it into a 128-bit fingerprint. Encoding the same string using the MD5 algorithm will always result in the same 128-bit hash output. MD5 hashes are commonly used with smaller strings when storing passwords, credit card numbers or other sensitive data in databases such as the popular MySQL. This tool provides a quick and easy way to encode an MD5 hash from a simple string of up to 256 characters in length.

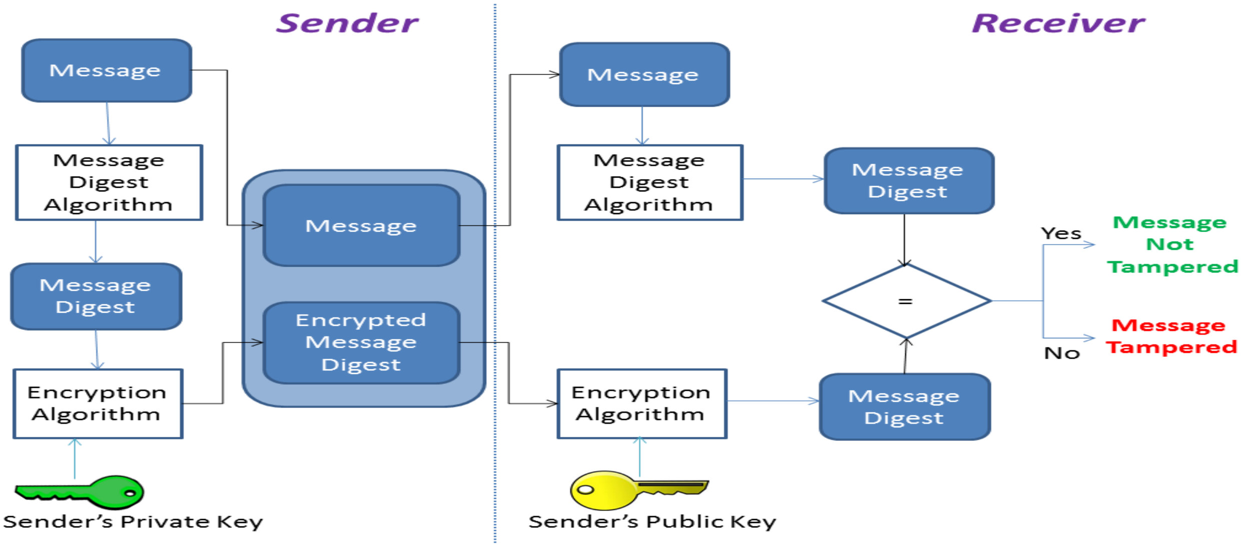
MD5 hashes are also used to ensure the data integrity of files. Because the MD5 hash algorithm always produces the same output for the same given input, users can compare a hash of the source file with a newly created hash of the destination file to check that it is intact and unmodified.

An MD5 hash is NOT encryption. It is simply a fingerprint of the given input. However, it is a one-way transaction and as such it is almost impossible to reverse engineer an MD5 hash to retrieve the original string.

How MD5 works

The MD5 message digest hashing algorithm processes data in 512-bit blocks, broken down into 16 words composed of 32 bits each. The output from MD5 is a 128-bit message digest value.

Computation of the MD5 digest value is performed in separate stages that process each 512-bit block of data along with the value computed in the preceding stage. The first stage begins with the message digest values initialized using consecutive hexadecimal numerical values. Each stage includes four message digest passes which manipulate values in the current data block and values processed from the previous block. The final value computed from the last block becomes the MD5 digest for that block.



Your Hash: 97cddd635cef02b3ceaf25641f9b2eee
Your String: Avinash

MD5 security

The goal of any message digest function is to produce digests that appear to be random. To be considered cryptographically secure, the hash function should meet two requirements: first, that it is impossible for an attacker to generate a message matching a specific hash value; and second, that it is impossible for an attacker to create two messages that produce the same hash value.

Secure Sockets Layer

SSL stands for **Secure Sockets Layer** and, in short, it's the standard technology for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems, preventing criminals from reading and modifying any information transferred, including potential personal details. The two systems can be a server and a client (for example, a shopping website and browser) or server to server (for example, an application with personal identifiable information or with payroll information).

It does this by making sure that any data transferred between users and sites, or between two systems remain impossible to read. It uses encryption algorithms to scramble data in transit, preventing hackers from reading it as it is sent over the connection. This information could be anything sensitive or personal which can include credit card numbers and other financial information, names and addresses.

Transport Layer Security

TLS (Transport Layer Security) is just an updated, more secure, version of SSL. We still refer to our security certificates as SSL because it is a more commonly used term, but when you are buying SSL from Symantec you are actually buying the most up to date TLS certificates with the option of ECC, RSA or DSA encryption.

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. When secured by TLS, connections between a client (e.g., a web browser) and a server should have one or more of the following properties:

The connection is private (or secure) because symmetric cryptography is used to encrypt the data transmitted. The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret that was negotiated at the start of the session. The server and client negotiate the details of which encryption algorithm and cryptographic keys to use before the first byte of data is transmitted .

The negotiation of a shared secret is both secure (the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themselves in the middle of the connection) and reliable (no attacker can modify the communications during the negotiation without being detected).

The identity of the communicating parties can be authenticated using public-key cryptography. This authentication can be made optional, but is generally required for at least one of the parties (typically the server).

The connection is reliable because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.

Hyper Text Transport Layer Security

HTTPS (Hyper Text Transfer Protocol Secure) appears in the URL when a website is secured by an SSL certificate. The details of the certificate, including the issuing authority and the corporate name of the website owner, can be viewed by clicking on the lock symbol on the browser bar.
HTTPS ensures data security over the network - mainly public networks like Wi-Fi. HTTP is not encrypted and is vulnerable to attackers who are eavesdropping and can gain access to website database and sensitive information. By virtue, HTTPS encryption is done bi-directionally, which means that the data is encrypted at both the client and server sides. Only the client can decode the information that comes from the server. So, HTTPS does encryption of data between a client and a server, which protects against eavesdropping, forging of information and tampering of data. But how do you ensure if you are seeing an HTTPS-enabled web page? Just check the address bar that carries the site name against different background colours with a lock icon at the left corner. However, this design can be different for different browsers. For example, consider going to a bank website, say hdfcbank.com. A non-secured HTTP will open up. But when we go to the login page, we can see an HTTPS in the address bar with some specific design. Implementation: HTTPS is mainly used by those websites which deal with monetary transactions or transfer user's personal data which could be highly sensitive. Banking websites are common examples. In layman's terms, HTTPS ensures that users watch websites that they want to watch. Data exchanged between the user and the website is not read, stolen or tampered with by a third party. But it can't encrypt everything - it has some limitations too. For example, HTTPS can't encrypt host addresses and port numbers.