实验报告

【实验目的】

1. 通过本次实验，熟练掌握 $SM4$ 加解密流程；
2. 通过本次实验，了解并掌握各种工作模式；
3. 感受工作模式与填充方式对安全性的意义。

【实验环境】

1. 语言：C
2. 平台：clion 2021.2 版本

【实验内容】

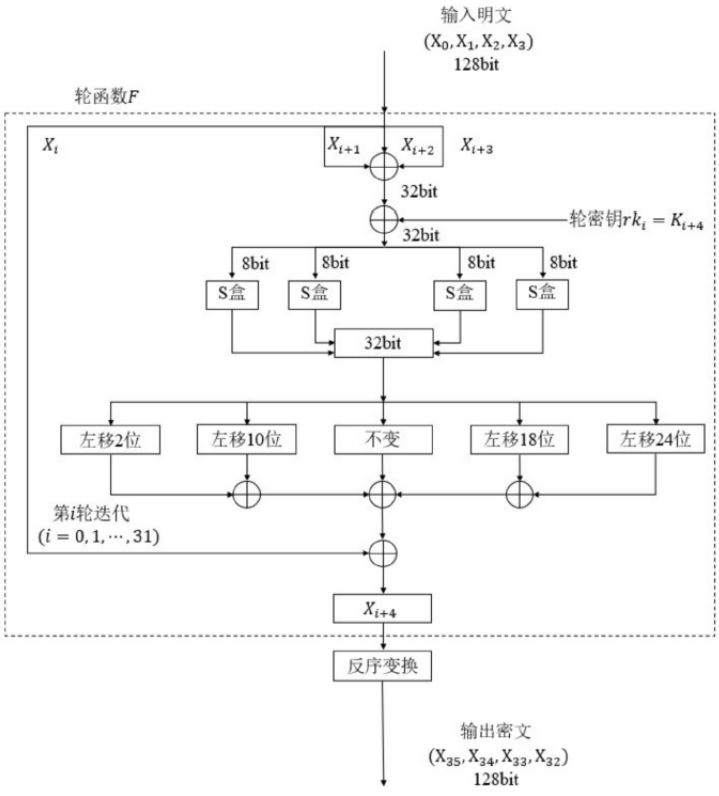
一、SM4算法

1. 算法流程

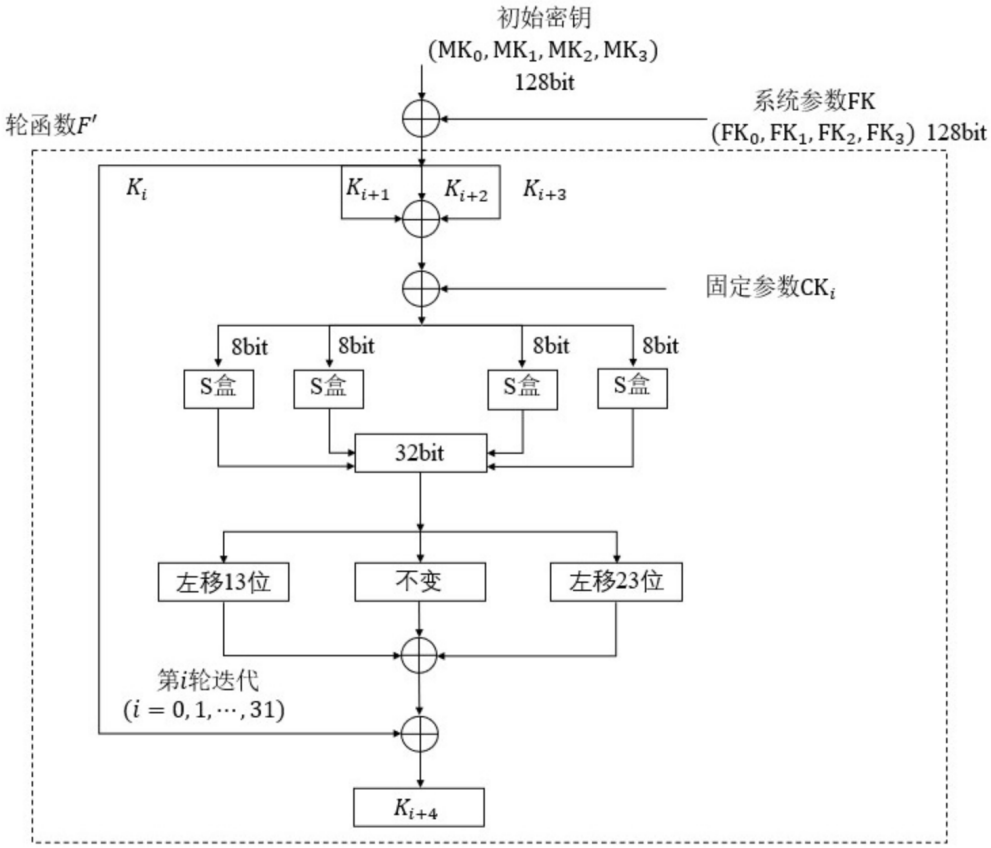
与DES与AES算法相似，$SM4$也是一种分组密码算法。其分组长度为128bit，密钥长度为128bit，加密算法与密钥扩展算法均采用32轮非线性迭代结构，以字（32位）为单位进行加密运算，每一次迭代运算均为一轮变换函数F，$SM4$算法加/解密算法的结构相同，只是使用轮密钥相反，其中解密轮密钥是加密轮密钥的逆序。

- **流程图：**

1. **SM4加密结构流程图：**



2. **密钥拓展算法：**



- **伪代码：**

1. **加密算法伪代码：**

算法 1 SM4加密算法

输入： rk轮密钥,$plain-text$

输出： $cipher-text$

function ENCRYPTION($plain-text, rk$)
  $(X_0, X_1, X_2, X_3) \leftarrow plain-text$
  for $i = 0 \rightarrow i = 31$ **do**
    $X_{i+4} \leftarrow F(X_i, X_{i+1}, X_{i+2}, X_{i+3})$
  end for
  $cipher-text \leftarrow Inverse(X_{32}, X_{33}, X_{34}, X_{35})$
  return $cipher-text$
end function

2. **F函数伪代码：**

算法 1 迭代函数F

输入： $X_i, X_{i+1}, X_{i+2}, X_{i+3}$,$rk_i$

输出： X_{i+4}

function FUNCTION$_F$($X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i$)
  $X \leftarrow X_{i+1} \bigoplus X_{i+2} \bigoplus X_{i+3} \bigoplus rk_i$
  $X \leftarrow Sbox(X)$
  $X_{i+4} = X_i \bigoplus (X << 2) \bigoplus (X << 10) \bigoplus (X << 18) \bigoplus (X << 24) \bigoplus X$
  return X_{i+4}
end function

3. **密钥拓展算法伪代码：**

算法 1 密钥扩展算法

输入： key,Fk,Ck

输出： rk

function KEY-EXPANSION(key, Ck, Fk)
  $(K_0, K_1, K_2, K_3) \leftarrow key \bigoplus Fk$
  for $i = 0 \rightarrow i = 31$ **do**
    $K_{i+4} \leftarrow F'(K_i, K_{i+1}, K_{i+2}, K_{i+3}, Ck)$
  end for
  $rk \leftarrow K$
  return rk
end function

4. **迭代函数算法伪代码：**

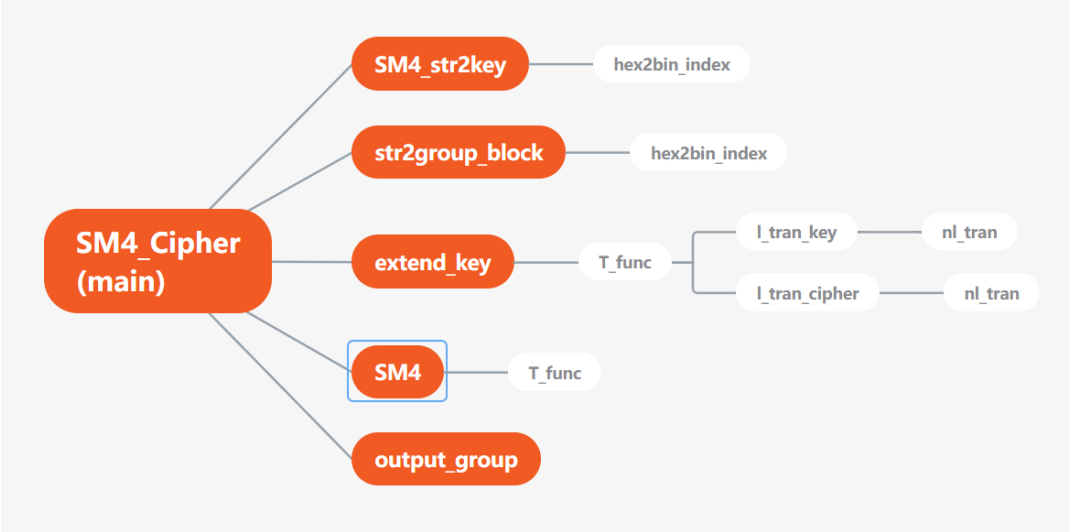
算法 1 迭代函数F'

输入： $K_i, K_{i+1}, K_{i+2}, K_{i+3}$,$ck_i$

输出： K_{i+4}

function FUNCTION$'_F$(K$_i, K_{i+1}, K_{i+2}, K_{i+3}, Ck_i$)
  $K \leftarrow K_{i+1} \bigoplus K_{i+2} \bigoplus K_{i+3} \bigoplus Ck_i$
  $K \leftarrow Sbox(K)$
  $K_{i+4} = K_i \bigoplus (K << 13) \bigoplus (K << 23) \bigoplus K$
  return K_{i+4}
end function

- **函数调用图：**

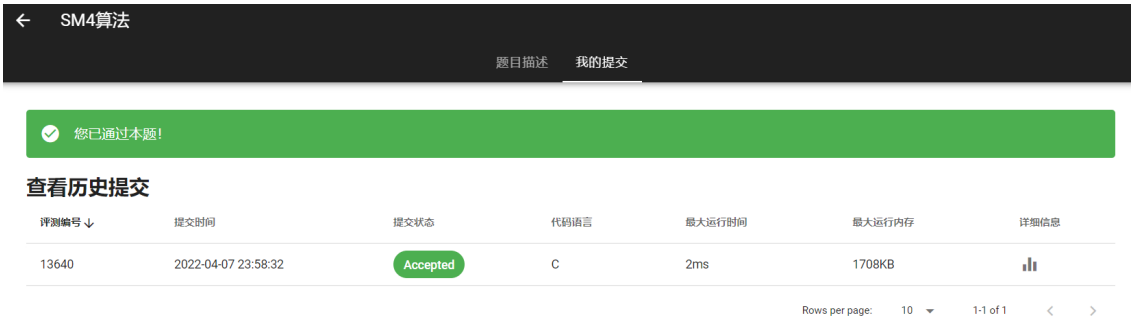


2. **测试样例及结果截图：**

本地测试样例的运行结果如下所示：

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\SM4_cipher.exe"
39
0x52a4fd41fee4f0bacfb0a3347b52f55d
0x0b93875661edea7fcd44c8643c76cfc6
1
0x8dd5a004b00ff3c9eb6bc513964e7d3b
Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\SM4_cipher.exe"
1
0xc04a9b311a2fc245f742c5719fcf249d
0x4046fb1985d94a7f1ff55ec7ec5f6054
0
0x6b956ddb0faff373bc338cb600739f23
Process finished with exit code 0
```



3. **讨论与思考：**

二、SM4_ECB

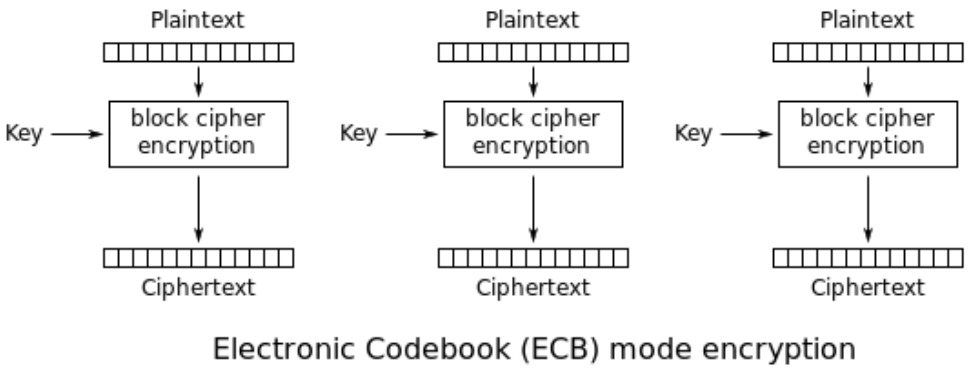
1. **填充标准PKCS#7**

该标准即：按分组长度对明文分组，最后一组可能长度小于分组长度b，那么剩余的位置按剩余长度代表的字节来填充。例如分组的16进制长度为32位（即16字节），还差7个字节，就按0x07填充7个字节。特别的，当明文长度恰好是32的倍数时，依然要在最后一行填充一整个分组长度的明文，以便在解密时具有统一性。

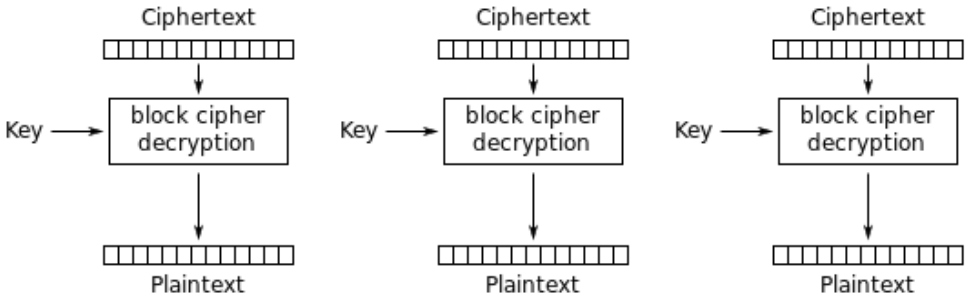
解密时保证密文长度一定是32的倍数，读取最后一个字节的值，删去倒数该值个字节即可恢复为明文。

2. **电码本模式ECB**

电码本模式：即用相同的密钥分别对明文分组分别加密：

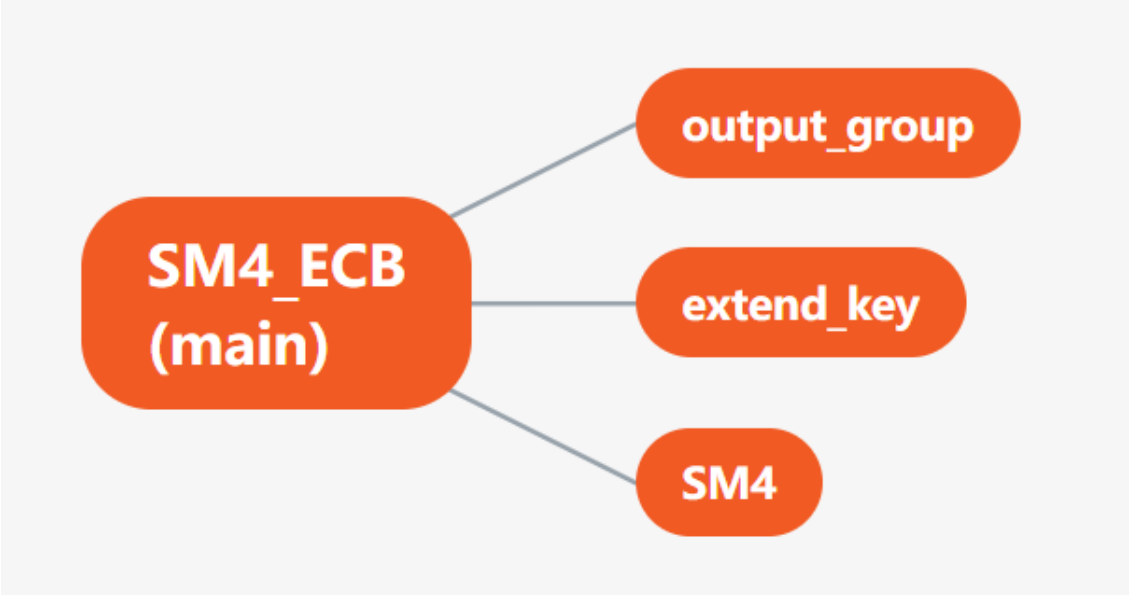


Electronic Codebook (ECB) mode encryption

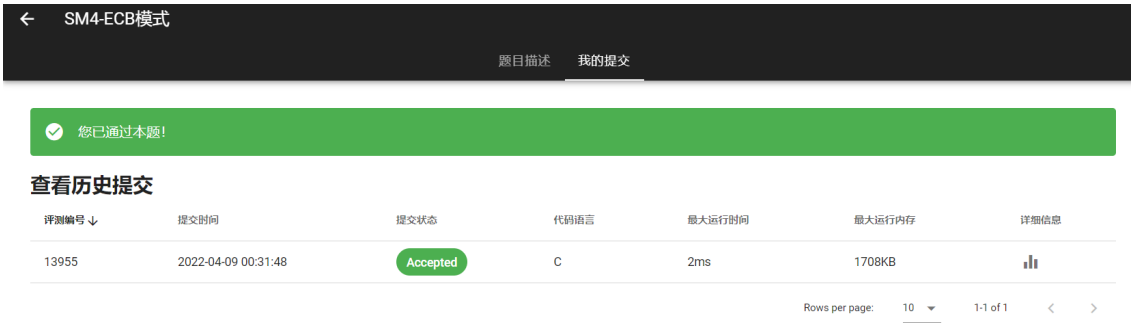


Electronic Codebook (ECB) mode decryption

3. **函数调用图：**

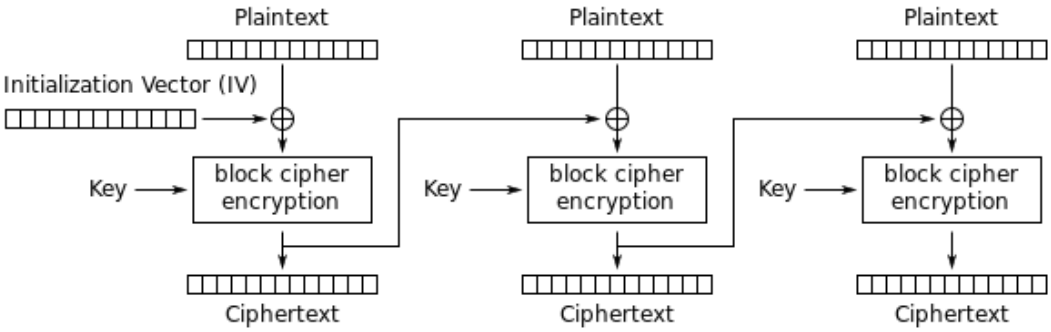


4. **测试样例及结果截图：**

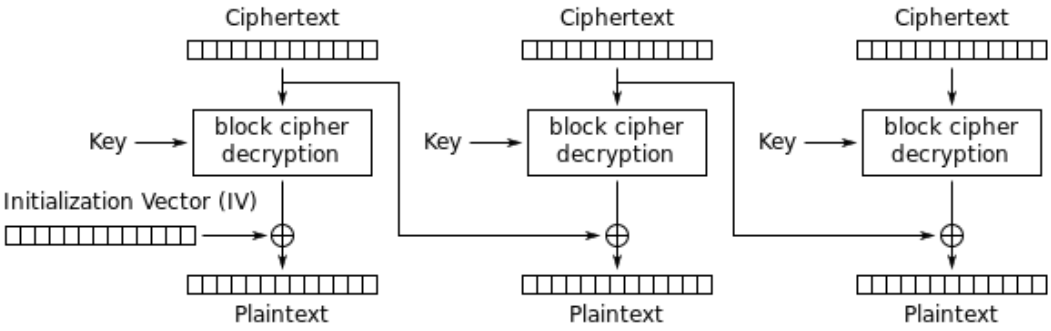


三、SM4_CBC

1. **加密算法的输入是上一个密文分组和下一个明文分组的异或。**

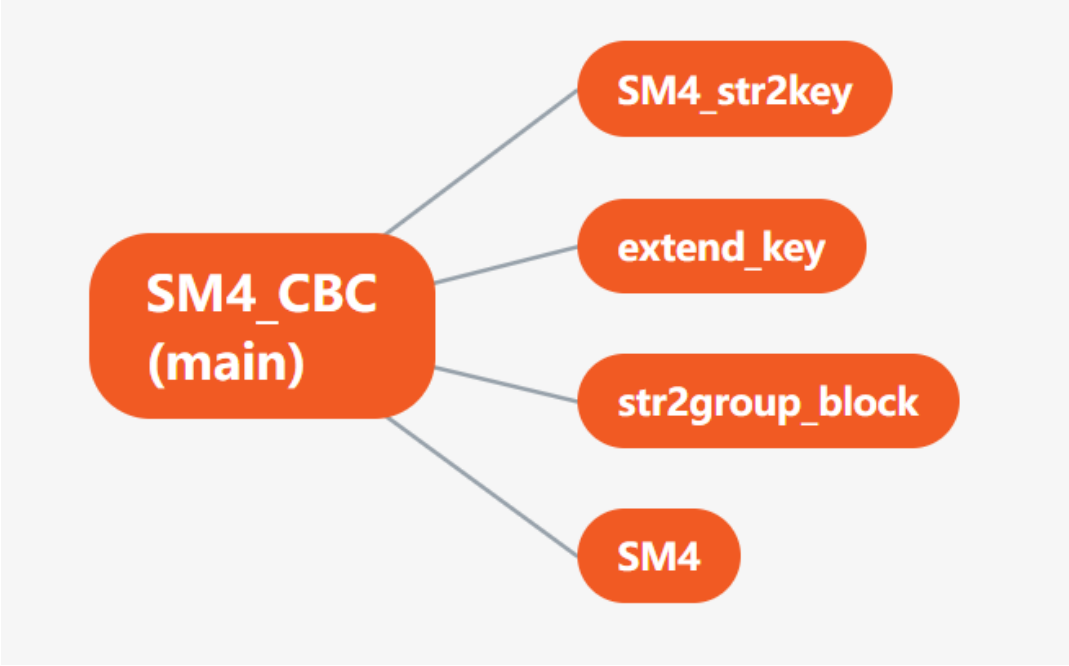


Cipher Block Chaining (CBC) mode encryption



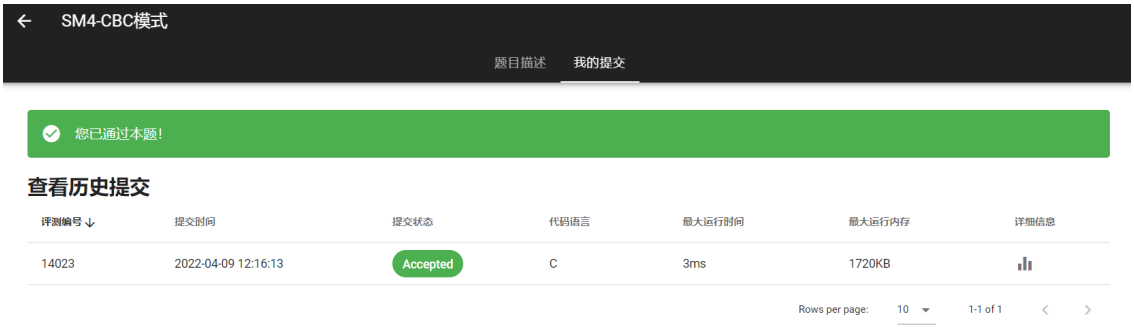
Cipher Block Chaining (CBC) mode decryption

2. **算法流程：**
   - **函数调用图：**



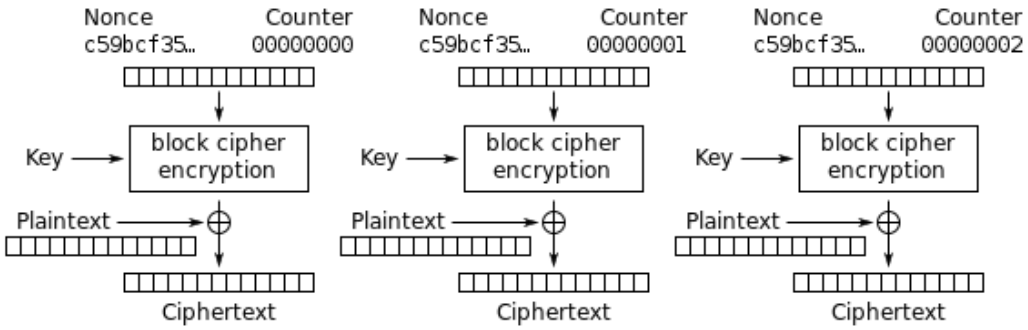
3. **测试样例及结果截图：**

```
0x02
0x7a 0xc7 0xbb 0x8d 0xe9 0x2b 0xa9 0x7a 0x3b 0xa2
0x38 0x53 0xaa 0xb0 0xc3
0x6c
0xfe 0x6f 0x88 0xd6 0x26 0xd4 0xe5 0x81 0x5c 0xdf
^D
0x2a 0x3f 0x54 0x40 0x48 0x1e 0x21 0x9d 0xac 0x9b 0x27 0xef 0x69 0xa9 0xff 0x72
0x44 0x1f 0x1b 0x4d 0x06 0x04 0x97 0x23 0x03 0x6c 0x33 0x95 0x6d 0x47 0xf2 0x93
0x41 0x5f 0x7c 0xf7 0x44 0x4d 0x71 0xeb 0x14 0x56 0x6c 0xc4 0x7c 0xa7 0x77 0x3f
0x64 0x5a 0x06 0x8d 0x1b 0xdc 0xa2 0x5a 0x40 0x45 0x4b 0x4e 0xc8 0x20 0xa6 0xb0
0x36 0xb7 0xa1 0xf4 0x09 0x7d 0x8d 0xcf 0xbe 0xa1 0x95 0x9b 0xfe 0xe4 0x6e 0x3e
0x51 0xf2 0x34 0xa3 0x98 0xca 0x06 0xa7 0x3d 0x3d 0xea 0x13 0x82 0x96 0x54 0x3e
0x6d 0xc8 0x00 0xb1 0x39 0x46 0xfc 0xf0 0x0c 0x9e 0x9c 0x5b 0x3b 0xc3 0x77 0x6e
0xce 0xb6 0x9f 0xd0 0x5e 0x34 0xde 0x13 0x3e 0x9f 0x3d 0x1f 0x59 0xdf 0xc7 0x10
0xfe 0xff 0x02 0xed 0x3f 0xe7 0x96 0x64 0xd6 0x96 0xd6 0xec 0x54 0xd4 0xae 0xe0
0xe0 0x2e 0x77
Process finished with exit code 0
```

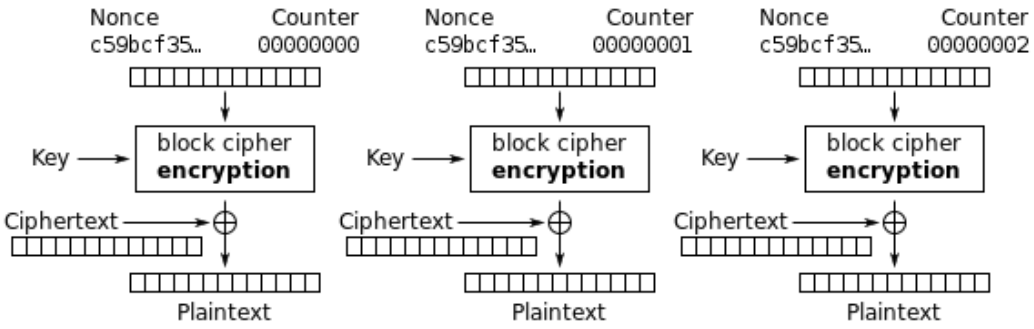


四、SM4-CTR

1. **每个明文分组都与一个经过加密的计数器异或，对每个后续的分组计数器增1。**



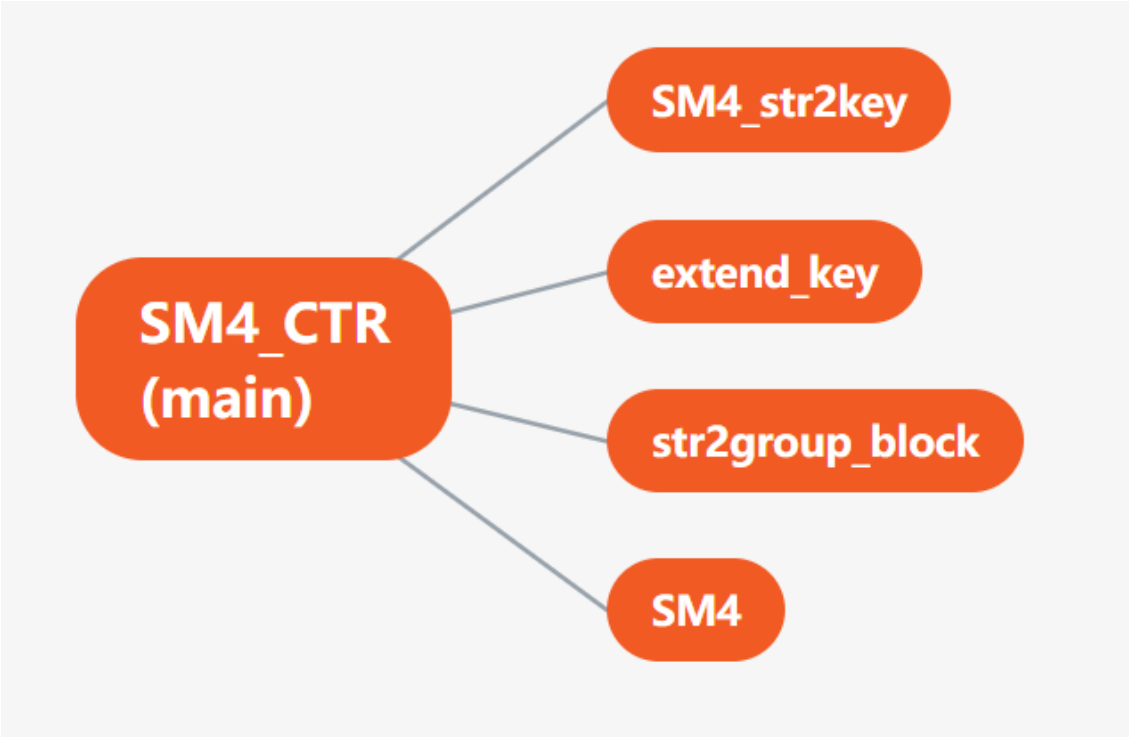
Counter (CTR) mode encryption



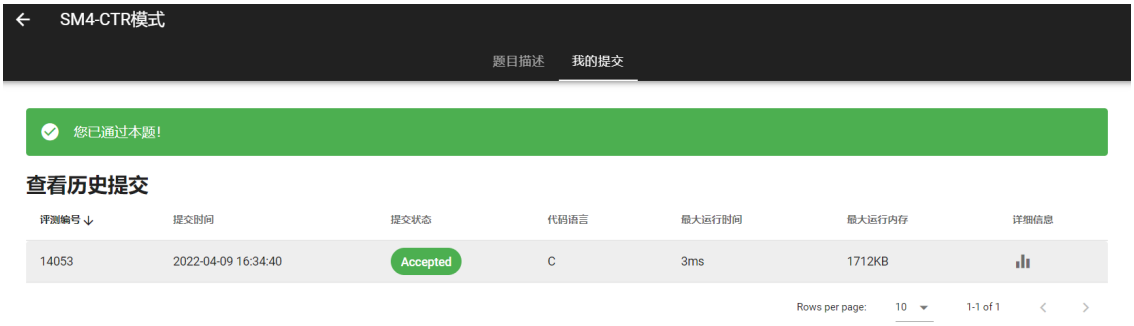
Counter (CTR) mode decryption

特别的，不需要对分组填充，因为参与加密的是计数器，所以不需要对明文填充。其次，涉及到明文密文的部分只在异或操作，然而异或特有的性质使得解密时无需使用SM4的decryption，只需要encryption即可。

2. **函数调用图：**

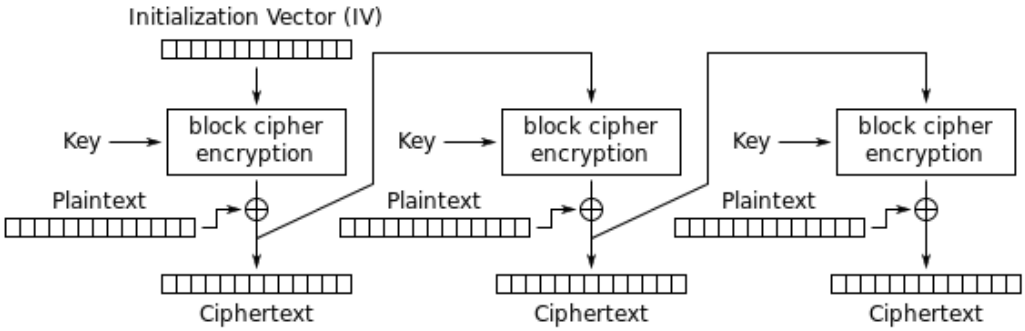


3. **测试样例及结果截图：**

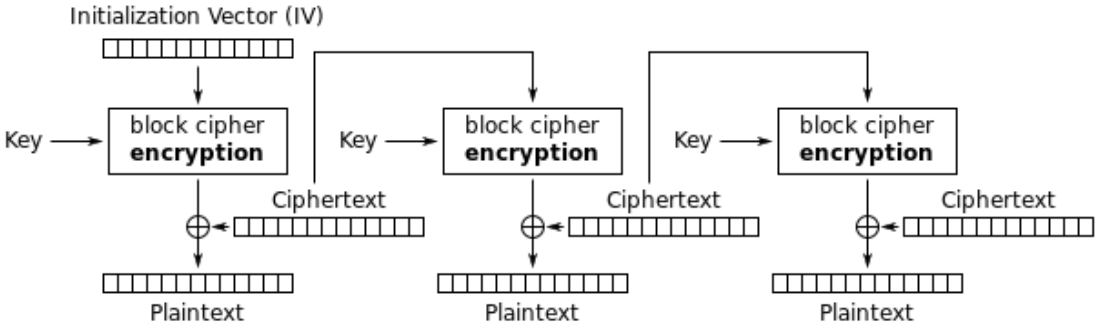


五、SM4-CFB

1. **一次处理输入的s位，上一个密文分组作为加密算法的输入，产生的伪随机数输出与明文异或后作为下一个单元的密文。**



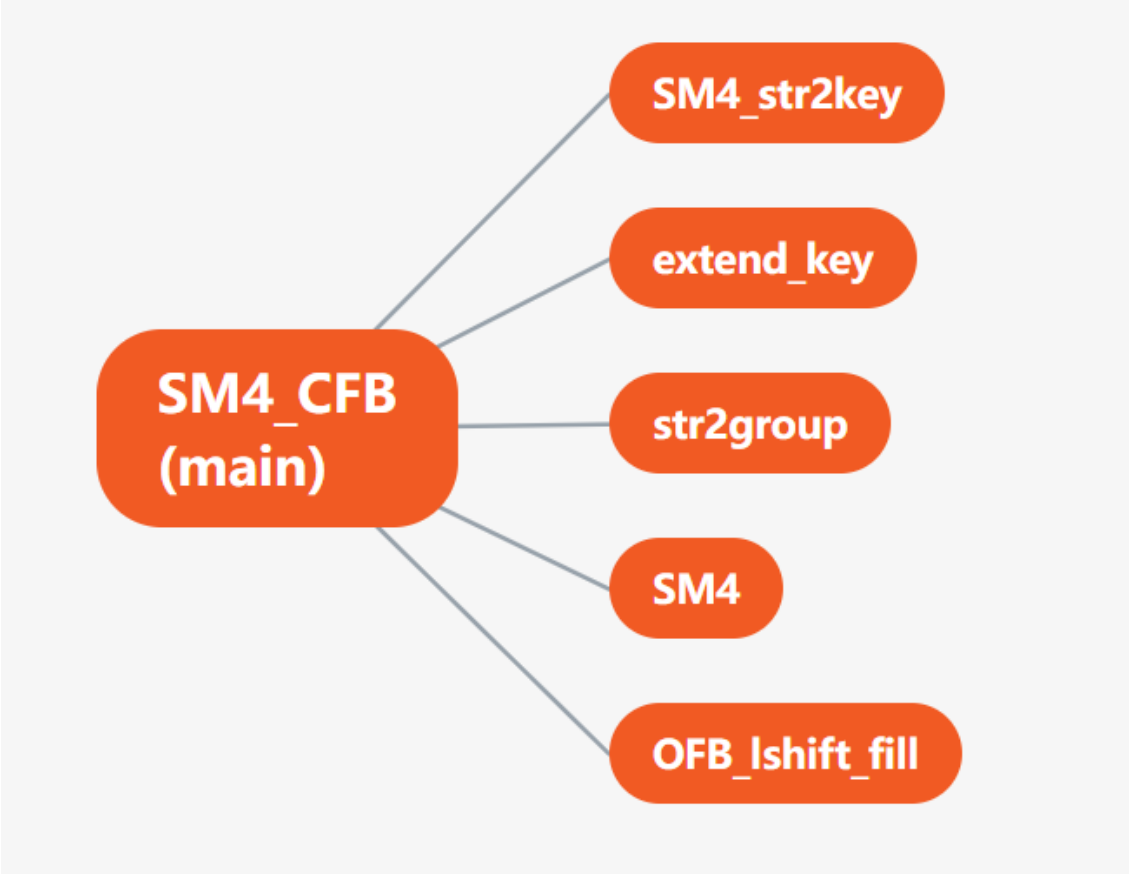
Cipher Feedback (CFB) mode encryption



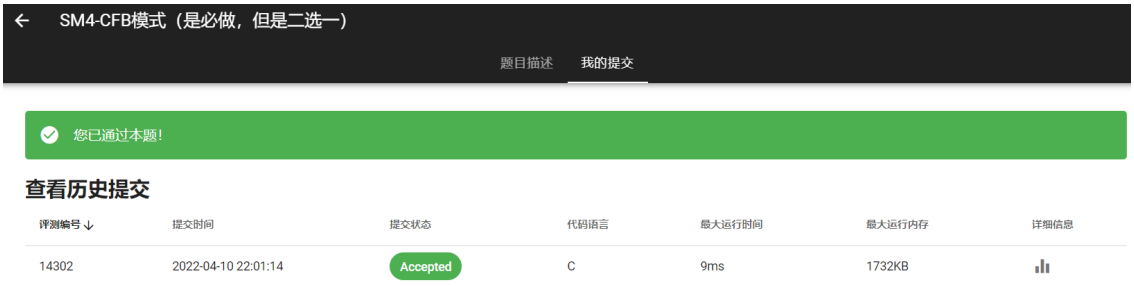
Cipher Feedback (CFB) mode decryption

同理，该工作模式也不需要填充以及加密模式，OFB同理，故之后省略。

2. **函数调用图：**

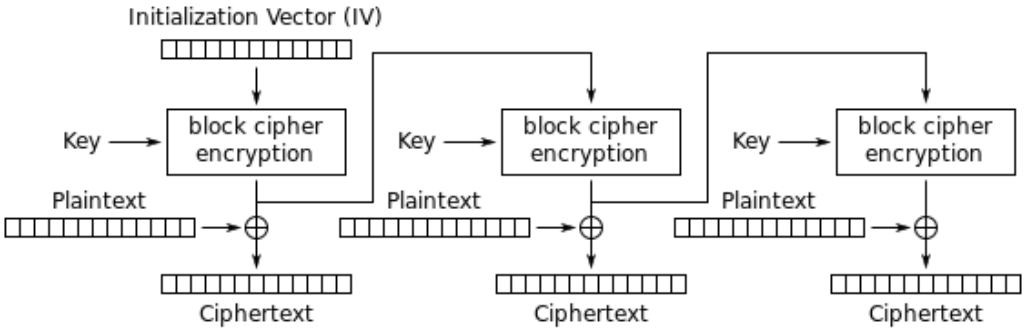


3. **测试样例及结果截图：**

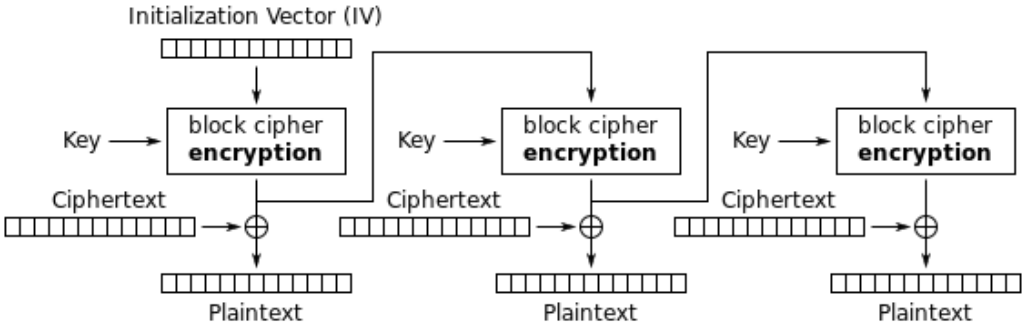


六、SM4-OFB

1. **与CFB类似，只是加密算法的输入是上一次加密的输出，并且使用整个分组。**

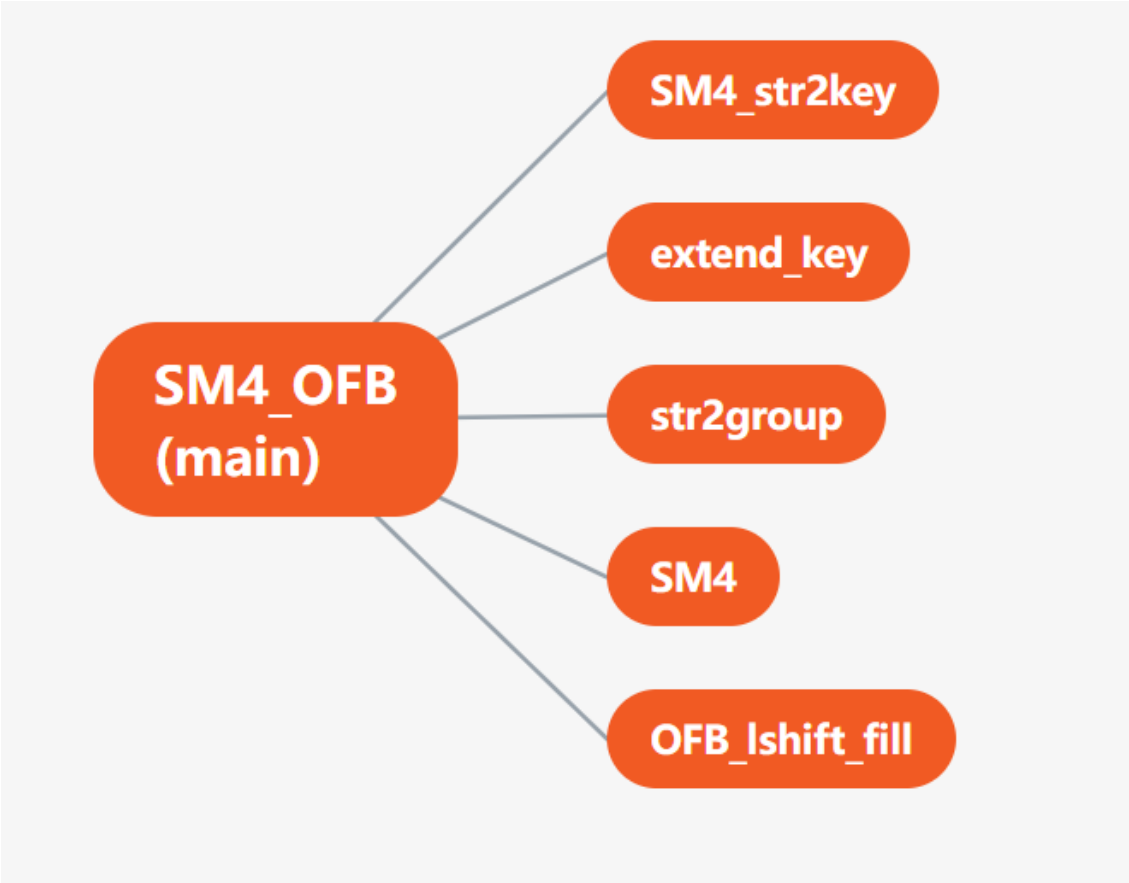


Output Feedback (OFB) mode encryption

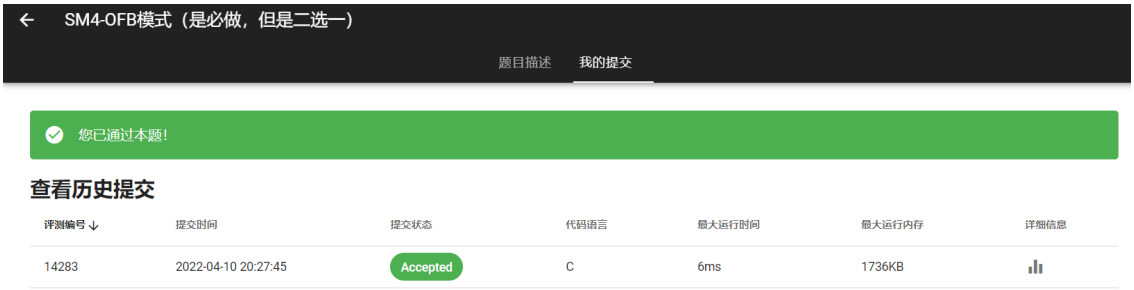


Output Feedback (OFB) mode decryption

2. **函数调用图：**



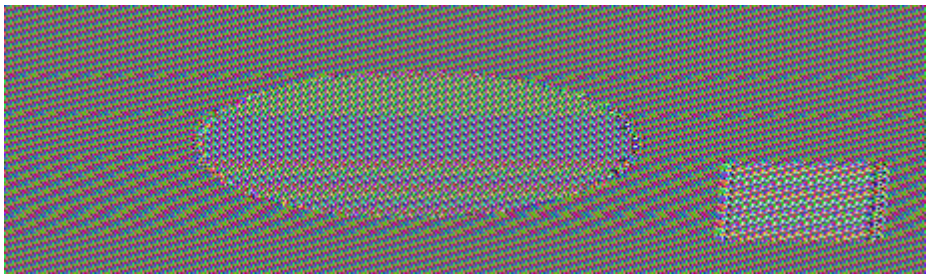
3. **测试样例及结果截图：**



七、讨论与思考

1. 经过分析可知：S盒所占时间约为轮函数的一半。
2. 三者均为对称分组密码，而DES分组长度与密钥长度最短，AES-128与SM4分组长度和密钥长度相同。DES有8个S盒，而AES与SM4的S盒固定。DES循环16轮，AES-128循环10轮，而SM4循环32轮，安全性较高。SM4密码的调度较为简单，解密时具有较好的可逆性。
3. **图片加密：**
    - 原图片：

    - ECB加密图片：



    - CBC加密图片：

由此可知：ECB不涉及反馈，所以加密后的文件依然有一定的位置对应信息，隐藏有部分图片信息，而CBC涉及反馈，前后加密之间有相互关联，因而加密结果的位置信息相对较少，加密效果就更好。

八：收获与感悟

经过本次实验，我深刻了解了五种工作模式的工作原理与方法，进一步加深了SM4的理解。希望之后可以进一步探索相关的知识。