
中国科学技术大学

本科学位论文



AES 加密，解密工具

作者姓名：	吴立凡
学科专业：	网络空间安全
导师姓名：	李卫海
完成时间：	2020 年 5 月

University of Science and Technology of China

A dissertation for bachelor's degree



AES encryption and decryption tools

Author :	<u>Lifan Wu</u>
Speciality :	<u>Cyberspace Security</u>
Supervisor :	<u>Weihai Li</u>
Finished Time :	<u>May, 2020</u>

致谢

感谢 CSDN 的前辈们，他们的论文博客为本次实验提供了丰富的经验与建议。本次实验的结果验证也使用了 CSDN 上的工具。如果不是他们的帮助，这篇文章的完成也会变得无比艰难。

另外也要感谢所有教过我的老师们，他们所教授的许多内容都成为了我实践的坚实基础。在此特别感谢李卫海老师，他所教授的密码学导论为这篇论文提供了莫大的帮助。

目录

摘要	5
关键词	6
Abstract	6
Keywords	6
正文:	7
题目要求	7
效果	7
代码文件简述	7
AES 算法流程	8
AES 输入分组	9
AES 算法子密钥生成	10
AES 的轮函数	11
AES 解密	11
加密在题目 1 中使用的书籍	12
对密文进行统计分析	13

摘要

随着密码破译技术和电子计算机的发展, DES 安全性上的不足越发明显。新的分组加密算法 AES 出现并取而代之, 成为了被美国国家标准技术协会 (NIST) 选中的全新密码方案, 在安全性、算法复杂度等方面表现良好, 被广泛应用于金融、电信、政府数据等领域的数据加密中。本次研究使用 c 语言实现了整个密码方案, 代码主要由子密钥生成、加密、解密工具等部分组成, 并对一整本《圣经》进行了加密, 对加密后的结果做了统计分析。

关键词

AES 有限域上的分组密码 轮函数 频率统计 子密钥生成 信息安全

Abstract

With the development of password breaking technology and electronic computers, the security deficiencies of DES have become more and more obvious. The new block encryption algorithm AES appeared and replaced it. It has become a new encryption scheme selected by the National Institute of Standards and Technology (NIST). It performs well in terms of security and algorithm complexity, and is widely used in finance, telecommunications, government data, etc. This study uses the C language to implement the entire cryptographic scheme. The code is mainly composed of sub-key generation, encryption, and decryption tools. The entire "Bible" is encrypted, and the encrypted results are statistically analyzed.

Keywords

AES block cipher on finite field round function frequency statistics subkey generation information security

正文：

题目要求

课程实践

63 / 63

题目2: AES加密、解密工具

- AES密码需采用课堂中介绍的快速查表方式实现。其中所查各表可在给定密钥后预先算好
- 使用公开的AES加密数据来测试你的加密、解密工具是否正确
- 加密你在题目1使用的书籍，统计密文的分布，结果是否均匀？
 - 注意到 $256 \times 256 \times 256 = 16,777,216$ 太大，以1个字节为1个字符显然比较困难。请考虑如何处理比较合理？

效果








本代码实现了 AES 算法经行加密、解密，先上效果图：

```
Microsoft Visual Studio 调试控制台
key:abcdefghijklmnop
plaintext:1bcdefghijklmnop
cipher:
25 63 0e 9e ef e5 e2 8f 45 c2 29 1e d1 3c 3e 04
decryption:1bcdefghijklmnop
```








经过公开的 AES 加密工具可以验证，我写的 AES 算法是没有问题的。

代码文件简述

本程序前一部分“AES 加密、解密工具”源代码存放在“AES 加密、解密工具”文件夹里面，由以下几个文件组成，decrypt.h 实现了 AES 算法的加密功能，encrypt.h 实现了 AES 中的解密部分，init_key.h 实现了密钥矩阵的初始化，key.txt 保存了密钥，main.cpp 为程序的主部分，table.h 存放了 AES 算法所使用到的一些常数矩阵。还有 plaintext.txt 存放第二部分要加密的文件《圣经》，cipher.txt 存放加密后的结果。由于我实在 linux 上完成这个实验的，所以有一个 linux makefile，AES 也是 main.cpp 经编译后的 elf 文件

名称	修改日期	类型	大小
 AES	2021/5/18 17:03	文件	22 KB
 cipher.txt	2021/5/18 17:03	文本文档	4,363 KB
 decode.txt	2021/5/18 17:03	文本文档	4,363 KB
 decrypt.h	2021/4/5 21:30	H 文件	2 KB
 encrypt.h	2021/5/18 17:10	H 文件	3 KB
 init_key.h	2021/4/5 21:13	H 文件	1 KB
 key.txt	2021/4/5 17:11	文本文档	1 KB
 main.cpp	2021/5/18 17:10	CPP 文件	2 KB
 makefile	2021/5/18 16:24	文件	1 KB
 plaintext.txt	2021/5/17 21:55	文本文档	4,363 KB
 table.h	2021/4/5 16:56	H 文件	3 KB

至于实验后一部分的内容“统计密文比特频率”所使用到的代码，放在了“统计频数”文件夹中，这一部分文件是由实验 1 中的文件经过一定的改写而来的，主要是统计的分类由 26 个字母改为 256 个比特串，alpha_frequency.py 则用于图像的可视化处理。

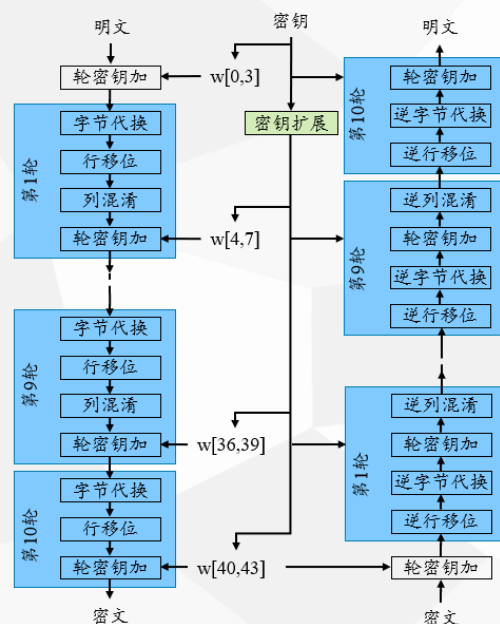
 a.out	2021/5/18 20:36	Wireshark captu...	17 KB
 alpha_frequency.py	2021/5/18 20:37	Python File	2 KB
 alpha_frequency.txt	2021/5/18 20:36	文本文档	2 KB
 double_alpha_frequency.txt	2021/5/26 15:55	文本文档	193 KB
 frequency.h	2021/5/18 20:36	H 文件	2 KB
 main.cpp	2021/5/18 20:36	CPP 文件	2 KB
 plaintext.txt	2021/5/18 17:03	文本文档	4,363 KB

AES 算法流程

AES 算法为有限域上的分组密码，一次加密将一个长度为 128b (16B, 4W) 的明文和长度为 128b (16B, 4W) 或 192b (24B, 6W) 或 256b (32B, 8W) 的密钥可逆映射到 128b (16B, 4W) 的密文空间上。AES 加密流程图如图所示：

● AES结构

1. 输入分组以正方形矩阵State描述
2. 密钥扩展为矩阵
3. 进行9/11/13轮迭代
 - ① 字节代换
 - ② 行移位
 - ③ 列混淆
 - ④ 轮密钥加
4. 最后一个不完整轮
5. 矩阵State转换为输出分组

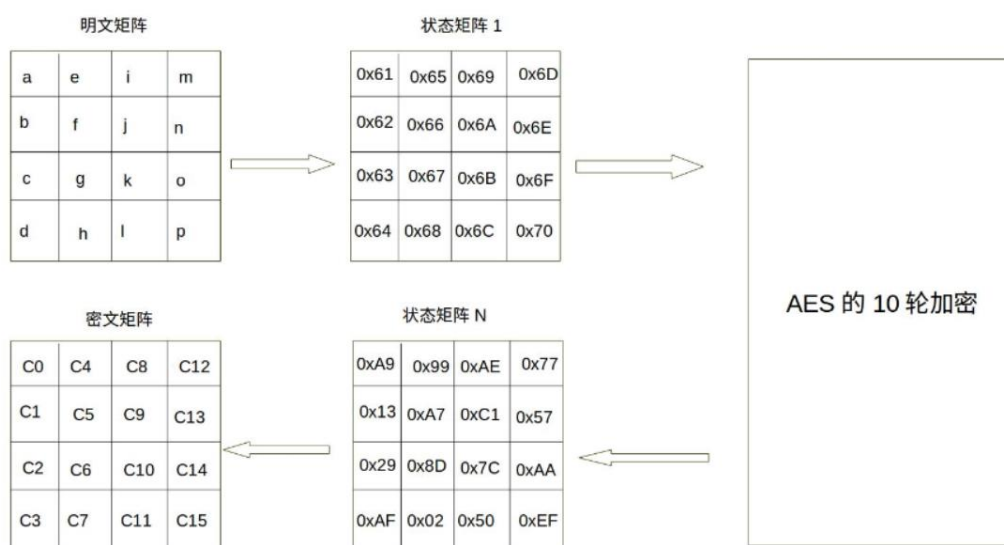


(图片来源: Crypt04-有限域上的分组密码. pptx 李卫海老师)

AES 结构分为输入描述、子密钥生成、轮函数等部分组成, 接下来对这些部分一一介绍:

AES 输入分组

AES 算法将输入用一个正方形矩阵来描述, 如图所示:



(图片来源: https://blog.csdn.net/qq_28205153/article/details/55798628)

对应到程序, 程序里的明文分组、密钥分组、密文分组均以 4*4 矩阵表示, 以一

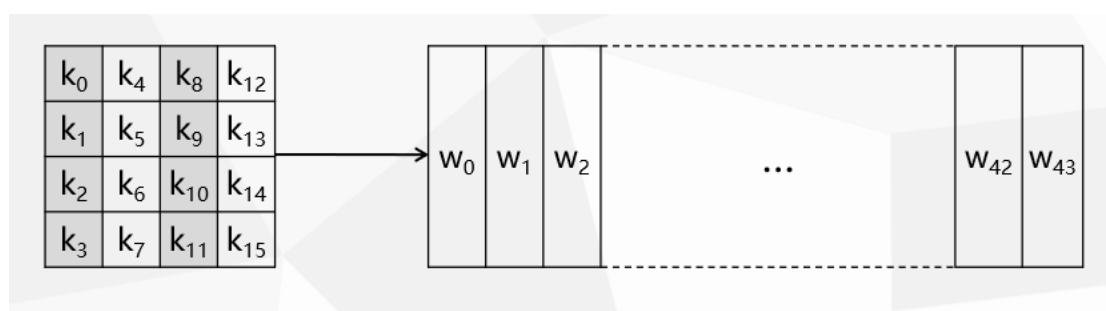
个字节为单位，如图：

```
unsigned char key[4][4] = { 0 };
unsigned char sub_key[4][44] = { 0 };
unsigned char plaintext[4][4] = { 0 };
unsigned char cipher[4][4] = { 0 };
unsigned char dec[4][4];
for (char i = 0; i < 16; i++) key[0][i] = 'a' + i;
```

AES 算法子密钥生成

这一部分的代码实现放在了 init_key.h 头文件里。

AES 通过密钥扩展算法，将一个 16 字节的原密钥扩展为 44*4 字节的密钥序列数组，每次轮密钥加运算使用 4 个字节，一共要加 11 轮



（图片来源：Crypt04-有限域上的分组密码.pptx 李卫海老师）

密钥扩展时，首先将 4 字的原始密钥放在前四字的密钥矩阵里，接着使用如下的递归方式扩展 40 个新列：

如果 4 不整除 i ： $W[i] = W[i-4] \oplus W[i-1]$

如果 4 整除 i ： $W[i] = W[i-4] \oplus g(W[i-1])$

其中， g 是一个复杂函数。

g 分三步：1. 字循环：四个字节循环左移一个字节

2. 字节代换：用 S 盒对每个字节进行代换

3. 与轮常量 $Rcon[j]$ 异或

轮常量可以通过查表得到，已经存放在 table.h 文件中：

```
/*轮常量表 The key schedule rcon table*/
static const unsigned char Rcon[10] = {
    0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1b, 0x36 };

```

AES 的轮函数

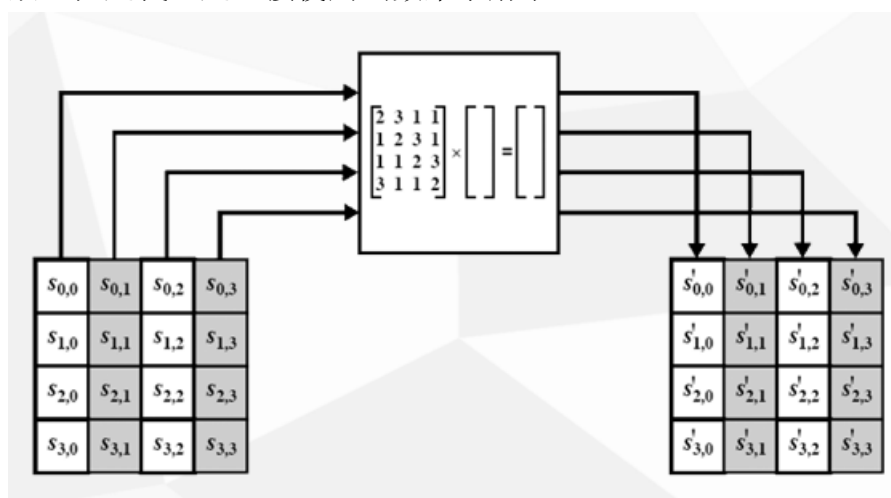
（本部分具代码在 encrypt.h 头文件中得到了实现）

AES 算法共有 10 轮加密，首先对明文用 $w[0, 3]$ 经行轮密钥加，再经过 9 轮的字节代换、行移位、列混淆和轮密钥加，最后第 10 轮只需要经行字节代换、行移位和轮密钥加。

字节代换：字节代换是简单的查表操作，利用一个 16×16 大小的 sbox 进行单字节代换，使用字节的高 4 位作为行值，低 4 位作为列值，取 S 盒中对应行列的元素输出，可以通过 table.h 头文件里的 sbox 数组来实现。

行移位：对状态矩阵进行操作，第一行不变，第二行循环左移一个字节，第三行循环左移两个字节，第四行循环左移三个字节。

列混淆：列混淆使用了 $GF(2^8)$ 上运算，对每列独立进行操作，使用一个列混淆矩阵与每一列相乘。这一部分实际上也可以用查表的方法来替代，会更加简单高效，但是我还是直接使用函数来求解了。



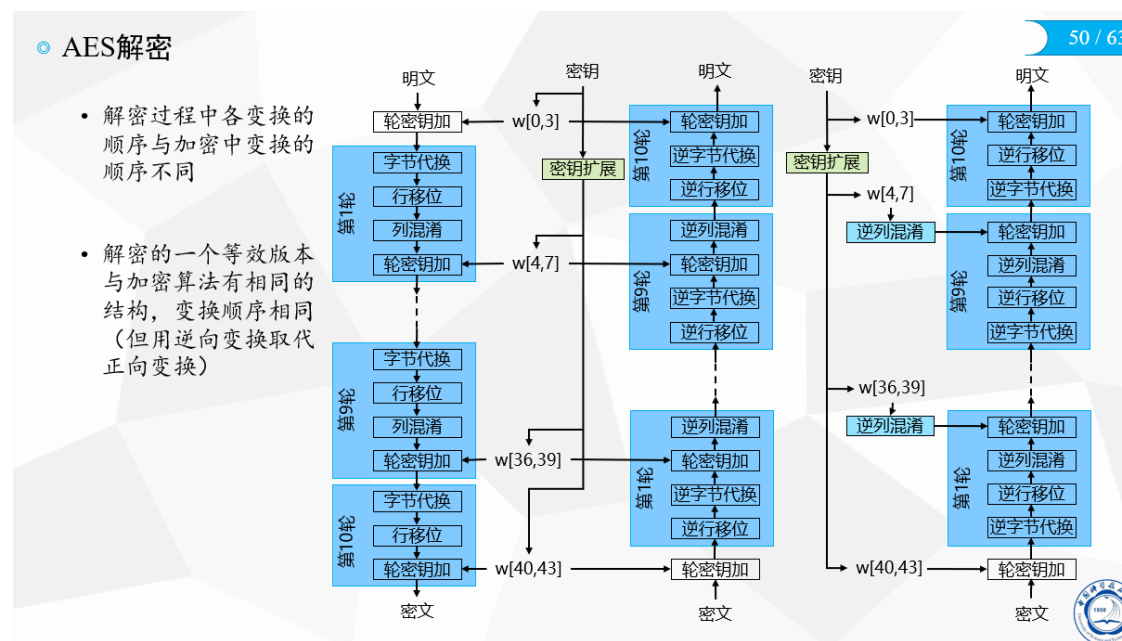
（图片来源：Crypt04-有限域上的分组密码.pptx 李卫海老师）

轮密钥加：直接将状态矩阵与对应的轮密钥相加，这里是在 $GF(2^8)$ 中的加法。

AES 解密

（解密在 decrypt.h 中实现）

AES 解密与加密算法有很多相同的部分，但是仅仅将变换改成加密的逆变换是不够的。AES 解密过程中解密过程中各变换的顺序与加密中变换的顺序不同。具体流程图如下：



（图片来源：Crypt04-有限域上的分组密码.pptx 李卫海老师）

加密《圣经》

对整个文件经行加密并解密。将加密后的结果保存在 cipher.txt 文件里，将解密后的结果保存在 decode.txt 里。可以看出，decode.txt 与 plaintext.txt 基本上完全一致，只有在 decode.txt 的最后出现了一些乱码，这是由于解密是以 4*4bytes 的分组进行的，解密后的文件长度也必须是 16 的整数倍。不过这一点乱码并不影响接下来的统计分析，所以我就不对其进行处理了。

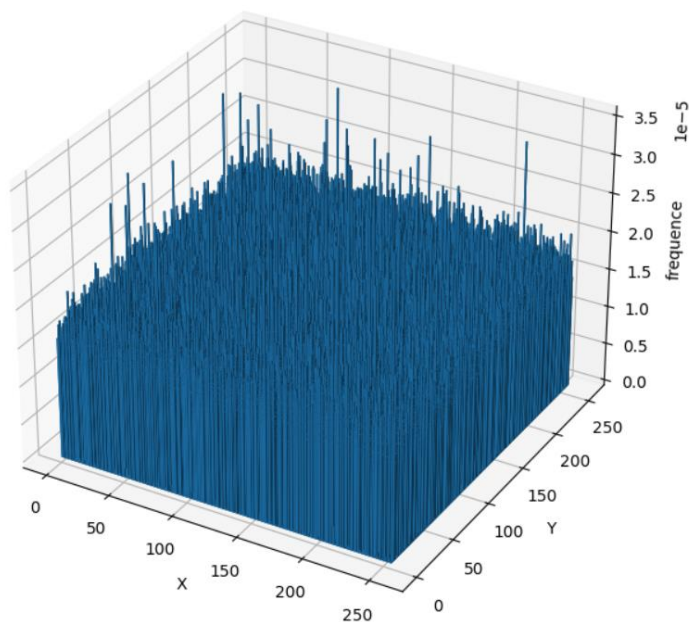
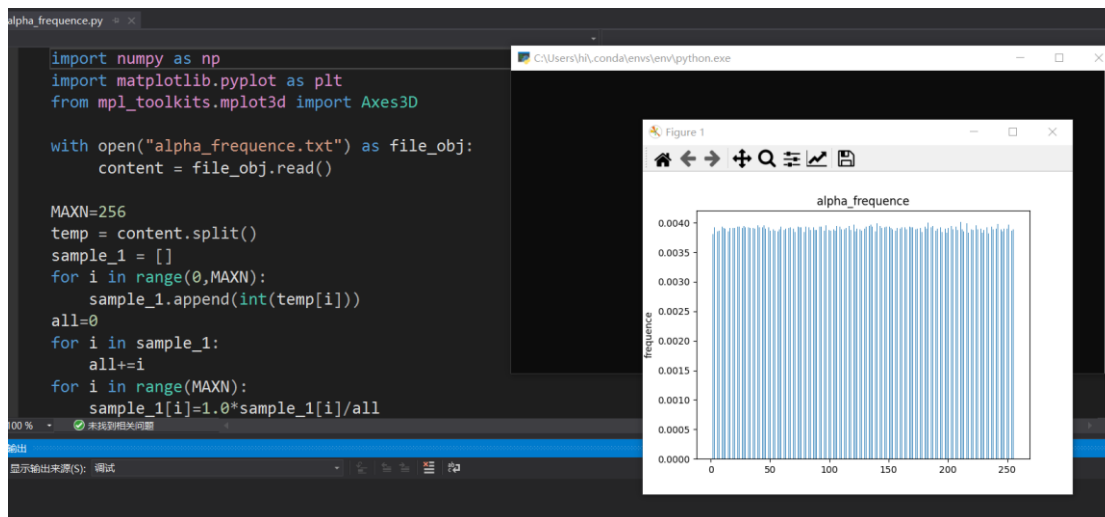
```

AES [WSL: UBUNTU]
├─ AES
├─ cipher.txt
├─ decode.txt
├─ decrypt.h
├─ encrypt.h
├─ init_key.h
├─ key.txt
├─ main.cpp
├─ makefile
├─ plaintext.txt
└─ table.h

main.cpp
31087 Rev 22:5 And there will be no more night; and they have no need of a light of
31088 Rev 22:6 And he said to me, These words are certain and true: and the Lord, th
31089 Rev 22:7 See, I come quickly. A blessing on him who keeps the words of this bo
31090 Rev 22:8 And I, John, am he who saw these things and to whose ears they came.
31091 Rev 22:9 And he said to me, See you do it not; I am a brother-servant with you
31092 Rev 22:10 And he said to me, Let not the words of this prophet's book be kept
31093 Rev 22:11 Let the evil man go on in his evil: and let the unclean be still uncl
31094 Rev 22:12 See, I come quickly; and my reward is with me, to give to every man
31095 Rev 22:13 I am the First and the Last, the start and the end.
31096 Rev 22:14 A blessing on those whose robes are washed, so that they may have a
31097 Rev 22:15 Outside are the dogs, and those who make use of evil powers, those w
31098 Rev 22:16 I, Jesus, have sent my angel to give witness to you of these things
31099 Rev 22:17 And the Spirit and the bride say, Come. And let him who gives ear, s
31100 Rev 22:18 For I say to every man to whose ears have come the words of this pro
31101 Rev 22:19 And if any man takes away from the words of this book, God will take
31102 Rev 22:20 He who gives witness to these things says, Truly, I come quickly. Ev
31103 Rev 22:21 The grace of the Lord Jesus be with the saints. So be it.
31104 I0Bq+4
  
```

对密文进行统计分析

对于单字节和双字节,可以参考之前的字母统计分析,只需将代码稍微修改一下,将 26 个英文字母的分类改为 256 个字节串的分类。统计效果如下:



可以看出, AES 算法的算法能够很好的消除分布中的统计特征,单字符、双字符出现次数的方差较小。