Secure Hash Algorithm 1(SHA-1): a hash algorithm used to convert data into a segment known as hash. SHA-1 was developed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST) in 1995. However, the use of SHA-1 in data security construction was warned because of its vulnerability. In 2005, a study indicated that SHA-1 could be compromised in an inexpensive way. In the following years, in-depth research provided additional evidence of SHA-1's vulnerability and exposure to collisions, a condition that occurs when there are two different series of data that are re-fragmenting. Due to these security vulnerabilities, SHA-1 has been set as an unsecured and not recommended algorithm in many sensitive uses, such as confirming emails and software signatures. Instead, it is best to use stronger and safer segment algorithms like SHA-256 or SHA-3 in apps that require high security.

Here is the complete algorithm of SHA-1:

step1. Divide the original text into blocks (blocks) of data size of 512 bits.

input text:

MOHAMED

text to Ascii:

109

111

104

97

109

101

100

Ascii to Binary:

01101101

01101111

01101000

01100001

01101101

01100101

01100100

step2. If the block is less than 512 bits, fill the difference with zero bits until you reach 512 bits.

step3. Increase the original length of the original text by structural representation and add it to the end of the last block. For example, if the original data contains 64 bits, the length will be constructively represented on 64 bits and add to the end of the last block.

step4. Divide each block into 16 32-bit words to perform operations on it.

step5. Expand the word list to 80 32-bit words by applying a specific function to the original words.

chunk 0

```
01101101011011110110100001100001   W[0]

01101101011001010110010010000000   W[1]

00000000000000000000000000000000   W[2]

00000000000000000000000000000000   W[3]

00000000000000000000000000000000   W[4]

00000000000000000000000000000000   W[5]

00000000000000000000000000000000   W[6]

00000000000000000000000000000000   W[7]

00000000000000000000000000000000   W[8]

00000000000000000000000000000000   W[9]

00000000000000000000000000000000   W[10]

00000000000000000000000000000000   W[11]

00000000000000000000000000000000   W[12]

00000000000000000000000000000000   W[13]

00000000000000000000000000000000   W[14]

00000000000000000000000000111000   W[15]
```

step6. Displacement of values in words based on the laws specified in the algorithm.

step7. Perform 80 cycles of operations on displaced words.

step8. Keep a steady initial value for each course and use it in all courses.

```
for I = 16 to 79

  I =16
```

W [i] = w[i-3] XOR w[i-8] XOR w[i-14] XOR w[i-16]

w[16]= w[16-3]XOR w[16-8] XOR w[16-14] XOR w[16-16]

#w[16]= w[13]XOR w[8] XOR w[2] XOR w[0]

#w[16]= 00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

01101101011011110110100001100001

#W[16]= 01101101011011110110100001100001

left rotated.

#W[16]= 11011010110111101101000011000010

Step9. Repeat the previous process until I reach word 80.

Step10. We arrange the80 words and divide them into 4groups, and each group contains 20words. The first group starts from (W0-W19) and takes function 1, and the second group starts from (W20-W39) and takes function 2, and so on for the rest of the groups.

A= h0= 01100111010001010010001100000001

B= h1= 11101111110011011010101110001001

C= h2= 10011000101110101101110011111110

D= h3= 00010000001100100101010001110110

E= h4= 11000011110100101110000111110000

FUNCTION1:

f= (B and C) or (!B and D)

FUNCTION2:

f= B xor C xor D

FUNCTION3:

f= (B and C) or(B and D) or(C and D)

FUNCTION4:

f= B xor C xor D

step11. We calculate the value of the temp variable and change the values of the variables.

TEMP= (A left rotated5) + F + E + K + current word.

E= D

D= C

C= B left rotated30

B= A

A= TEMP

Step12. After the courses are completed, you will get a final hash value of 160 bits. This is the full algorithm of SHA-1.

h0= h0+A

h1= h1+B

h2= h2+C

h3= h3+D

h4=h4+E