Overview of SHA-256

Justin Wyss-Gallifent

April 14, 2022

0.1 Introduction

This is a very high-level outline of how the SHA-256 hash works.

0.2 History

The SHA-2 family of algorithms was developed by the NSA and given to NIST. They were first published in 2001. They include SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256. The values indicate the number of bits in the output. The last two indicate slight modifications, for example SHA-512/225 is essentially SHA-512 with a little bit of tweaking to convert its output to 224 bits.

0.3 High Level

SHA-256 maps an arbitrary-length binary string to a 256-bit string. It does so by starting with a default set of eight 32-bit hash values (256 bits) and then "mixing in" 512-bit blocks of the message, block by block, until the entire message has been mixed in. It then returns the resulting eight 32-bit hash values as a single 256-bit string.

0.4 Making Soup

Think about SHA-256 as making soup in the following sense:

- Start with some soup stock.
- Have some spices on hand.
- Chop up your ingredients.
- One-by-one, add each ingredient with some spices and stir; each ingredient updates the stock.
- When you're done, you have soup.

0.5 Process

1. Take the original message M and append a 1.
2. Append enough 0s to make the message 64 bits short of the next highest multiple of 512 bits.
3. Append the 64-bit representation of the length of the original message. Note that this limits the original message to a length of 2^{64} bits but in practice that is okay.
4. Call the result M as well for simplicity. This M is then a multiple of 512 bits long.
5. Sort the ingredients: Break M into 512-bit blocks into a variable N blocks of length 512:

$$M = M^{(1)} | M^{(2)} | ... | M^{(N)}$$

6. Chop each ingredient: Each $M^{(i)}$ is then again broken into sixteen 32-bit blocks:

$$M^{(i)} = M_0^{(i)} | M_1^{(i)} | ... | M_{15}^{(i)}$$

7. The soup stock: Initalize eight (32-bit) hash values given in hexadecimal as follows.

$$\begin{aligned} H_1^{(0)} &= \texttt{6A09E667} \\ H_2^{(0)} &= \texttt{BB67AE85} \\ H_3^{(0)} &= \texttt{3C6EF372} \\ H_4^{(0)} &= \texttt{A54FF53A} \\ H_5^{(0)} &= \texttt{510E527F} \\ H_6^{(0)} &= \texttt{9B05688C} \\ H_7^{(0)} &= \texttt{1F83D9AB} \\ H_8^{(0)} &= \texttt{5BE0CD19} \end{aligned}$$

   Notice that together there are 256 bits. These do not change, they are fixed in the algorithm. The first is:

$$\left\lfloor 2^{32}\left(\sqrt{2}-\lfloor\sqrt{2}\rfloor\right)\right\rfloor_{16} = \lfloor 1779033703 \rfloor_{16} = 6A09E667$$

   The rest are obtained by replacing $\sqrt{2}$ by $\sqrt{3}$, $\sqrt{5}$, $\sqrt{7}$, $\sqrt{11}$, $\sqrt{13}$, $\sqrt{17}$, and $\sqrt{19}$ and subtracting the integer part of each (all together, the first eight primes)

8. Prepare the spices: Initialize sixty-four 32-bit key words given in hexadecimal as:

$$\begin{aligned} K_0 &= \texttt{428A2F98} \\ K_1 &= \texttt{71374491} \\ \vdots \quad & \quad\ \vdots \\ K_{63} &= \texttt{C67178F2} \end{aligned}$$

   These do not change, they are fixed in the algorithm. They are calculated like the $H^{(0)}$ values except using cube roots and using the first sixty-four primes.

9. Ready for the first ingredient: Set $i = 1$.

10. The soup stock is in the pot: We initialize eight registers a,...,h with $H_1^{(i-1)}$, ..., $H_8^{(i-1)}$ respectively. These will initially be the hash values above but they will change since we will come back here.

11. Add the ingredients and spices and stir to get an updated stock: Initialize words W_0,...,W_{15} with $M_0^{(i)}$,...,$M_1^{(i)}$. We will then follow a process involving and modifying the words W, the key words K, and the registers a,...,h. This process is complicated and involves multiple rounds of shifts and XORs. The practical result it that $H_1^{(i)}$, ..., $H_8^{(i)}$ are set. These are also 32-bit hash values.

12. If there are more ingredients, go back: If $i < N$ then increase i and go back to step 10 and do it again.

13. If not then the soup is done: If $i = N$ then we've taken care of the whole message and we join the $H_1^{(i)}$, ..., $H_8^{(i)}$ together to form the message digest:

$$h(M) = H_1^{(i)} ... H_8^{(i)}$$