南京邮电大学

《密码学》实验报告（二）

（ 2024 / 2025 学年 第 二 学期）

题　　目：Hash 算法实现

专　　业	信息安全
学 号 姓 名	B230410
	B23041011
	于明宏
指 导 教 师	李琦
指 导 单 位	计算机学院、软件学院、网络空间安全学院
日　　期	2025.4.30

分组密码算法实现

一、课题内容和要求

本实验的目标是实现 SM3 算法。

二、实现分析

SM3 算法对数据首先进行填充，再进行迭代压缩后生成哈希值。

三、概要设计

采用 Python 语言编写，完整实现 SM3 算法，并调用 Hashlib 密码学库对其结果进行验证。

四、源程序代码

```
import hashlib

MAX_32 = 0xffffffff

def lshift(x, i):
    return ((x << (i % 32)) & MAX_32) | (x >> (32 - i % 32))

def T(j):
    return 0x79cc4519 if j <= 15 else 0x7a879d8a

def FF(j, x, y, z):
    return (x ^ y ^ z) if j <= 15 else ((x & y) | (x & z) | (y & z))

def GG(j, x, y, z):
    return (x ^ y ^ z) if j <= 15 else ((x & y) | (~x & z))

def P0(x):
    return x ^ lshift(x, 9) ^ lshift(x, 17)

def P1(x):
    return x ^ lshift(x, 15) ^ lshift(x, 23)

def fill(s):
    m = ''.join([bin(ord(c))[2:].zfill(8) for c in s])
    l = len(m)
    m += '1'
    k = (448 - (l + 1)) % 512
    m += '0' * k
    m += bin(l)[2:].zfill(64)
```

```
    return hex(int(m, 2))[2:].zfill(len(m) // 4)

def sm3(s):
    V = 0x7380166f4914b2b9172442d7da8a0600a96f30bc163138aae38dee4db0fb0e4e
    m = fill(s)

    for i in range(len(m) // 128):
        Bi = m[i * 128:(i + 1) * 128]
        W = [int(Bi[j * 8:(j + 1) * 8], 16) for j in range(16)]

        for j in range(16, 68):
            W.append(P1(W[j - 16] ^ W[j - 9] ^ lshift(W[j - 3], 15)) ^ lshift(W[j - 13], 7) ^ W[j -
6])

        W_ = [W[j] ^ W[j + 4] for j in range(64)]

        A, B, C, D, E, F, G, H = [(V >> (224 - i * 32)) & MAX_32 for i in range(8)]

        for j in range(64):
            ss1 = lshift((lshift(A, 12) + E + lshift(T(j), j)) & MAX_32, 7)
            ss2 = ss1 ^ lshift(A, 12)
            tt1 = (FF(j, A, B, C) + D + ss2 + W_[j]) & MAX_32
            tt2 = (GG(j, E, F, G) + H + ss1 + W[j]) & MAX_32
            D, C, B, A = C, lshift(B, 9), A, tt1
            H, G, F, E = G, lshift(F, 19), E, P0(tt2)

        V ^= ((A << 224) | (B << 192) | (C << 160) | (D << 128) |
              (E << 96) | (F << 64) | (G << 32) | H)

    return hex(V)[2:].zfill(64)

data = input("Please input your string: ")

my_hash = sm3(data)
print("SM3 hash 1 by B23041011: ", my_hash)

sm3_obj = hashlib.new('sm3')
sm3_obj.update(data.encode('utf-8'))
lib_hash = sm3_obj.hexdigest()
print("SM3 hash 2 by Hashlib:     ", lib_hash)

if my_hash == lib_hash:
    print("Correct!")
else:
    print("Wrong!")
```

五、测试数据及其结果分析

Please input your string: I love Nanjing Universitry of Posts and Telecommunications very much!
SM3 hash 1 by B23041011: 3a00a9c5af9e1cc7fadbf728a3092519ae94cbbf55010078d076134a144b08d2
SM3 hash 2 by Hashlib: 3a00a9c5af9e1cc7fadbf728a3092519ae94cbbf55010078d076134a144b08d2
Correct!

六、调试过程中的问题

调试过程中未出现问题。

七、课程总结

通过本次实验，深入理解了 SM3 密码的工作原理，掌握了基于 Python 的数据结构和算法的实现方法。