Muhammad Izham Bin Norhamadi
B032020039 3BITZ

Tutorial 8: SHA-256

1. Write a string of your full name and high school alma mater. Full string must be between 20 and 40 characters only.

1.1 Convert this message into bytes and write them in hexadecimal.

M =
4D756861 **6D6D6164** 20497A68 **616D2053** 4D4B2054 **756E2050** 6572616B
M = 'Muhammad Izham SMK Tun Perak'.

1.2 There are L = 28·8 = 224 bits. Add another bit '1' which will give us 8 in hexa. Then, add $E0_{16} = 224_{10}$ at the end of 512 bits = 64 bytes array.

M =
4D756861**6D6D6164**20497A68**616D2053**4D4B2054**756E2050**6572616B80000000
00000000000000000000000000000000000000000000000000000000000000E0

1.3 Convert them into 16 unsigned long of 32 bits.

$M_0^{(0)}$	4D756861
$M_1^{(0)}$	6D6D6164
$M_2^{(0)}$	20497A68
$M_3^{(0)}$	616D2053
$M_4^{(0)}$	4D4B2054
$M_5^{(0)}$	756E2050
$M_6^{(0)}$	6572616B
$M_7^{(0)}$	80000000
$M_8^{(0)}$	00000000
$M_9^{(0)}$	00000000
$M_{10}^{(0)}$	00000000
$M_{11}^{(0)}$	00000000
$M_{12}^{(0)}$	00000000
$M_{13}^{(0)}$	00000000
$M_{14}^{(0)}$	00000000
$M_{15}^{(0)}$	000000E0

Muhammad Izham Bin Norhamadi
B032020039 3BITZ

2. In this tutorial we will do only one round for j = 0, 1, 2, …, 31.

for j = 0, 1, …, 15

$W_j = M_j^{(i)}$

W_0	=	$M_0^{(0)}$	=	4D756861
W_1	=	$M_1^{(0)}$	=	6D6D6164
W_2	=	$M_2^{(0)}$	=	20497A68
W_3	=	$M_3^{(0)}$	=	616D2053
W_4	=	$M_4^{(0)}$	=	4D4B2054
W_5	=	$M_5^{(0)}$	=	756E2050
W_6	=	$M_6^{(0)}$	=	6572616B
W_7	=	$M_7^{(0)}$	=	80000000
W_8	=	$M_8^{(0)}$	=	00000000
W_9	=	$M_9^{(0)}$	=	00000000
W_{10}	=	$M_{10}^{(0)}$	=	00000000
W_{11}	=	$M_{11}^{(0)}$	=	00000000
W_{12}	=	$M_{12}^{(0)}$	=	00000000
W_{13}	=	$M_{13}^{(0)}$	=	00000000
W_{14}	=	$M_{14}^{(0)}$	=	00000000
W_{15}	=	$M_{15}^{(0)}$	=	000000E0

for j = 16 to 63

{

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

}

Muhammad Izham Bin Norhamadi
B032020039 3BITZ

3. Initialize the value of the 32-bit registers a, b, c, d, e, f, g, h for i =1st 512-bit block.

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

a	=	$H_1^{(i-1)}$	=	6A09E667
b	=	$H_2^{(i-1)}$	=	BB67AE85
c	=	$H_3^{(i-1)}$	=	3C6EF372
d	=	$H_4^{(i-1)}$	=	A54FF53A
e	=	$H_5^{(i-1)}$	=	510E527F
f	=	$H_6^{(i-1)}$	=	9B05688C
g	=	$H_7^{(i-1)}$	=	1F83D9AB
h	=	$H_8^{(i-1)}$	=	5BE0CD19

4. Apply the SHA-256 compression function to update registers $a, b, c, d, e, f, g, h,$
In this tutorial we will do only **one round** for $j = 0$. Take round keys as cube roots of the first 64 primes.

for j = 0 to 63 do
{
\\Compute $\lambda(e, f, g)$, $\mu(a, b, c)$, $\Sigma_0(a)$, $\Sigma_1(e)$, and W_j
\\see Definitions of the 5 logical functions above.
$T_1 = h + \Sigma_1(e) + \lambda(e, f, g) + K_j + W_j$
$T_2 = \Sigma_0(a) + \mu(a, b, c)$

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

Muhammad Izham Bin Norhamadi
B032020039 3BITZ

Consider the following notation of the 7 operations in Secure Hashing Algorithm SHA-2:

Symbol	Operation
$\oplus$	Bitwise XOR
$\wedge$	Bitwise AND
$\vee$	Bitwise OR
$\neg$	Bitwise complement
$+$	Mod 2^{32} addition
R^n	Right shift by n bits >>n
S^n	Right rotation by n bits

Table 1: Notations on 32-bit operations

All of these operators act on 32-bit words. Unsigned long 32-bit register is popular programming variable in the year 2000 onwards.

Let us compute $T_1 = h + \Sigma_1(e) + \lambda(e, f, g) + K_j + W_j$,
First, we need to compute $\Sigma_1(e)$
where $\Sigma_1(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x)$

$\Sigma_1(e) = \Sigma_1(510E527F) = ?$

$S^6(e) = S^6(510E527F) = FD443949$

01010001000011100101001001111111
01010001000011100101001001111111
11111101010001000011100101001001
1111 1101 0100 0100 0011 1001 0100 1001

$S^{11}(e) = S^{11}(510E527F) = 4FEA21CA$

01010001000011100101001001111111
01001111111010100010000111001010
0100 1111 1110 1010 0010 0001 1100 1010

$S^{25}(e) = S^{25}(510E527F) = 87293FA8$

01010001000011100101001001111111
10000111001010010011111110101000
1000 0111 0010 1001 0011 1111 1010 1000

$\Sigma_1(510E527F) = S^6(510E527F) \oplus S^{11}(510E527F) \oplus S^{25}(510E527F)$
$= FD443949 \oplus 4FEA21CA \oplus 87293FA8$
$= 3587272B$

Second, we need to compute $\lambda(e, f, g)$
where

$\lambda(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$
$\lambda(e, f, g) = \lambda(510E527F, 9B05688C, 1F83D9AB)$

Muhammad Izham Bin Norhamadi
B032020039 3BITZ

= (510E527F^9B05688C)⊕(¬510E527F^1F83D9AB)
= 1104400C ⊕ 0E818980
= 1F85C98C

Note: ¬510E527F = AEF1AD80.

Coming back to compute

$T_1 = h + \Sigma_1(e) + \lambda(e, f, g) + K_0 + W_0$,
= 5BE0CD19 + 3587272B + 1F85C98C + 428A2F98 + 4D756861,
= 140ED55C9
= 40ED55C9 $\pmod{2^{32}}$

Note: K_0 is a preset from 32-bit fractional on cube root of prime 2.

Third, we need to compute $T_2 = \Sigma_0(a) + \mu(a, b, c)$

$\Sigma_0(a) = S^2(a) \oplus S^{13}(a) \oplus S^{22}(a)$ where a = 6A09E667
Σ_0(6A09E667) = S^2(6A09E667) ⊕ S^{13}(6A09E667) ⊕ S^{22}(6A09E667)

S^2(6A09E667) = S^2(01101010000010011110011001100111)
= 11011010100000100111100110011001
= 11011010100000100111100110011001
= DA827999

S^{13}(6A09E667) = S^{13}(01101010000010011110011001100111)
= S^{13}(01101010000010011110011001100111)
= 00110011001110110101000001001111
= 00110011001110110101000001001111
= 333B504F

S^{22}(6A09E667) = S^{22}(01101010000010011110011001100111)
= S^{22}(01101010000010011110011001100111)
= 00100111100110011001110110101000
= 00100111100110011001110110101000
= 27999DA8

Σ_0(6A09E667) = S^2(6A09E667) ⊕ S^{13}(6A09E667) ⊕ S^{22}(6A09E667)
= DA827999 ⊕ 333B504F ⊕ 27999DA8
= CE20B47E

$\mu(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$

$\mu(a, b, c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$
= (6A09E667 ^ BB67AE85) ⊕ (6A09E667 ^ 3C6EF372) ⊕ (BB67AE85 ^ 3C6EF372)
= 2A01A605 ⊕ 2808E262 ⊕ 3866A200
= 3A6FE667

Muhammad Izham Bin Norhamadi
B032020039 3BITZ

Fourth, we can compute $T_2 = \Sigma_0(a) + \mu(a, b, c)$
= CE20B47E + 3A6FE667
= 108909AE5
= 08909AE5 (mod 2^{32})

Lastly, from the previous values of registers *a, b, c, d, e, f, g, h*,

a	=	$H_1^{(i-1)}$	=	6A09E667
b	=	$H_2^{(i-1)}$	=	BB67AE85
c	=	$H_3^{(i-1)}$	=	3C6EF372
d	=	$H_4^{(i-1)}$	=	A54FF53A
e	=	$H_5^{(i-1)}$	=	510E527F
f	=	$H_6^{(i-1)}$	=	9B05688C
g	=	$H_7^{(i-1)}$	=	1F83D9AB
h	=	$H_8^{(i-1)}$	=	5BE0CD19

Update on the register to get ready to go to next round,

$h = g =$ 1F83D9AB
$g = f =$ 9B05688C
$f = e =$ 510E527F
$e = d + T_1 =$ A54FF53A + 40ED55C9 = E63D4B03
$d = c =$ 3C6EF372
$c = b =$ BB67AE85
$b = a =$ 6A09E667
$a = T_1 + T_2 =$ 40ED55C9 + 08909AE5 = 497DF0AE

Fill up the new values of the 32-bit registers after the first loop $j = 0$. At the end of the computation, Block 1 has been processed. The values of $\{H_i\}$ are

a	=	497DF0AE
b	=	6A09E667
c	=	BB67AE85
d	=	3C6EF372
e	=	E63D4B03
f	=	510E527F
g	=	9B05688C
h	=	1F83D9AB

Register: a	b	c	d	e	f	g	h
497DF0AE	6A09E667	BB67AE85	3C6EF372	E63D4B03	510E527F	9B05688C	1F83D9AB

\\ After the full 64 loops, we may compute the i^{th} intermediate hash value $H^{(i)}$

Muhammad Izham Bin Norhamadi
B032020039 3BITZ

$H_1^{(i)} = a + H_1^{(i-1)}$
$H_2^{(i)} = b + H_2^{(i-1)}$
$H_3^{(i)} = c + H_3^{(i-1)}$
$H_4^{(i)} = d + H_4^{(i-1)}$
$H_5^{(i)} = e + H_5^{(i-1)}$
$H_6^{(i)} = f + H_6^{(i-1)}$
$H_7^{(i)} = g + H_7^{(i-1)}$
$H_8^{(i)} = h + H_8^{(i-1)}$

Finally,

H_1 = 497DF0AE+6A09E667 = B387D715
H_2 = 6A09E667+BB67AE85 = 1257194EC
H_3 = BB67AE85+3C6EF372 = F7D6A1F7
H_4 = 3C6EF372+A54FF53A = E1BEE8AC
H_5 = E63D4B03+510E527F = 1374B9D82
H_6 = 510E527F+9B05688C = EC13BB0B
H_7 = 9B05688C+1F83D9AB = BA894237
H_8 = 1F83D9AB+5BE0CD19 = 7B64A6C4.

The message digest is, at the moment, after an initial round 0, ready for the next round j=1.

B387D715 257194EC F7D6A1F7 E1BEE8AC 374B9D82 EC13BB0B BA894237
7B64A6C4

=
B387D715257194ECF7D6A1F7E1BEE8AC374B9D82EC13BB0BBA8942377B64A6C4