Project 1

1、SM4 算法实现

1.1 算法实现过程

1.1.1 左循环移位

对32位整数进行左循环移位操作，用于 SM4 的线性变换步骤

1.1.2 密钥扩展

将128位（16字节）的密钥扩展为32轮的子密钥，每轮生成一个32位的子密钥，存储在 `round_keys` 向量中。

- 初始密钥由前四个字节组成
- 后续子密钥通过前一轮子密钥左移8位并与 `Fk` 数组中的对应值异或得到

1.1.3 非线性变换

使用 S 盒对每个字节进行替换，实现非线性变换

1.1.4 线性变换

对32位字进行两次左循环移位并异或，实现线性变换

1.1.5 轮函数

结合轮密钥的异或、非线性变换和线性变换，完成单轮加密操作

1.1.6 加密函数

- **密钥扩展**：调用 `key_expansion` 生成32轮子密钥
- **数据分组**：将16字节的明文分为4个32位字
- **轮加密**：对每个32位字执行32轮加密操作，使用相同的轮密钥
- **密文输出**：将加密后的32位字重新组合成16字节的密文

1.1.7 十六进制打印函数

将16字节的数据以十六进制格式打印出来，便于查看明文和密文

1.1.8 主函数

- **初始化**：
  - 使用当前时间作为随机数种子
  - 定义一个固定的16字节密钥

- **加密迭代：**
  - 执行十次加密操作，每次生成随机的16字节明文
  - 对每次加密进行计时，并累加总耗时
  - 打印每次加密的明文和对应的密文
- **结果输出：**计算并输出总加密时间和平均加密时间

1.2 运行结果

该算法实现了 SM4 加密功能，并通过对加密操作进行十次迭代以计算平均时间：

```
第0次加密： 明文： 2432592085da3e14feae43c7e441fa8f
密文： 0cdc3672fd09e13f1910c4b8edc094ff
第1次加密： 明文： 02c32831dd2f25af3dc58661e8bbfa47
密文： 10308bb7f111781db0db1b928c2703ef
第2次加密： 明文： b130729eb16185b1d51f12f299812749
密文： dcef4781371b551e5b9fec07d9646de5
第3次加密： 明文： 12d621ac32dd15f048396fdfc61fd98a
密文： 000687a4432b1f358188008b73001f32
第4次加密： 明文： 47525fe1d051d08056ce67cc8163aa3f
密文： 88ed1d0181b690c1479373afe44fd981
第5次加密： 明文： b08193003da9eeb702de029f1c5a73dd
密文： aebc4e993a65b21016bb4b7d0fae714f
第6次加密： 明文： ee8067106912e8298ba48a7d26504d18
密文： 43d341f7f579b77346b604b727df34a5
第7次加密： 明文： e3acc45486f977ab749d87ca72d490e6
密文： 132eb6f061f67f2e77a040670f78dc2f
第8次加密： 明文： afa2d353050c95537c87c22c10b1d67d
密文： 7b1fd2439b1f061b0fd71f78b54bc1a2
第9次加密： 明文： b43ffdd196367975a55f458751f4f850
密文： 01a20cd1a8420ce7f09facd424c4950c
总加密时间： 0.1863 ms
平均加密时间： 0.01863 ms
```

可见此时的平均加密时间为0.01863 ms

2、SM4 算法优化

2.1 T-table

T-table 是一种预计算表，在优化中，首先需要根据加密算法的具体步骤预先计算出所有可能的中间结果，并将它们存储在 T-table 中，于 SM4 的32轮加密过程，可以预先计算出每一轮可能的中间状态，并将其存储在相应的 T-table 中。此时，算法不再逐轮进行完整的计算，而是通过查表的方式直接获取预计算的结果，这大大减少了每轮加密所需的计算量，实现了算法效率的提高。

2.1.1 初始化 T-table

定义一个二维数组 `T_table` ，预计算所有可能的字节值在 SM4 中的变换结果

- 遍历所有256个可能的字节值，对于每个字节值 `i` ：

- 将其左移24位，形成一个32位的字 `word` （高字节为 `i` ，低三个字节为0）
- 对该字进行非线性变换（S盒替换）
- 对替换后的结果进行线性变换
- 将变换后的32位结果拆分为4个字节，分别存储到 `T_table[i][0]` 到 `T_table[i][3]` 中

2.1.2 轮函数的优化实现

执行单轮加密操作，利用预计算的T-table，减少了每轮加密中需要进行的计算量

- 将输入与轮密钥进行异或，得到中间结果 `output`
- 将 `output` 拆分为4个字节
- 对每个字节，通过查表获取对应的T-table值，并重新组合成一个新的32位字 `transformed`
- 返回 `transformed` 作为本轮的输出

2.1.3 主函数

- 初始化 T-table：调用 `init_T_table()` 进行预计算
- 定义一个固定的128位密钥
- 进行10次加密操作，每次随机生成明文，记录加密时间，并输出明文和密文
- 计算并输出总加密时间和平均加密时间

2.2 运行结果

通过预计算，将每轮加密中的复杂运算转化为简单的查表操作，降低了每轮的计算开销，此时运行代码结果如下：

```
第0次加密： 明文： a9380b43f57c034b4a7b4ae56e71813b
密文： b714eac62222221b8888888864ce8f01
第1次加密： 明文： bae7545ad5725d85c29d45cf1d242de9
密文： fb6d3c4822222240888488886ce6401
第2次加密： 明文： a1634c38ee0dd92a89212bcb251a1f93
密文： 340b93141b234922881688108beaa464
第3次加密： 明文： 2167ba82511a0f7001e4639e37419090
密文： c5a2fb47231b22221688958866a4cece
第4次加密： 明文： 39f0a71f4fa6d9caca03984d45c5f975
密文： a60ca4c34d2349228810888866e96601
第5次加密： 明文： 2b3c5d327afe50eec6dbb11d80131415
密文： db99f980234d721b88888f958fe98f64
第6次加密： 明文： 78c76420a1aecdd01409d89358d9d887
密文： a3720fca2222232216109516ed64ede9
第7次加密： 明文： e14f8690d684e8a7186a77757f038020
密文： 9876a1544d22494988169510ce8f8fa4
第8次加密： 明文： 67d0cb656b8d7be7701b35c6e6be8474
密文： a29fa506222232401095888888fed8b8f
第9次加密： 明文： 4a887cc4ef159e44ff79fccdbc779e58
密文： 315609a02223221b889516888fea8fed
总加密时间： 0.1314 ms
平均加密时间： 0.01314 ms
```