# 作业二：Simple-AES算法实现

### 作业任务

根据教科书附录D的内容，两人一组，学习了解S-AES算法，并使用最擅长的程序语言来编程实现加、解密程序。

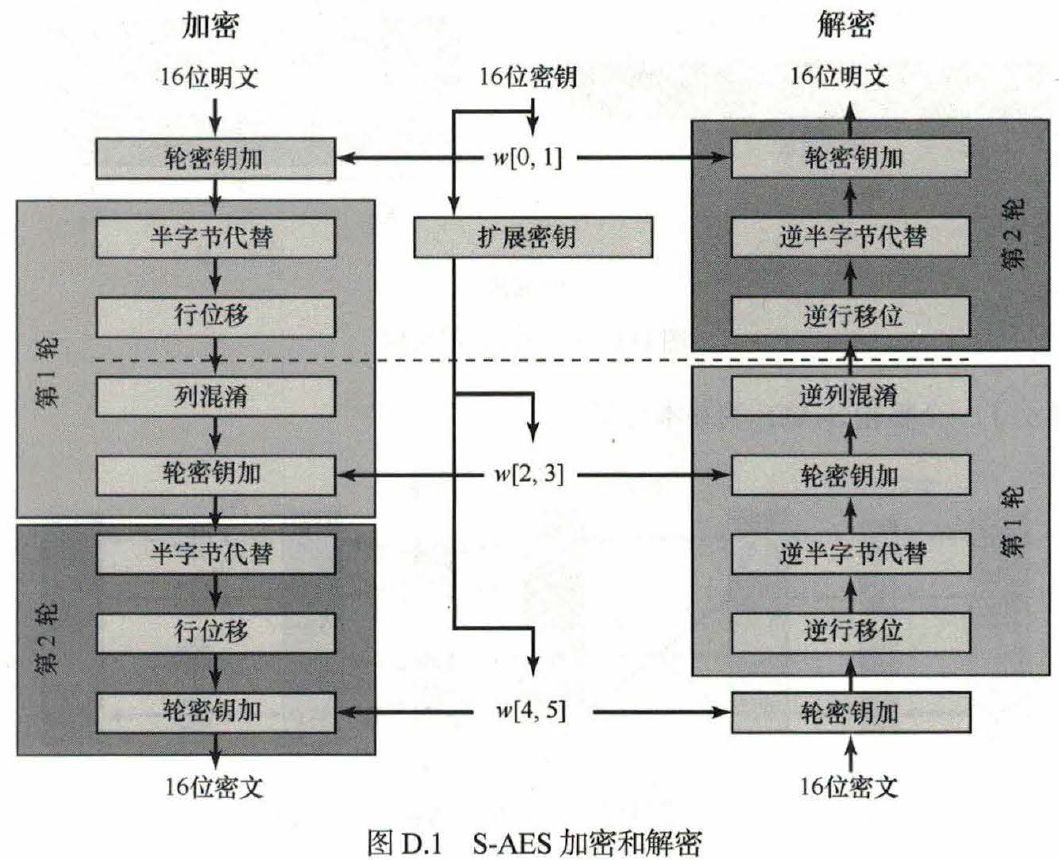


图1 S-AES加密和解密框架

### 软硬件开发环境

系统环境：Windows 11 23H2

编码环境：Visual Studio Code、Dev C++

编程语言：CSS JavaScript HTML C++

### 项目结构

* AESone.html：网页界面，样式设计表
* AES.js：实现部分关卡程序代码
* Double.cpp：实现双重加密的程序代码
* Middle\_attack.cpp：实现中间相遇攻击寻找密钥的程序代码
* Triple.cpp：实现三重加密的程序代码

### 编程和测试

**第一关：**根据S-AES算法编写和调试程序，提供GUI解密支持用户交互。输入是16bit的数据和16bit的密钥，输出是16bit的密文。

表1 S-AES算法基本功能实现结果

|  |  |  |
| --- | --- | --- |
| 明文 | 密钥 | 密文 |
| 1111000011110000 | 1111111100000000 | 0100011000110001 |
| 1111000011111111 | 1111111100000000 | 0011011000110100 |
| 1111000011110000 | 1010101010101010 | 0011000100100011 |
| 0101010101010101 | 1010101010101010 | 1000101010010110 |
| 0101010101010101 | 1111111111111111 | 1111101011101110 |



图2 基础加密功能运行效果

**第二关：**与《我玩原神怎么你了》组进行交叉测试，以证明满足算法标准，选择密文为0110111101101011，密钥为1010011100111011，得到密文结果0000011100111000。两组同学编写的程序对明文P进行加密得到相同的密文C，说明代码符合S-AES算法标准且具有通用性。



图3 《我玩原神怎么你了》组测试结果



图4 本组测试结果

**第三关：**扩展功能。考虑到向实用性扩展，加密算法的数据输入可以是ASII编码字符串(分组为2 Bytes)，对应地输出也可以是ACII字符串(很可能是乱码)。第三关选取字符串”hello,world”进行测试，输出结果为乱码。



图5 第三关测试结果

**第四关：多重加密**

4.1 双重加密

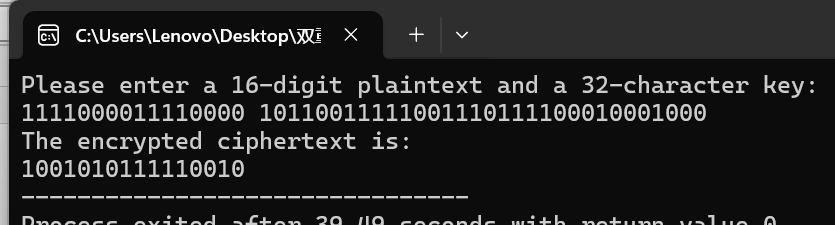
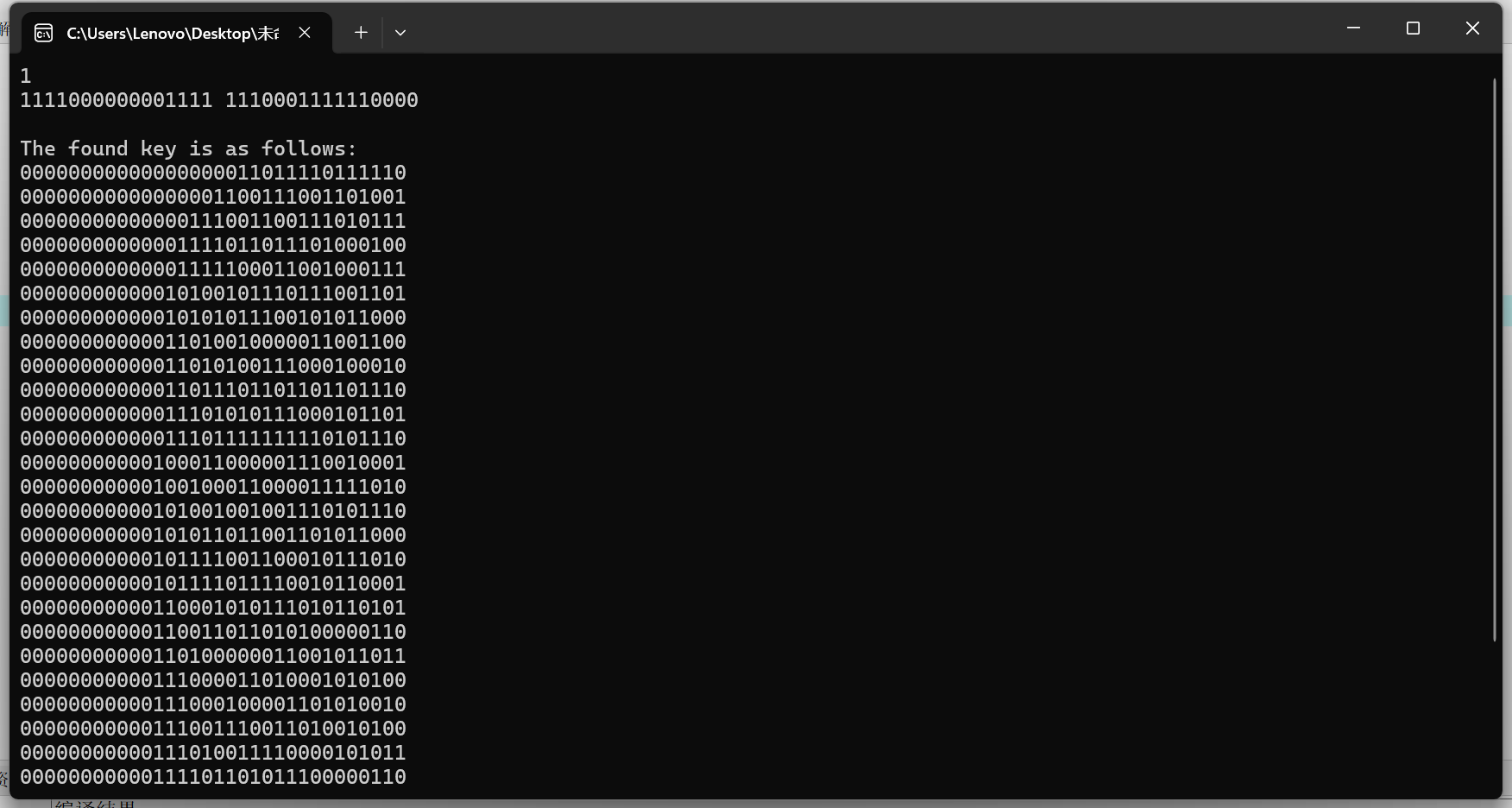
将S-AES算法通过双重加密进行扩展，分组长度仍然是16 bits，但密钥长度为32 bits。

图6 第四关测试结果

4.2 中间相遇攻击

假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用中间相遇攻击的方法找到正确的密钥Key(K1+K2)。

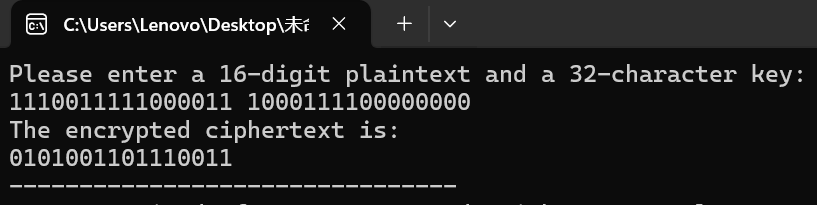
图7 第四关测试结果

输入明密文对的数量以及相应的明密文对，程序会列出所有符合条件的密钥解。

4.3 三重加密

将S-AES算法通过三重加密进行扩展

按照32 bits密钥Key(K1+K2)的模式进行三重加密解密

图8 第四关测试结果

**第五关：**工作模式

基于S-AES算法，使用密码分组链(CBC)模式对较长的明文消息进行加密。注意初始向量(16 bits) 的生成，并需要加解密双方共享。

在CBC模式下进行加密，并尝试对密文分组进行替换或修改，然后进行解密，请对比篡改密文前后的解密结果。

****图9 第五关测试结果

最终得出结论，篡改密文前能成功解密得到原始明文，但篡改后无法正确解密，得到解密结果和明文完全不同。