Name:-Amrit Ranjan

Reg No:-21BCE1847

Cryptography And Network Security

Assignment 4

Faculty Name:-Dr Tapabarta Roy

MD5 Algorithm Implementation



Ques :- Choose a 1000 bit message of your own and hash it using MD5 algorithm.

Code:-

Implementation using python:-

Hashlib is standard library in python

import math

def md5\_padding(message):

    original\_length = len(message) \* 8

    message += b'\x80'

    while len(message) % 64 != 56:

        message += b'\x00'

    message += original\_length.to\_bytes(8, byteorder='little')

    return message

def md5\_left\_rotate(x, n):

    return ((x << n) | (x >> (32 - n))) & 0xFFFFFFFF

def md5\_F(X, Y, Z):

    return (X & Y) | (~X & Z)

def md5\_G(X, Y, Z):

    return (X & Z) | (Y & ~Z)

def md5\_H(X, Y, Z):

    return X ^ Y ^ Z

def md5\_I(X, Y, Z):

    return Y ^ (X | ~Z)

def md5\_rounds(a, b, c, d, k, s, i, block):

    a = (b + md5\_left\_rotate((a + md5\_F(b, c, d) + block[k] + i) & 0xFFFFFFFF, s)) & 0xFFFFFFFF

    return d, a, b, c

def md5\_hash(message):

    # Initialize variables

    a0 = 0x67452301

    b0 = 0xEFCDAB89

    c0 = 0x98BADCFE

    d0 = 0x10325476

    # Pre-processing: padding the message

    padded\_message = md5\_padding(message)

    # Process each 512-bit block

    for i in range(0, len(padded\_message), 64):

        block = padded\_message[i:i+64]

        words = [int.from\_bytes(block[j:j+4], byteorder='little') for j in range(0, 64, 4)]

        A, B, C, D = a0, b0, c0, d0

        # Round 1

        for j in range(16):

            A, B, C, D = md5\_rounds(A, B, C, D, j, (j % 4) \* 7, (j % 4) \* 5 + 1, words)

        # Round 2

        for j in range(16):

            k = (j \* 5 + 1) % 16

            A, B, C, D = md5\_rounds(A, B, C, D, k, (j % 4) \* 5 + 16, (j % 4) \* 3 + 5, words)

        # Round 3

        for j in range(16):

            k = (j \* 3 + 5) % 16

            A, B, C, D = md5\_rounds(A, B, C, D, k, (j % 4) \* 7 + 32, (j % 4) \* 7, words)

        # Round 4

        for j in range(16):

            k = j % 4

            A, B, C, D = md5\_rounds(A, B, C, D, k, (j % 4) \* 5 + 48, (j % 4) \* 3 + 9, words)

        a0 = (a0 + A) & 0xFFFFFFFF

        b0 = (b0 + B) & 0xFFFFFFFF

        c0 = (c0 + C) & 0xFFFFFFFF

        d0 = (d0 + D) & 0xFFFFFFFF

    # Concatenate A, B, C, D to get the final hash

    return (a0.to\_bytes(4, byteorder='little') +

            b0.to\_bytes(4, byteorder='little') +

            c0.to\_bytes(4, byteorder='little') +

            d0.to\_bytes(4, byteorder='little'))

# Example usage:

message = b"Hello, this is a test message for MD5 hashing!"

hashed\_result = md5\_hash(message)

print("MD5 Hash:", hashed\_result.hex())

Output:-

