Algorithm for SHA-256

The SHA-256 algorithm is a cryptographic hash function that produces a fixed 256-bit (32-byte) hash value from an input message. Below is the step-by-step algorithm:

---

1. Preprocessing

1. **Input Message**: Take the input message (M) as a binary string.
2. **Padding**:
   - Append a single 1 bit to the end of the message.
   - Append k 0 bits such that the total length (in bits) is congruent to 448 modulo 512.
   - Append the 64-bit representation of the original message length to the end of the message.

Result: A padded message whose length is a multiple of 512 bits.

---

2. Initialize Hash Values

Set the initial hash values H_0, H_1..., H_7 to the following constants (in hexadecimal):

H_0 = 6a09e667

H_1 = bb67ae85

H_2 = 3c6ef372

H_3 = a54ff53a

H_4 = 510e527f

H_5 = 9b05688c

H_6 = 1f83d9ab

H_7 = 5be0cd19

---

3. Prepare Message Schedule

1. Divide the padded message into blocks of 512 bits each.
2. For each 512-bit block:
   - Break it into 16 words (W_0, W_1 ..., W_{15}) of 32 bits each.

- Extend the 16 words into 64 words (W_0, W_1 ..., W_{63}) using the formula:

  $W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}$

  Where:

  $\sigma_0(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3)$

  $\sigma_1(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10)$

---

4. Compression Function

1. Initialize working variables:

   $a = H_0$, $b = H_1$, $c = H_2$, $d = H_3$, $e = H_4$, $f = H_5$, $g = H_6$, $h = H_7$

2. For $t = 0$ to 63:

   - Calculate:

     $T_1 = h + \Sigma_1(e) + \mathit{Ch}(e,f,g) + K_t + W_t$

     $T_2 = \Sigma_0(a) + \mathit{Maj}(a,b,c)$

     Where:

     $\Sigma_0(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22)$

     $\Sigma_1(x) = (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25)$

     $\mathit{Ch}(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z)$

     $\mathit{Maj}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$

     Update the working variables:

     $h = g$, $g = f$, $f = e$, $e = d + T_1$, $d = c$, $c = b$, $b = a$, $a = T_1 + T_2$

---

5. Update Hash Values

After processing each block, update the hash values as:

$H_0 = H_0 + a$, $H_1 = H_1 + b$, $H_2 = H_2 + c$, $H_3 = H_3 + d$, $H_4 = H_4 + e$, $H_5 = H_5 + f$, $H_6 = H_6 + g$, $H_7 = H_7 + h$

---

6. Produce Final Hash

Concatenate the final hash values H_0, H_1..., H_7 to produce the final 256-bit hash.

---

7. Symbols and Notation

- $\ggg$: Rotating right (circular right shift).
  - Bits shifted out of the right side are reinserted on the left.
  - Example: $1011 \ggg 2 = 1110$
- $\gg$: Logical (arithmetic) right shift.
  - Bits shifted out of the right side are discarded, and zeros fill on the left.
  - Example: $1011 \gg 2 = 0010$
- $\oplus$: XOR (exclusive OR).
  - Bitwise operation where $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, and so on.
