**\*Project1: implement the naïve birthday attack of reduced SM3**

代码说明：

此项目是实现SM3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。生日攻击的基本思想是寻找两个不同的输入消息，它们经过SM3算法后产生相同的hash值。根据生日悖论，n位的哈希值经过2^(n/2)次尝试预计可以约1/2的概率产生一次碰撞。

本项目我先用python实现了FF、GG、p0、p1、消息填充、消息扩展、压缩、迭代等函数，进而实现了SM3算法，再进行生日攻击，实现SM3生日攻击的前n位比特的碰撞的步骤如下：

（1）定义v函数，接受参数n，生成一个随机8位（可改为自己想要的位数）字符串s，并将其转换为字节数组a。然后，对a进行SM3哈希，并截取前n位比特得到j。最后，将a和j作为一对返回。

（2）定义birthday\_attack函数，接受参数n，将n转换为前n位比特数，并调用v函数生成两个不同的pair（a1, j1）和（a2, j2）。然后，在for循环中比较a1、a2，j1、j2。如果找到两个不同的字符串（即a1≠a2），且它们的前n位哈希值相同（即j1=j2），则认为攻击成功。如果不满足，则继续生成新的随机字符串对，并进行下一轮循环。

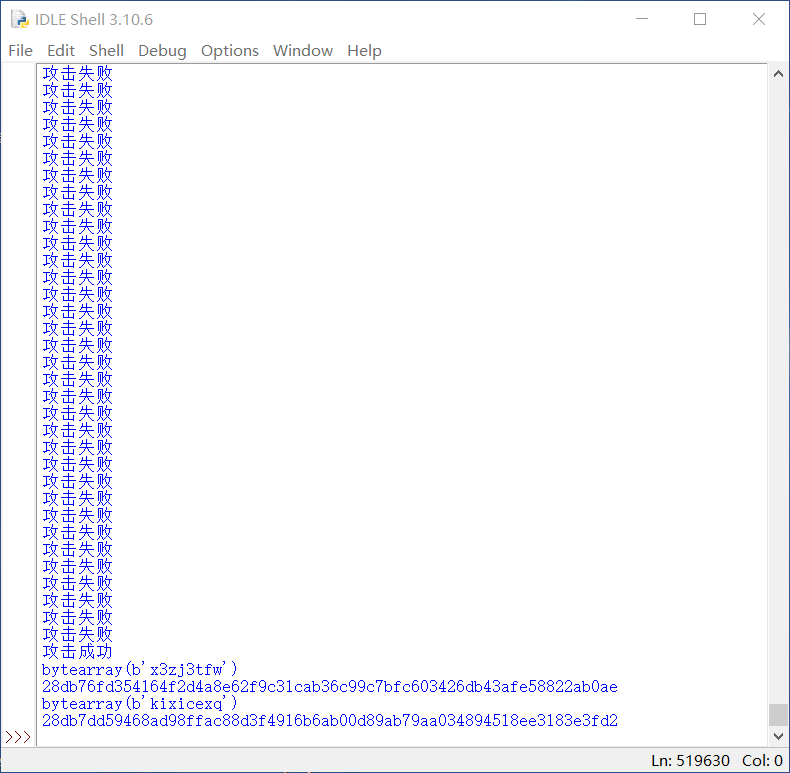
Ps：birthday\_attack(n)函数中的n=int(n / 4)是因为产生的随机消息一位单位为一个字节，除4后单位为1bit。

birthday\_attack(n)中可根据需要将函数变量n改为任意整数。

实现方式：python

效果：在自己电脑上CPU：11代i7

这里以birthday\_attack(20)为例，展示其结果：



分工：自己独立完成