山东大学网络空间安全学院

网络空间安全创新创业实践

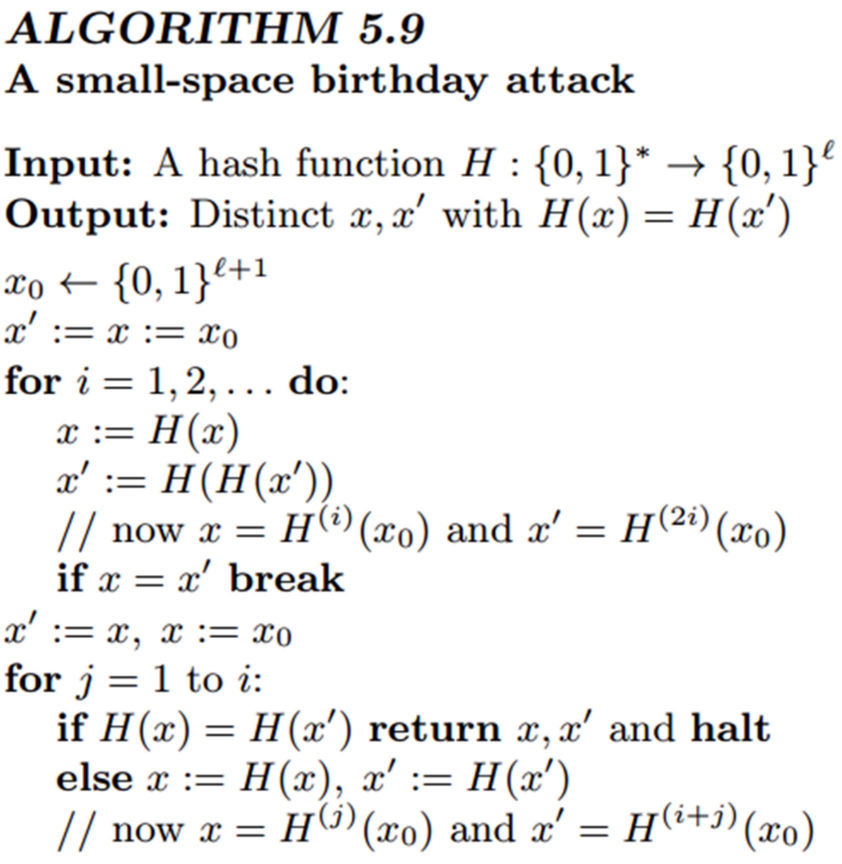
Project 2 Rho Method寻找reduced SM3 碰撞

姓名：张麟康

学号：201900301107

1 原理分析

Rho方法的具体算法描述如下所示。



ALGORITHM 5.9

A small-space birthday attack

Input: A hash function $H : \{0,1\}^* \to \{0,1\}^\ell$

Output: Distinct x, x' with $H(x) = H(x')$

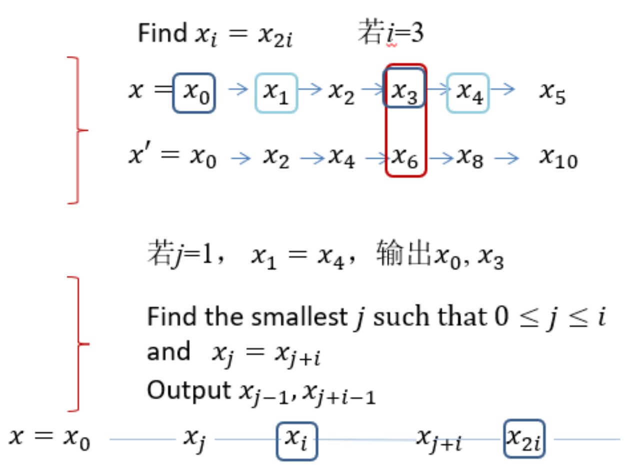
```
$x_0 \leftarrow \{0,1\}^{\ell+1}$
$x' := x := x_0$
for $i = 1, 2, \ldots$ do:
    $x := H(x)$
    $x' := H(H(x'))$
    // now $x = H^{(i)}(x_0)$ and $x' = H^{(2i)}(x_0)$
    if $x = x'$ break
$x' := x,\ x := x_0$
for $j = 1$ to $i$:
    if $H(x) = H(x')$ return $x, x'$ and halt
    else $x := H(x),\ x' := H(x')$
    // now $x = H^{(j)}(x_0)$ and $x' = H^{(i+j)}(x_0)$
```

其具体实现过程如下：首先随机选取$l+1$长的x_0并成对计算$x_i = H^{(i)}(x_0)$和$x_{2i} = H^{(2i)}(x_0)$，对比x_i和x_{2i}，若二者相等，则序列$x_0, .., x_{2i-1}$存在碰撞，从而只需要找到最小的$0 \leq j \leq i$使得$x_j = x_{j+i}$并输出x_{j-1}, x_{j+i-1}即可。

1.随机选取l+1长的x_0, 并成对计算$x_i = H^{(i)}(x_0)$, $x_{2i} = H^{(2i)}(x_0)$, i=1,2,...

2.对比x_i, x_{2i}, 若相等, 则序列$x_0, x_1, ..., x_{2i-1}$存在碰撞

3.找最小的 $0 \le j \le i$, 使得 $x_j = x_{j+i}$,输出x_{j-1}, x_{j+i-1}



Find $x_i = x_{2i}$ 若i=3

$x = \boxed{x_0} \to \boxed{x_1} \to x_2 \to \boxed{x_3} \to \boxed{x_4} \to x_5$

$x' = x_0 \to x_2 \to x_4 \to x_6 \to x_8 \to x_{10}$

若j=1, $x_1 = x_4$, 输出x_0, x_3

Find the smallest j such that $0 \le j \le i$
and $x_j = x_{j+i}$
Output x_{j-1}, x_{j+i-1}

$x = x_0$ — x_j — $\boxed{x_i}$ — x_{j+i} — $\boxed{x_{2i}}$

2 具体实现

Rho方法实现寻找碰撞的过程相对比较简单，首先定义缩减输出的SM3杂凑函数便于具体操作，只需要简单截取标准SM3实现的前TRUNC个字即可（每个字为一个16进制数4比特，因此每次可以成功找到4×TRUNC比特长度的碰撞）。

```
def reduced_sm3(m):
    return sm3.SM3(m)[:TRUNC]
```

与项目1类似，为了便于寻找碰撞，定义生成随机字符串的方法get_random_input()，该函数利用random模块中的sample方法生成一个长度为N的随机字符串。

```
def get_random_input():
    random_input = ''.join(random.sample(char_set, N))
    res = ''
    for x in random_input:
        res += str(ord(x))
    return res
```

定义一个变量MAX_TRIAL表示允许最大尝试寻找碰撞的次数，定义一个字典数据结record用于记录每次生成随机字符串及其对应的哈希值以便后续寻找碰撞。

```python
start_time = time.time()
N = 32
TRUNC = 32
char_set = string.ascii_letters+string.digits
trial_times = 0
record = {}
MAX_TRIAL=12000000
```

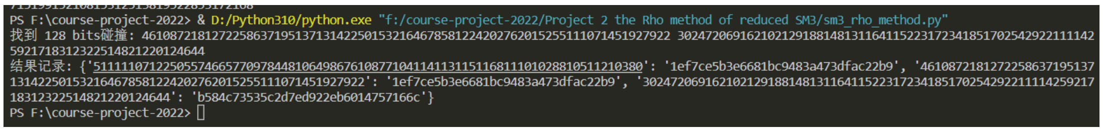
调用get_random_input()生成一个随机初始字符串h，声明另一个字符串h_与其值一致。

```python
record={}
h=get_random_input()
h_=h
```

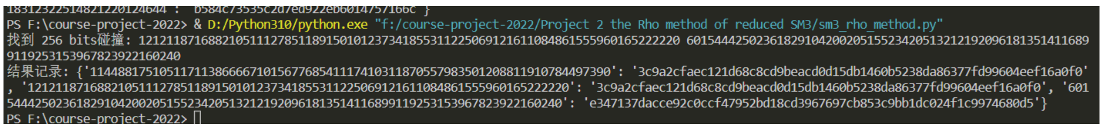
之后进入循环，每次对字符串h进行一次哈希，对h_进行两次哈希，记录二者哈希值的前TRUNC个字部分，如果发生碰撞，则输出h和h_，否则一直处于循环之中直至达到MAX_TRIAL，随后更换初始字符串重新进行尝试。

```python
for i in range(MAX_TRIAL):
    hash=sm3.SM3(h)
    hash_=sm3.SM3(hex2ord(sm3.SM3(h)))
    record[h]=hash[:TRUNC]
    record[h_]=hash_[:TRUNC]
    if len(record) !=len(set(record.values())):
        print("找到",TRUNC*4,'bits碰撞:',h,h_)
        break
    h=hex2ord(hash)
    h_=hex2ord(hash_)
```

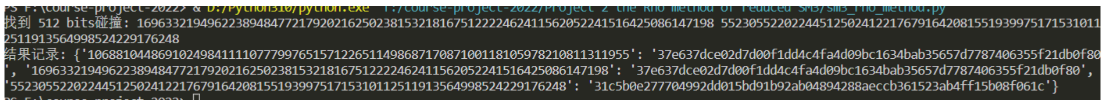
运行结果如下图所示。



```
PS F:\course-project-2022> & D:/Python310/python.exe "f:/course-project-2022/Project 2 the Rho method of reduced SM3/sm3_rho_method.py"
找到 128 bits碰撞: 4610872181272258637195137131422501532164678581224202762015255111071451927922 3024720691621021291881481311641152231723418517025429221111425921718312322514821220124644
结果记录: {'5111110712250557466577097844810649867610877104114113115116811101028810511210380': '1ef7ce5b3e6681bc9483a473dfac22b9', '4610872181272258637195137131422501532164678581224202762015255111071451927922': '1ef7ce5b3e6681bc9483a473dfac22b9', '30247206916210212918814813116411522317234185170254292211114259217183123225148212201246 44': 'b584c73535c2d7ed922eb6014757166c'}
PS F:\course-project-2022> 
```



```
PS F:\course-project-2022> & D:/Python310/python.exe "f:/course-project-2022/Project 2 the Rho method of reduced SM3/sm3_rho_method.py"
找到 256 bits碰撞: 121211871688210511127851189150101237341855311225069121611084861555960165222220 601544425023618291042002051552342051321219209618135141168991192531539678239221602 40
结果记录: {'1144881751051171138666671015677685411174103118705579835012088119107844973 90': '3c9a2cfaec121d68c8cd9beacd0d15db1460b5238da86377fd99604eef16a0f0', '121211871688210511127851189150101237341855311225069121611084861555960165222220': '3c9a2cfaec121d68c8cd9beacd0d15db1460b5238da86377fd99604eef16a0f0', '601544425023618291042002051552342051321219209618135141168991192531539678239221602 40': 'e347137dacce92c0ccf47952bd18cd3967697cb853c9bb1dc024f1c9974680d5'}
PS F:\course-project-2022> 
```



```
PS F:\course-project-2022> & D:/Python310/python.exe f:/course-project-2022/Project 2 the Rho method of reduced SM3/sm3_rho_method.py
找到 512 bits碰撞: 16963321949622389484772179202162502381532181675122224624115620522415164250861471198 552305522022445125024122176791642081551939975171531011
2511913564998524229176248
结果记录: {'1068810448691024984111107779976515712265114986871708710011810597821081131955': '37e637dce02d7d00f1dd4c4fa4d09bc1634bab35657d7787406355f21db0f80
', '16963321949622389484772179202162502381532181675122224624115620522415164250861471198': '37e637dce02d7d00f1dd4c4fa4d09bc1634bab35657d7787406355f21db0f80',
'552305522022445125024122176791642081551939975171531011251191356499852422917 6248': '31c5b0e277704992dd015bd91b92ab04894288aeccb361523ab4ff15b08f061c'}
PS F:\course-project-2022> 
```