Experimental report——the naïve birthday attack of reduced SM3

姓名：李祥方

学号：201900460041

1 前置知识

*SM*3**的定义**：SM3密码杂凑算法是中国国家密码管理局2010年公布的中国商用密码杂凑算法标准。具体算法标准原始文本参见参考文献[1]。该算法于2012年发布为密码行业标准(GM/T 0004-2012)，2016年发布为国家密码杂凑算法标准(GB/T 32905-2016)。SM3适用于商用密码应用中的数字签名和验证，是在[SHA-256]基础上改进实现的一种算法，其安全性和SHA-256相当。SM3和MD5的迭代过程类似，也采用Merkle-Damgard结构。消息分组长度为512位，摘要值长度为256位。整个算法的执行过程可以概括成四个步骤：消息填充、消息扩展、迭代压缩、输出结果。

生日攻击方法:

1 生日攻击

生日攻击是利用概率论中的生日问题，找到冲突的Hash值，伪造报文，使身份验证算法失效。

生日攻击的理论描述有些复杂，不易理解，请参考相关资料。
本文以实例方式介绍生日攻击方法和防范方法。

> **关于生日问题的简单说明：**
> 如果输出是256位，我们随机地选择输入，并计算哈希值，在检验第2^{256}+1个输入之前便很可能找到碰撞。
> 实际上，如果我们随机选择2^{130}+1个输入，找到至少两个相同哈希值的概率为99.8%。
> 仅仅通过检验可能输出数量的平方根次数，便大体能找到碰撞，这在概率论中称为生日悖论（birthday paradox）。

2 实验过程

根据以上前置知识，编写了以下的代码，**代码寻找的是32位的碰撞**.

```python
def getRandomList(n):
    numbers = []
    while len(numbers) < n:
        i = random.randint(0, 2**48)
        if i not in numbers:
            numbers.append(bytes(str(hex(i)[2:]),encoding='utf-8'))
    return numbers

num1 = getRandomList(100000)
num2 = getRandomList(100000)
OUT1=[]
OUT=[]
OUT_number_1=[]
OUT_number_2=[]
#采用两个循环的方式，对于随机产生的数据集合进行哈希
for i in num1:
    outcome = sm3.sm3_hash(func.bytes_to_list(i))[0:8]#设置了32位的碰撞
    OUT1.append(outcome)
for i in num2:
    outcome = sm3.sm3_hash(func.bytes_to_list(i))[0:8]
    if outcome in OUT1:#判断是否有32位的碰撞
        OUT.append(outcome)
```

3 实验结果

运行代码，找到了32位的碰撞，碰撞的输入对是'10c702165a08','841870a199a0'.测试运行结果如下(**图中数据为16进制**)：

```python
str_b = bytes('cea63e735e9a', encoding='utf-8')
result = sm3.sm3_hash(func.bytes_to_list(str_b))
print(result)  #50f03b05d10fa07f1169aff1d1e119ae3169107035b1abd24f76009ee05a8e2c
print(len(result))
str_b = bytes('3d8ef82cf0e4', encoding='utf-8')
result = sm3.sm3_hash(func.bytes_to_list(str_b))
print(result)  #50f03b05d10fa07f1169aff1d1e119ae3169107035b1abd24f76009ee05a8e2c
print(len(result))
```

```
023c1e5fa069dbcf734755f8107fda76886f1ca093ca586aae2cf06e79018466
64
023c1e5f391bb5543e7cbb9909696e17d0ae84dfbdbc6ea14a0ed0de6ef5a65a
64
```

由运行结果可知，是**发生了32位的碰撞**.