SHA_2

SHA-2 is a hash function that plays a critical role in much of our online security.

History of SHA-2:

The predecessors to SHA-2 were important stepping stones in arriving at the current algorithm. The National Institute of Standards and Technology (NIST) published the federal information processing standard (FIPS) 180 in 1993 for what we now refer to as SHA-0 in 1993. Weaknesses in the algorithm were discovered relatively quickly, so the algorithm was revised and an updated version, SHA-1, was released in 1995 with FIPS 180-1.

SHA-1 was also specified in RFC 3174.It was modeled closely on MD4, the 4th version of Ron Rivest's Message Digest hashing algorithm.

The hash of SHA-1 is only 160 bits in length, which began posing security problems as technology and cryptographic techniques improved over time. This led NIST to introduce another update, which was outlined in FIPS 180-2. This document set out three versions of SHA:

- SHA-256
- SHA-384
- SHA-512

Despite the varying hash lengths, they all have the same underlying algorithm. In 2008, a 224-bit version of SHA-2 was added with the publication of FIPS-3, which delved into the algorithm's details. SHA-2 is specified in RFC 4634.

By 2005, NIST announced its intention to phase out its approval for SHA-1 by 2010 due to the range of security issues that had been discovered.

Soon after, researchers published a paper that showed an attack where two separate messages could be found to result in the same SHA-1 hash in 269 operations, which was significantly lower than the previously presumed 280 operations. This showed that the security situation of SHA-1 was even more dire than had been thought, and pushed the community toward further adoption of the SHA-2 family.

SHA-2 is a family of cryptographic hash functions that includes:

- SHA-224
- SHA-256
- SHA-384
- SHA-512
- SHA-512/224
- SHA-512/256

The SHA-2 family of algorithms are used extensively in our online world, and make up a significant component of our online security. They are still considered safe in most applications, and are preferred over the insecure MD5 in the majority of use cases.

The applications of SHA-2:

The SHA-2 family of hashing algorithms are the most common hash functions in use. **SHA-256 is particularly widespread**. These hash functions are often involved in the underlying security mechanisms that help to protect our daily lives. You may have never noticed it, but SHA-2 is everywhere.

To begin with, SHA-2 is involved in many of the security protocols that help to protect much of our technology:

- Transport Layer Security (TLS) — This is one of the most widely used security protocols. You will notice it most prominently when connecting to a website that begins with **https** rather than **http**. The **s** at the end stands for **secure**, which indicates that TLS is being used to encrypt the data between your device and the server. This makes SHA-2 an important part of many of the connections you make to websites when surfing online.
- Internet Protocol Security (IPSec) — IPSec is used to secure the connection between two points, and it's most commonly seen in VPNs.
- Pretty Good Privacy (PGP) — PGP is one of the most popular protocols for encrypting emails so that they can only be read by the recipient. It protects the messages from hackers and other parties that may be able to read the data, such as your ISP.
- Secure/Multipurpose Internet Mail Extensions (S/MIME) — S/MIME is another prominent security protocol involved in email encryption.
- Secure Shell (SSH) — SSH is most commonly used for remotely accessing computers and servers, but it also has port forwarding, tunneling and file transfer applications.
- In addition to being a core component of the above-mentioned security protocols, the SHA-2 family has a range of other uses. These include:
- Authenticating data — Secure hash functions can be used to prove that data hasn't been altered, and they are involved in everything from evidence authentication to verifying that software packages are legitimate.
- Password hashing — SHA-2 hash functions are sometimes used for password hashing, but this is not a good practice. It's better to use a solution that's tailored to the purpose like bcrypt instead.
- Blockchain technologies — SHA-256 is involved in the proof-of-work function in Bitcoin and many other cryptocurrencies. It can also be involved in proof-of-stake blockchain projects.
- US Government data — Since 2015, the US National Institute of Standards and Technologies (NIST) has advised that all federal agencies should use either algorithms from the SHA-2 family or those from the SHA-3 family for most applications.

What is a hash function?

Before we can go into the specifics of what SHA-2 is, we need to cover the basics. It's not particularly useful to know that SHA-2 is a hash function with Merkle-Damgard construction if you don't know what a hash function is yet.

At their most basic level, hash functions take inputs of any size and then convert them into fixed-length outputs, which are known as hashes.

The simplest hash functions are used for tasks like data retrieval and storage. One of the main advantages of these simple hash functions is that they allow data to be found and accessed within a short and consistent time frame.

We've covered how simple hash functions work in far more detail in our What is MD5 and how is it used? article. It goes into the basics of how these types of hash functions work, and covers why they are useful in greater depth.

What are cryptographic hash functions?

Cryptographic hash functions are special types of hash functions that have a range of strange properties. Not only do they change data of any length into fixed-length values, but they are also:

- **Deterministic** – This means that the same input always leads to the same fixed length hash as its output. When you enter, "hashing is complicated" into an SHA-256 calculator, you always get a hash of d6320decc80c83e4c17915ee5de8587bb8118258759b2453fce812d47d3df56a.
- **Designed so that slight changes significantly alter the output** – If you change the initial input by even a little, you will end up with a hash that seems completely unrelated. As an example, if we put "hashing is complicated**z**" into the same SHA-256 hash calculator, we get a hash of 54afff2602d37e8dee0d696d7f6a352e8d1bae481b31cb50622b29b20594c2e5. As you can see, there doesn't seem to be any overlap between this and the prior hash, despite such a subtle difference in the inputs.
- **Fast to calculate** – When you put an input into the SHA-256 calculator, the result seems to come up instantly (unless you have poor internet connection or an older device). When you consider each of the steps that take place in the SHA-2 algorithm, you may be surprised at just how rapid the whole process is.
- **One-way functions** – If you took either of the two hashes we have just shown you and you didn't already know the initial inputs, it would be essentially impossible to figure out an input that results in either of these specific hashes. Using current techniques and technology, it's considered so impractical to figure out a suitable input from just the SHA-2 hash that much of the world's online security is built on the assumption that these attacks aren't feasible. In this sense, these functions are one-way. It's relatively easy to take an input and calculate the hash, but almost impossible to do the reverse.
- **Collision-resistant** – Cryptographic hash functions are designed so that it is unfeasible for two different inputs to result in the same hash. When separate inputs result in the same hash, it's known as a **collision**, so the required property of these functions is known as **collision resistance**. While there are differing inputs that do result in the very same hash value, the likelihood of finding these needs to be almost impossible in order for a cryptographic hash function to be considered secure. If "hashing is complicated"'s hash is 54afff2602d37e8dee0d696d7f6a352e8d1bae481b31cb50622b29b20594c2e5, it needs to be incredibly unlikely for anyone to be able to stumble across an input that results in this same hash, regardless of how much time they put into it.

What is SHA-2

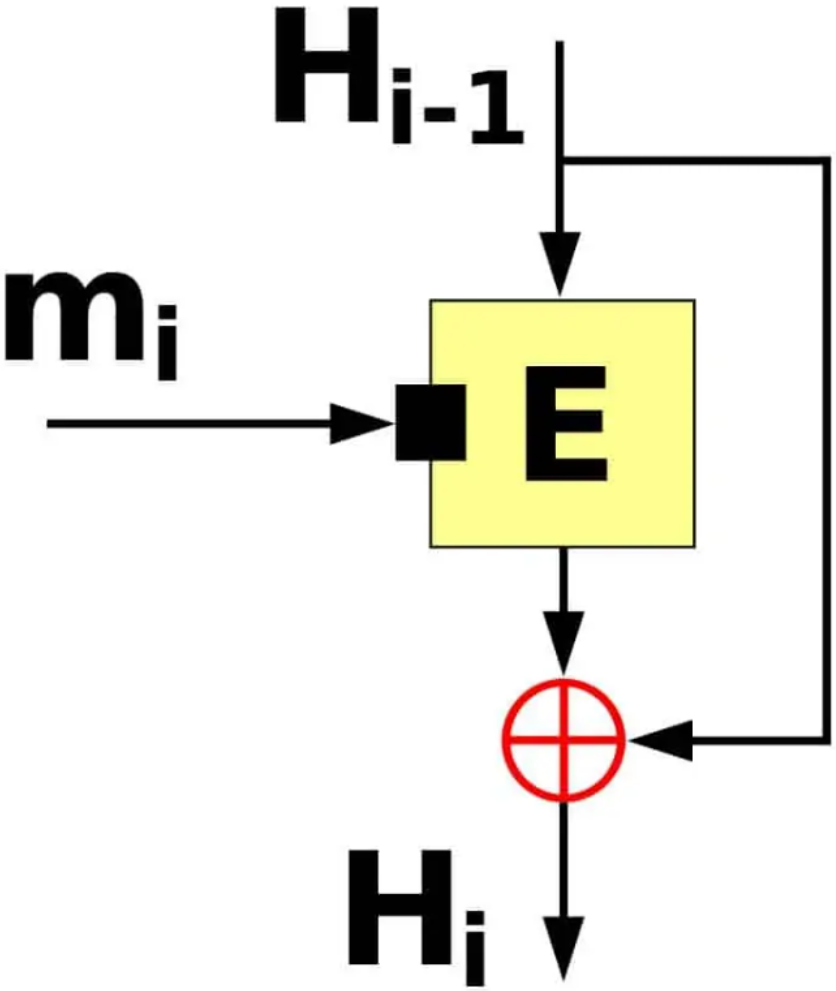
Now that you know what a hash function is, and that the SHA-2 family is a specific subtype known as cryptographic hash functions, we can get into the more specific details of SHA-2.

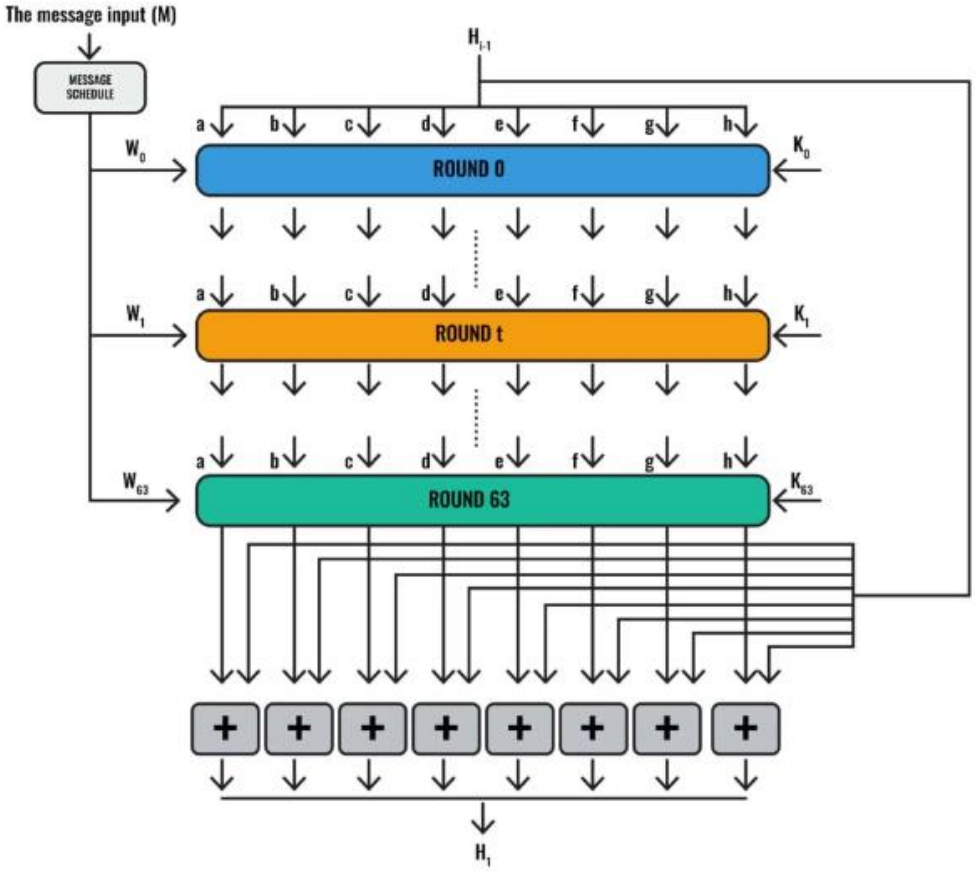
As we have mentioned, SHA-2 is not just a single hash function, but a family of six. They are collectively referred to as SHA-2 because the family are the replacements to SHA-1, which was just a single algorithm. The SHA-2 family are as follows:

- **SHA-224** – This version of SHA-2 produces a 224-bit hash. It has a block size of 512 bits, and the initial input is divided into 32-bit words for processing. The initialization variables are also 32 bits in length, as are the constants, K. Each block of data goes through 64 rounds of operations before the final hash (or the intermediate hash, in cases where multiple blocks of data are being processed) is produced.
- **SHA-256** – SHA-256 results in a 256-bit hash and has a 512-bit block size. The message input is processed in 32-bit words, while the initialization variables and constants are also 32 bits in length. SHA-256 also involves 64 rounds.

- **SHA-384** – This version produces a 384-bit hash. It differs from the prior two in that it has a 1,024-bit block size. It also varies in that it has 64-bit words, initialization variables and constants. Instead of 64 rounds, it requires 80 rounds of processing for each block of message data.
- **SHA-512** – SHA-512 results in a 512-bit hash. Apart from that, it's much like SHA-384 in that it has a 1,024-bit block size, 64-bit words, 64-bit initialization variables and 64-bit constants. However, the particular initialization variables it begins with are different from those in SHA-384. It also involves 80 rounds.
- **SHA-512/224** – This version is much like SHA-512, except that it results in a truncated hash of 224-bits. This means that it involves a process that is largely the same, except that only the left-most 224 bits are taken as the hash, while the rest is discarded. The block size is also 1024 bits, while the words, constants and initialization variables are all 64-bits long. However, the initialization variables are different from those used in SHA-512 or SHA-384. SHA-512/224 also requires 80 rounds for each block of message data.
- **SHA-512/256** – Like SHA-512/224, this iteration is also similar to SHA-512, except it produces a truncated 256-bit hash by only taking the left-most 64 bits. It has a 1,024 bit block size, as well as 64-bit words, constants and initialization variables. SHA-512/256 also has its own set of initialization variables. It involves 80 rounds.

You can view the specifics, including the values of each of the initialization variables and the constants in FIPS 180-4. Our **What is the SHA-2 algorithm?** article goes through every step of the process for SHA-256. If you are curious about one of the other versions of SHA-2, you can simply follow the steps for SHA-256 in our article, while changing the values to those specified in the FIPS 180-4 document.





SH_2 CODE C:

https://github.com/B-Con/crypto-algorithms/blob/master/sha256.c