Rho method

Project: implement the Rho method of reduced SM3

Rho method 是生日悖论的随机方法的改进。首先随机取值，将取得的值放入 sm3 加密，得到的结果再次代入加密函数，由此得到一个顺序的数列。由于函数的特性，在一定范围内可以保证数列的随机性。当出现循环时，此时找到碰撞。

代码说明：

首先声明一个随机取值的生成函数，随机取一个值作为函数开头。

```
def Random():
    alphabet = 'abcdefghijklmnopqrstuvwxyz0123456789'
    data=[]
    for i in range(128):
        data.append(random.choice(alphabet))
    return (''.join(data))
```

将随机值代入 sm3 中，将得到的结果的前 28 比特存入列表，再将结果对应的数据存入列表。将结果再次代入加密函数，得到一个顺序的列表。当哈希值相等时出现环，找到碰撞。输出碰撞对。

```
def rho():
    lst=[]
    temp=Random()#取随机值
    temph=sm3h(temp)#加密
    lst.append(temph[0:7])#将结果的前28比特存入列表
    lst.append(temp)#将对应的数据存入列表
    while 1:
        temp=temph
        temph=sm3h(temph)
        i=0
        for i in range(0,len(lst),2):
            if (lst[i]==temph[0:7])&(temp!=lst[i+1]):#当哈希值相等时出现环，找到
                print("找到碰撞，碰撞为\n%s\n%s"%(temp,lst[i+1]))
                return 0
        lst.append(temph[0:7])
        lst.append(temp)

rho()
```

运行指导：

可改变 temph 切片长度找到其他长度加密值相等的碰撞。

运行结果：

```
========== RESTART: C:\Users\ano\Desktop\rho.py ==========
找到碰撞，碰撞为
43E905A3BD719C895EAD145919282C7C34565D11734D55C07BA3ED1B692DA0B8
71B715358D83B8D13E042291B5151EDE8A2C301C78F1FF3DF49228A0886B9789
```