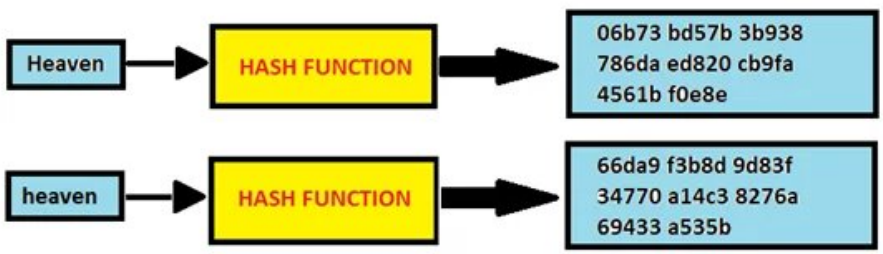
What is SHA? What is SHA used for?

Table of Contents

Introduction to SHA

SHA stands for secure hashing algorithm. SHA is a modified version of MD5 and used for hashing data and **certificates**. A hashing algorithm shortens the input data into a smaller form that cannot be understood by using bitwise operations, modular additions, and compression functions. You may be wondering, can hashing be cracked or decrypted? Hashing is similar to **encryption**, the only difference between hashing and encryption is that hashing is one-way, meaning once the data is hashed, the resulting hash digest cannot be cracked, unless a brute force attack is used. See the image below for the working of SHA algorithm. SHA works in such a way even if a single character of the message changed, then it will generate a different hash. For example, hashing of two similar, but different messages i.e., Heaven and heaven is different. However, there is only a difference of a capital and small letter.



The initial message is hashed with SHA-1, resulting in the hash digest "06b73bd57b3b938786daed820cb9fa4561bf0e8e". If the second, similar, message is hashed with SHA-1, the hash digest will look like "66da9f3b8d9d83f34770a14c38276a69433a535b". This is referred to as the avalanche effect. This effect is important in cryptography, as it means even the slightest change in the input message completely changes the output. This will stop attackers from being able to understand what the hash digest originally said and telling the receiver of the message whether or not the message has been changed while in transit.

SHAs also assist in revealing if an original message was changed in any way. By referencing the original hash digest, a user can tell if even a single letter has been changed, as the hash digests will be completely different. One of the most important parts of SHAs are that they are deterministic. This means that as long as the hash function used is known, any computer or user can recreate the hash digest. The determinism of SHAs is one of reasons every SSL certificate on the Internet is required to have been hashed with a SHA-2 function.

Different SHA Forms

When learning about SHA forms, several different types of SHA are referenced. Examples of SHA names used are SHA-1, SHA-2, SHA-256, SHA-512, SHA-224, and SHA-384, but in actuality there are only two types: SHA-1 and SHA-2. The other larger numbers, like SHA-256, are just versions of SHA-2 that note the bit lengths of the SHA-2. SHA-1 was the original secure hashing algorithm, returning a 160-bit hash digest after hashing. Someone may wonder, can SHA-2 be cracked like SHA-1? The answer is yes. Due to the short length of the hash digest, SHA-1 is more easily brute forced than SHA-2, but SHA-2 can still be brute forced. Another issue of SHA-1 is that it can give the same hash digest to two different values, as the number of combinations that can be created with 160 bits is so small. SHA-2 on the other hand gives every digest a unique value, which is why all certificates are required to use SHA-2.

SHA-2 can produce a variety of bit-lengths, from 256 to 512 bit, allowing it to assign completely unique values to every hash digest created. Collisions occur when two values have the same hash digest. SHA-1 can easily create collisions, making it easier for attackers to get two matching digests and recreate the original **plaintext** Compared to SHA-1, SHA-2 is much more secure and has been required in all digital signatures and certificates since 2016. Common attacks like brute force attacks can take years or even decades to crack the hash digest, so SHA-2 is considered the most secure hash algorithm.

What SHA is used for and Why

As previously mentioned, Secure Hashing Algorithms are required in all digital signatures and certificates relating to SSL/TLS connections, but there are more uses to SHAs as well. Applications such as **SSH**, S-MIME (Secure / Multipurpose Internet Mail Extensions), and IPSec utilize SHAs as well. SHAs are also used to hash passwords so that the server only needs to remember hashes rather than passwords. In this way, if an attacker steals the database containing all the hashes, they would not have direct access to all of the plaintext passwords, they would also need to find a way to crack the hashes to be able to use the passwords. SHAs can also work as indicators of a file's integrity. If a file has been changed in transit, the resulting hash digest created from the hash function will not match the hash digest originally created and sent by the file's owner.

We have now learned what SHAs are used for, but why use a Secure Hashing Algorithm in the first place? A common reason is their ability to stop attackers. Though some methods, like brute force attacks, can reveal the plaintext of the hash digests, these tactics are made extremely difficult by SHAs. A password hashed by a SHA-2 can take years, even decades to break, thus wasting resources and time on a simple password, which may turn many attackers away. Another reason to use SHAs is the uniqueness of all the hash digests. If SHA-2 is used, there will likely be few to no collisions, meaning a simple change of one word in a message would completely change the hash digest. Since there are few or no collisions, a pattern cannot be found to make breaking the Secure Hashing Algorithm easier for the attacker. These are just a few reasons why SHA is used so often.

SHA 2 Limitations

Browser Support

Browser	Minimum Browser Version
Chrome	26+
Firefox	1.5+
Internet Explorer	6+ (With XP SP3+)
Netscape	7.1+
Safari	3+ (Ships with OS X 10.5)
Mozilla	1.4+
Opera	9.0+

Server Support

Server	Minimum Server Version
AWS (Amazon Web Services)	YES
Apache	2.0.63+ w/ OpenSSL 0.9.8o+

Server	Minimum Server Version
Cisco ASA 5500	8.2.3.9+ for AnyConnect VPN Sessions; 8.4(2)+ for other functionalities
Java based products	Java 1.4.2+
IBM Domino Server	9.0+ (Bundled with HTTP 8.5+)
IBM HTTP Server	8.5+ (Bundled with Domino 9+)
IBM z/OS	v1r10+
OpenSSL based products	OpenSSL 0.9.8o+
Oracle Wallet Manager	11.2.0.1+
Oracle Weblogic	10.3.1+
Web Sphere MQ	7.0.1.4+

OS Support

Operating System	SSL Certificate Minimum OS Version	Client Certificate Minimum OS Version
Android	2.3+	2.3+
iOS	3.0+	3.0+
ChromeOS	YES	YES
Mac OS X	10.5+	10.5+
Windows XP	SP3+ XP	SP3+ (partial)
Windows Server	2003 SP2 +Hotfixes (Partial)	2003 SP2 +Hotfixes (Partial)
Windows Phone	7+	7+
Blackberry	5.0+	5.0+

The Future of Hashing

At this point in time, SHA-2 is the industry standard for hashing algorithms, though SHA-3 may eclipse this in the future. SHA-3 was released by the NIST, which also created SHA-1 and SHA-2, in 2015 but was not made the industry standard for many reasons. During the release of SHA-3, most companies were in the middle of migrating from SHA-1 to SHA-2, so switching right on to SHA-3 while SHA-2 was still very secure did not make sense. Along with this, SHA-3 was seen as slower than SHA-2, although this is not exactly the case. SHA-3 is slower on the software side, but it is much faster than SHA-1 and SHA-2 on the hardware side, and is getting faster every year. For these reasons, we will likely see the move to SHA-3 later on down the line, once SHA-2 becomes unsafe or deprecated.

Certificates ›

Encryption Basics ›

Cloud Key Management Service Options ›

Public Key Infrastructure (PKI) ›

Common Encryption Algorithms ›

Comparisons ›

Regulations, Standards & Compliance ›

DevOps ›

Bring your own Key Terminology ›

Cybersecurity Frameworks ›

Key Management Interoperability Protocol ›

IoT ›

Ready to get started?

Get more information about one of the fastest growing new attack vectors, latest cyber security news and why securing keys and certificates is so critical to our Internet-enabled world.

REQUEST A CONSULTATION

Global Headquarters

130 N Preston Rd,
Prosper, TX 75078, USA

Contact Sales

info@encryptionconsulting.com

+1- 469-815-4136

Services

Encryption Advisory Services

Hardware Security Modules

Public Key Infrastructure Services

Certificate Lifecycle Management

Enterprise Encryption Platforms

Cloud Data Protection Services

Data Loss Prevention (DLP) Services

Cloud Access Security Broker (CASB) Services

Products

Code Signing Solution – CodeSign Secure

PKI-as-a-Service

Certificate Management Solution – CertSecure Manager

HSM-as-a-Service

Resources

Education Center

Training

Company

About Us

Careers

Global Partners

Newsroom