1901210488时绍森

SM3长度拓展攻击原理：

长度扩展攻击（length extension attack），是指针对某些允许包含额外信息的加密散列函数的攻击手段。对于满足以下条件的散列函数，都可以作为攻击对象： ① 加密前将待加密的明文按一定规则填充到固定长度（例如512或1024比特）的倍数；

② 按照该固定长度，将明文分块加密，并用前一个块的加密结果，作为下一块加密的初始向量（Initial Vector）

对于长度拓展攻击，要知道以下信息：

1. 不知道salt+data，但知道salt+datat的长度

2. 知道Hash(salt+data)的值。

具体步骤如下：

1.填充

        拿到明文后，SM3先将明文转为二进制文件，然后将二进制文件的长度除以512比特（即64字节），如果余数等于448比特（即64-8字节），那么直接在后面加上八个字节的长度标识，使之成为512比特的倍数。否则则在明文后填一个1，再填充0直至其长度除以512等于448，再加上8位的长度标识。长度是使用大端序来存储，即低字节放在高地址位上。

2.分块运算

         填充完毕后，函数就将填充后的明文以512比特的长度分块，进行运算。在运算中会用到四个初始向量（称为链变量，Chaining Variable），分别是A=0x67452301，B=0xefcdab89，C=0x98badcfe，D=0x10325476。经过一系列复杂的数学运算，函数会得到第一块的SM3值，然后将该SM3值分成四块，以大端序形成新的链变量，投入到第二块的运算，形成新的SM3值……以此类推，直到算出最后一块的SM3值，就是整个数据块的SM3值。

3.攻击实施

如果攻击者知道SM3(salt+data)的值并可控制data的值，我们可以设定data为与data+padding等长的任意字符串（在程序中用100...0和长度进行填充），然后计算SM3(str+append)。我们知道，SM3需要先填充再运算，攻击者可以在程序计算append所在块之前，将SM3(salt+data)的值直接替换掉初始的链变量，就能够算出SM3(salt+data+padding+append)的值了。设SM3(CV,data)表示以链变量CV计算data的SM3值，那么SM3(IV, salt+data+padding+append) = SM3(SM3(IV, salt+data), append)。

## 代码结构：

func.py和sm3Encrypt.py是用于利用密钥填充后的hash值替换新的IV进行第二轮sm3加密，使最后hash结果一样。

sm3.py是主函数，包含了攻击的填充部分以及攻击最后验证的过程。

执行时执行sm3.py程序，它包括了程序的入口main函数