SHA256

Saravanan Vijayakumaran
sarva@ee.iitb.ac.in

Department of Electrical Engineering
Indian Institute of Technology Bombay

September 3, 2019

SHA-256

- SHA = Secure Hash Algorithm, 256-bit output length
- Accepts bit strings of length upto $2^{64} - 1$
- Output calculation has two stages
  - Preprocessing
  - Hash Computation
- Preprocessing
  1. A 256-bit state variable $H^{(0)}$ is set to

$$
\begin{aligned}
H_0^{(0)} &= \texttt{0x6A09E667}, & H_1^{(0)} &= \texttt{0xBB67AE85}, \\
H_2^{(0)} &= \texttt{0x3C6EF372}, & H_3^{(0)} &= \texttt{0xA54FF53A}, \\
H_4^{(0)} &= \texttt{0x510E527F}, & H_5^{(0)} &= \texttt{0x9B05688C}, \\
H_6^{(0)} &= \texttt{0x1F83D9AB}, & H_7^{(0)} &= \texttt{0x5BE0CD19}.
\end{aligned}
$$

  2. The input M is padded to a length which is a multiple of 512

SHA-256 Input Padding

- Let input M be l bits long
  - Find smallest non-negative k such that

$$k + l + 65 = 0 \bmod 512$$

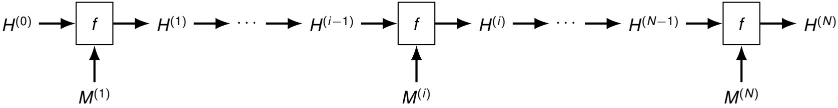
  - Append $k + 1$ bits consisting of single 1 and k zeros
  - Append 64-bit representation of l
- Example: $M = 101010$ with $l = 6$
  - $k = 441$
  - 64-bit representation of 6 is $000 \cdots 00110$
  - 512-bit padded message

$$\underbrace{101010}_{M}\ 1\ \underbrace{00000 \cdots 00000}_{441 \text{ zeros}}\ \underbrace{00 \cdots 00110}_{l}.$$

SHA-256 Hash Computation

1. Padded input is split into N 512-bit blocks $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$
2. Given $H^{(i-1)}$, the next $H^{(i)}$ is calculated using a function f

$$H^{(i)} = f(M^{(i)}, H^{(i-1)}), \quad 1 \leq i \leq N.$$



3. f is called a *compression function*
4. $H^{(N)}$ is the output of SHA-256 for input M

SHA-256 Compression Function Building Blocks

- U, V, W are 32-bit words
- $U \wedge V$, $U \vee V$, $U \oplus V$ denote bitwise AND, OR, XOR
- $U + V$ denotes integer sum modulo 2^{32}
- $\neg U$ denotes bitwise complement
- For $1 \leq n \leq 32$, the shift right and rotate right operations

$$\mathrm{SHR}^n(U) = \underbrace{000\cdots000}_{n\text{ zeros}} u_0 u_1 \cdots u_{30-n} u_{31-n},$$

$$\mathrm{ROTR}^n(U) = u_{31-n+1} u_{31-n+2} \cdots u_{30} u_{31} u_0 u_1 \cdots u_{30-n} u_{31-n},$$

- Bitwise choice and majority functions

$$\mathrm{Ch}(U, V, W) = (U \wedge V) \oplus (\neg U \wedge W),$$

$$\mathrm{Maj}(U, V, W) = (U \wedge V) \oplus (U \wedge W) \oplus (V \wedge W),$$

- Let

$$\Sigma_0(U) = \mathrm{ROTR}^2(U) \oplus \mathrm{ROTR}^{13}(U) \oplus \mathrm{ROTR}^{22}(U)$$

$$\Sigma_1(U) = \mathrm{ROTR}^6(U) \oplus \mathrm{ROTR}^{11}(U) \oplus \mathrm{ROTR}^{25}(U)$$

$$\sigma_0(U) = \mathrm{ROTR}^7(U) \oplus \mathrm{ROTR}^{18}(U) \oplus \mathrm{SHR}^3(U)$$

$$\sigma_1(U) = \mathrm{ROTR}^{17}(U) \oplus \mathrm{ROTR}^{19}(U) \oplus \mathrm{SHR}^{10}(U)$$

SHA-256 Compression Function Calculation

- Maintains internal state of 64 32-bit words $\{W_j \mid j = 0, 1, \ldots, 63\}$
- Also uses 64 constant 32-bit words $K_0, K_1, \ldots, K_{63}$ derived from the first 64 prime numbers $2, 3, 5, \ldots, 307, 311$
- $f(M^{(i)}, H^{(i-1)})$ proceeds as follows
  1. Internal state initialization
  $$W_j = \begin{cases} M_j^{(i)} & 0 \le j \le 15, \\ \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16} & 16 \le j \le 63. \end{cases}$$
  2. Initialize eight 32-bit words
  $$(A, B, C, D, E, F, G, H) = \left(H_0^{(i-1)}, H_1^{(i-1)}, \ldots, H_6^{(i-1)}, H_7^{(i-1)}\right).$$
  3. For $j = 0, 1, \ldots, 63$, iteratively update $A, B, \ldots, H$
  $$\begin{aligned} T_1 &= H + \Sigma_1(E) + \mathrm{Ch}(E, F, G) + K_j + W_j \\ T_2 &= \Sigma_0(A) + \mathrm{Maj}(A, B, C) \\ (A, B, C, D, E, F, G, H) &= (T_1 + T_2, A, B, C, D + T_1, E, F, G) \end{aligned}$$
  4. Calculate $H^{(i)}$ from $H^{(i-1)}$
  $$(H_0^{(i)}, H_1^{(i)}, \ldots, H_7^{(i)}) = \left(A + H_0^{(i-1)}, B + H_1^{(i-1)}, \ldots, H + H_7^{(i-1)}\right).$$

The Merkle-Damgård Transform

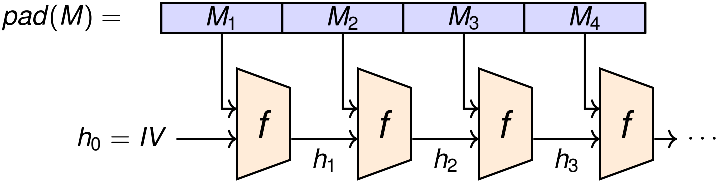


Figure source: `https://www.iacr.org/authors/tikz/`

- The SHA-256 construction is an example of the MD transform
- Typical hash function design
  - Construct collision-resistant compression function
  - Extend the domain using MDT to get collision-resistant hash function

References

- Chapter 3 of *An Introduction to Bitcoin*, S. Vijayakumaran, `www.ee.iitb.ac.in/~sarva/bitcoin.html`