SHA Hashing Notes

Mimanshu Maheshwari

Thursday 28^{th} March, 2024, 04:06

Contents

List of Tables

Chapter 1

SHA 256

1.1 Introduction

SHA256 is a 256 bits hash. Ment to provide 128 bits of security against collision attack.

1.2 Implementation

SHA256 operates in a manner of MD4, MD5 and SHA-1. The message to be hashed is

1. Padded with its length in such a way that the result is multiple of 512 bits long.
2. Parsed into 512 bits message blocks $M^1, M^1, \ldots, M^1$,
3. Message blocks are processed one block at a time: Beginning with a fixed initial hash value $H^{(0)}$, sequentially compute
$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}(H^{(i-1)})$$
where C is the SHA-256 *compression function* and $+$ means word-wise $\mod 2^{32}$ addition. $H^{(N)}$ is the ***hash*** of M.

SHA-256 operates on 512-bits *message block* and a 256-bits *intermidiate hash value*. It essentially is a 256-bit cypher algorithm which encripts intermidiate hash value using the message block as key. Hence, their are two main components:

- Compression Function
- message schedule

Notation	Meaning
$\oplus$	Bitwise XOR
$\vee$	Bitwise AND
$\wedge$	Bitwise OR
$\neg$	Bitwise Complement
$+$	$\mod 2^{32}$ addition
R^n	right shift by n bits
S^n	right rotate by n bits

Table 1.1: Notation Reference

All of the operators in 1.1 table act on 32-bit words.

The initial value of $H^{(0)}$ is the following sequence of 32 bit words (which are obtained by taking the fractional parts of the square roots of the first eight primes.)

$$H_1^{(0)} = 6a09e667 \tag{1.1}$$
$$H_2^{(0)} = bb67ae85 \tag{1.2}$$
$$H_3^{(0)} = 3c6ef372 \tag{1.3}$$
$$H_4^{(0)} = a54ff53a \tag{1.4}$$
$$H_5^{(0)} = 510e527f \tag{1.5}$$
$$H_6^{(0)} = 9b05688c \tag{1.6}$$
$$H_7^{(0)} = 1f83d9ab \tag{1.7}$$
$$H_8^{(0)} = 5be0cd19 \tag{1.8}$$

1.3 Preprocessing

Computing the hash of message begins by padding the message:

1. Pad the message in usual way: Suppose the lenght of message M, in bits, is l. Append the bit "1" to the end of message, and the the k zero bits, where k is the smallest non-negative solution to the equation $l + 1 + 1 \equiv 448 \mod 512$. To this append the 64-bit block which is equal to the number l written in binary. For example, the (8-bit ASCII) message "abc" has length $8 \cdot 3 = 24$ so it is padded with a one, then $448 - (24 + 1) = 423$ zero bits, and thenthe length to become the 512-bit padded message:

$$01100001 \quad 01100010 \quad 01100011 \quad \underbrace{0000\ldots0}_{423-bits} \quad \overbrace{00\ldots011000}^{64-bits}$$

   The length of the padded message should now be 512 bits.

2. Parse the message into N 512-bits block $M^{(1)}, M^{(1)}, \ldots, M^{(1)}$ The first 32 bits of message block i are denoted $M_0^{(i)}$, the next 32 bits are $M_1^{(i)}$, and so on up to $M_{15}^{(i)}$. We use big-endian convention througout, so within each 32-bit word, the left most bit is stored in the most significant bit position.

1.4 Main loop

The hash computation proceeds as follows:
$for\ i = 1 \to N$ (N = Number of blocks in the padded message)

- Initialize registers a, b, c, d, e, f, g, h with the $(i-1)^{st}$ intermidiate hash value (= initial hash value when $i = 1$)

$$\begin{gathered} a \leftarrow H_1^{(i-1)} \\ b \leftarrow H_2^{(i-1)} \\ c \leftarrow H_3^{(i-1)} \\ d \leftarrow H_4^{(i-1)} \\ \vdots \\ h \leftarrow H_8^{(i-1)} \end{gathered}$$

- Apply the SHA-256 *compression function* to update registers $a, b, c, \ldots, h$
  $for\ j = 0 \rightarrow to63$
  Compute $Ch(e,f,g), Maj(a,b,c), \sum_0(a), \sum_1(e), andW_j$

$$\begin{aligned}
T_1 &\leftarrow h + \sum_1(e) + Ch(e,f,g) + K_j + W_j \\
T_2 &\leftarrow \sum_0(a) + Maj(a,b,c) \\
h &\leftarrow g \\
g &\leftarrow f \\
f &\leftarrow e \\
e &\leftarrow d + T_1 \\
d &\leftarrow c \\
c &\leftarrow b \\
b &\leftarrow a \\
a &\leftarrow T_1 + T_2
\end{aligned}$$

- Compute the i^{th} intermidiate hash value $H^{(i)}$

$$\begin{aligned}
H_1^{(i)} &\leftarrow H_1^{(i-1)} \\
H_2^{(i)} &\leftarrow H_2^{(i-1)} \\
H_3^{(i)} &\leftarrow H_3^{(i-1)} \\
H_4^{(i)} &\leftarrow H_4^{(i-1)} \\
&\vdots \\
H_8^{(i)} &\leftarrow H_8^{(i-1)}
\end{aligned}$$

$H^{(N)} = \left(H_1^{(N)}, H_2^{(N)}, H_3^{(N)}, \ldots, H_8^{(N)}\right)$ is the hash of M.

1.5 Definations

Six logical functions are used in SHA-256. Each function operates on 32-bits words and produces a 32-bit word as output.

$$Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) \tag{1.9}$$
$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \tag{1.10}$$
$$\sum_0(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x) \tag{1.11}$$
$$\sum_1(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x) \tag{1.12}$$
$$\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x) \tag{1.13}$$
$$\sigma_1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x) \tag{1.14}$$

1.5.1 Expanded Message Blocks

$W_0, W_1, \ldots, W_{63}$ computed as follows via the **SHA-256 message schedule**:
$W_j = M_j^{(i)}$ for $j = 0, 1, 2, \ldots, 15$, and

$for\ j = 16 \to 63$

$$W_j \leftarrow \sigma_1(W_{(j-2)}) + W_{(j-7)} + \sigma_0(W_{(j-15)}) + W_{(j-16)}$$

A sequence of contant words $K_0, K_1, K_2, \ldots, K_{63}$ is used in SHA-256. in Hex, these are given by:

0x428a2f98	0x71374491	0xb5c0fbcf	0xe9b5dba5
0x3956c25b	0x59f111f1	0x923f82a4	0xab1c5ed5
0xd807aa98	0x12835b01	0x243185be	0x550c7dc3
0x72be5d74	0x80deb1fe	0x9bdc06a7	0xc19bf174
0xe49b69c1	0xefbe4786	0x0fc19dc6	0x240ca1cc
0x2de92c6f	0x4a7484aa	0x5cb0a9dc	0x76f988da
0x983e5152	0xa831c66d	0xb00327c8	0xbf597fc7
0xc6e00bf3	0xd5a79147	0x06ca6351	0x14292967
0x27b70a85	0x2e1b2138	0x4d2c6dfc	0x53380d13
0x650a7354	0x766a0abb	0x81c2c92e	0x92722c85
0xa2bfe8a1	0xa81a664b	0xc24b8b70	0xc76c51a3
0xd192e819	0xd6990624	0xf40e3585	0x106aa070
0x19a4c116	0x1e376c08	0x2748774c	0x34b0bcb5
0x391c0cb3	0x4ed8aa4a	0x5b9cca4f	0x682e6ff3
0x748f82ee	0x78a5636f	0x84c87814	0x8cc70208
0x90befffa	0xa4506ceb	0xbef9a3f7	0xc67178f2

Table 1.2: First 32 bits of the fractional part of the cube roots of the first 64 primes

use 1.2 table to set initial value of buffer.