CRYPTOGRAPHY #04.1

SHA – 2 supplement

Jacek Tchórzewski, jacek.tchorzewski@pk.edu.pl

1. SHA - 512

SHA – 512 is producing 512 bits of hash. The algorithm is operating on 8B Integers. Pseudocode is presented below.

Step 1: H = (0x6a09e667f3bcc908, 0xbb67ae8584caa73b, 0x3c6ef372fe94f82b, 0xa54ff53a5f1d36f1, 0x510e527fade682d1, 0x9b05688c2b3e6c1f, 0x1f83d9abfb41bd6b, 0x5be0cd19137e2179)

Step 2: K = (0x428a2f98d728ae22, 0x7137449123ef65cd, 0xb5c0fbcfec4d3b2f, 0xe9b5dba58189dbbc, 0x3956c25bf348b538, 0x59f111f1b605d019, 0x923f82a4af194f9b, 0xab1c5ed5da6d8118, 0xd807aa98a3030242, 0x12835b0145706fbe, 0x243185be4ee4b28c, 0x550c7dc3d5ffb4e2, 0x72be5d74f27b896f, 0x80deb1fe3b1696b1, 0x9bdc06a725c71235, 0xc19bf174cf692694, 0xe49b69c19ef14ad2, 0xefbe4786384f25e3, 0x0fc19dc68b8cd5b5, 0x240ca1cc77ac9c65, 0x2de92c6f592b0275, 0x4a7484aa6ea6e483, 0x5cb0a9dcbd41fbd4, 0x76f988da831153b5, 0x983e5152ee66dfab, 0xa831c66d2db43210, 0xb00327c898fb213f, 0xbf597fc7beef0ee4, 0xc6e00bf33da88fc2, 0xd5a79147930aa725, 0x06ca6351e003826f, 0x142929670a0e6e70, 0x27b70a8546d22ffc, 0x2e1b21385c26c926, 0x4d2c6dfc5ac42aed, 0x53380d139d95b3df, 0x650a73548baf63de, 0x766a0abb3c77b2a8, 0x81c2c92e47edaee6, 0x92722c851482353b, 0xa2bfe8a14cf10364, 0xa81a664bbc423001, 0xc24b8b70d0f89791, 0xc76c51a30654be30, 0xd192e819d6ef5218, 0xd69906245565a910, 0xf40e35855771202a, 0x106aa07032bbd1b8, 0x19a4c116b8d2d0c8, 0x1e376c085141ab53, 0x2748774cdf8eeb99, 0x34b0bcb5e19b48a8, 0x391c0cb3c5c95a63, 0x4ed8aa4ae3418acb, 0x5b9cca4f7763e373, 0x682e6ff3d6b2b8a3, 0x748f82ee5defb2fc, 0x78a5636f43172f60, 0x84c87814a1f0ab72, 0x8cc702081a6439ec, 0x90befffa23631e28, 0xa4506cebde82bde9, 0xbef9a3f7b2c67915, 0xc67178f2e372532b, 0xca273eceea26619c, 0xd186b8c721c0c207, 0xeada7dd6cde0eb1e, 0xf57d4f7fee6ed178, 0x06f067aa72176fba, 0x0a637dc5a2c898a6, 0x113f9804bef90dae, 0x1b710b35131c471b, 0x28db77f523047d84, 0x32caab7b40c72493, 0x3c9ebe0a15c9bebc, 0x431d67c49c100d4c, 0x4cc5d4becb3e42b6, 0x597f299cfc657e2a, 0x5fcb6fab3ad6faec, 0x6c44198c4a475817)

Step 3: M = pad(M, 128)
Step 4: i = 0
Step 5: tmp = (0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00)

Step 6: while i < len(M) do Steps 7-23
Step 7: W = () , **note** W can be declared as 80B array
Step 8: for j = 0,...,15 do Step 9
Step 9: $W.append(parseUint64(M(\left\lfloor\frac{i}{128}\right\rfloor * 128 + j * 8,\ \left\lfloor\frac{i}{128}\right\rfloor * 128 + j * 8 + 7)))$.
Step 10: for j = 16,...,79 do Step 11
Step 11: W.append(W(j−16)+W(j−7)+s(W(j−15),1,8,7)+s(W(j−2),19,61,6))
Step 12: for j = 0,...,7 do Step 13
Step 13: tmp(j) = H(j)
Step 14: for j = 0,...,79 do Step 15-20
Step 15: t1 = K(j)+W(j)+S(tmp(4),14,18,41)+Ch(tmp(4),tmp(5),tmp(6))+tmp(7)
Step 16: t2 = Maj(tmp(0),tmp(1),tmp(2)) +S(tmp(0),28,34,39)
Step 17: for k = 7,...,1 do Step 18
Step 18: tmp(k) = tmp(k −1)
Step 19: tmp(0) = t1+t2
Step 20: tmp(4) = tmp(4) +t1
Step 21: for j = 0,...,7 do Step 22
Step 22: H(j) = H(j) +tmp(j)
Step 23: i = i+128
Step 24: return H(0)||H(1)||H(2)||H(3)||H(4)||H(5)||H(6)||H(7)

Exercise HoHoHo:

Write a program which will produce SHA-512 hash. Check your hashing function for three messages below:

- message = „”
- message = „aaa”
- message = „aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa”

Compare your hashes with [an online hash calculator].