DES 算法报告

- 米家龙
- 18342075
- 数据科学与计算机学院

目录

算法原理总综述

> 本次采用的是 des-ecb 加密算法，因此是对一个8字节的块进行加密，并且需要进行填充

加密

填充

填充采用的是 PKCS#5 规范进行字节填充：

- 当原始明文最后分组不足8字节，则填满至8字节，填充的值为**需要填充的字节牧户**
- 如果原始明文分组完全，则需要额外增加一个分组，每个字节的值都是0x08

子密钥生成

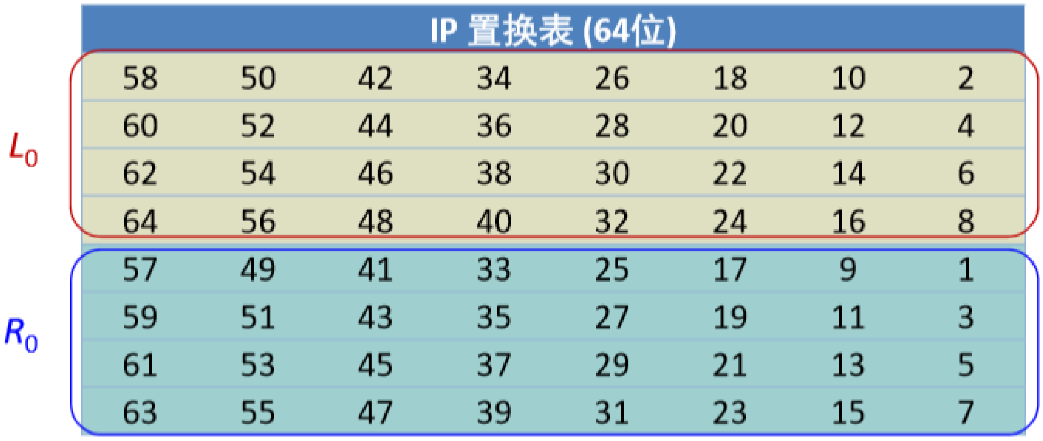
1. 获取给定的64位密钥 K
2. 使用**PC-1置换表**进行置换，得到56位的C_0D_0 ， C_0和D_0分别由置换结果的前28位和后28位组成
3. 对一下操作进行16次循环，生成子密钥$K_1 - K_{16}$：
    1. 计算子C_iD_i：$C_i = LS_i(C_{i-1})$， $D_i = LS_i(D_{i-1})$， LS 代表循环左移，当$i = 1, 2, 9, 16$时，**循环左移一位**；否则**循环左移两位**
    2. 对C_iD_i进行**PC-2置换**，压缩成48位，得到对应的子密钥K_i
    3. $i = i + 1$

块加密

> 基于上述分组和填充后，对每个8字节的块进行块加密

初始置换

基于下图对8字节的块（共64位）进行初始置换，途中置换表中数字对应的原始64位的下标编号序列



IP 置换表 (64位)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

> 由于该下标编号序列是1到64，因此在直接使用时需要-1，用于匹配数组的下标

迭代

根据初始置换，得到了L_0R_0，以该数组为基础，进行16次迭代，下面列表表示一次迭代：

- $L_i = R_{i-1}$
- $R_i = L_{i-1} \bigoplus f(R_{i-1}, Ki)$
- $i = i + 1$
- 其中f是轮函数，输出一个32位数组；$\bigoplus$是32位二进制串按位**异或**

轮函数

> 轮函数接受32位的输入，并且返回一个32位的输出

具体步骤如下：

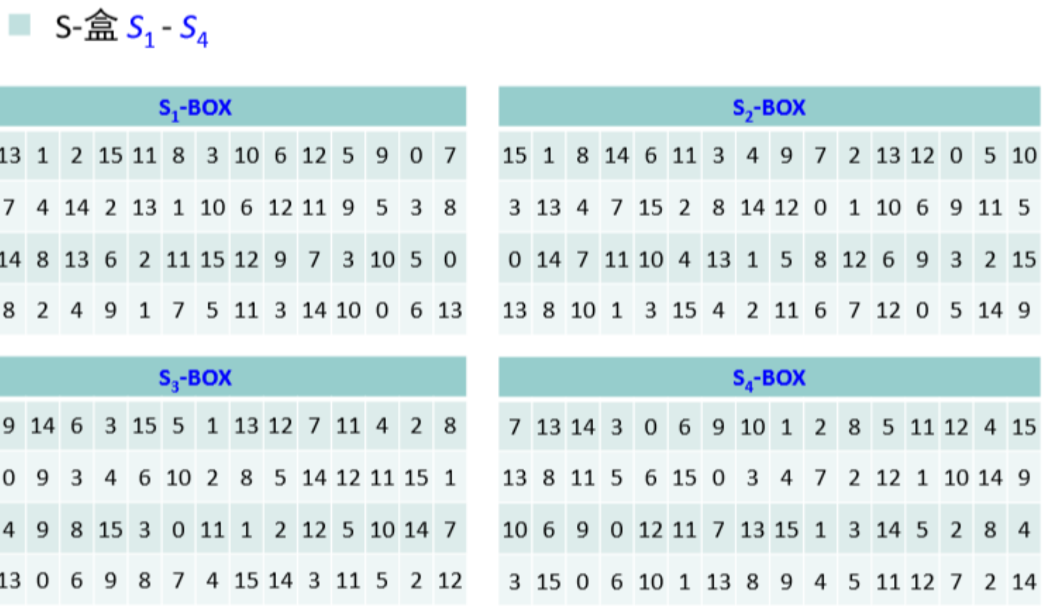
1. 将长度为32位的串 R_{i-1} 作 **E-扩展**，得到一个48位的串 $E(R_{i-1})$
2. 将 $E(R_{i-1})$ 和长度为48位的子密钥 K_i 作48位二进制串**按位异或**运算， K_i 由密钥 K 生成
3. 将上一步得到的结果平均分成8个分组，每个分组长度6位。各个分组分别经过8个不同的**S-盒**进行6-4 转换，得到8个长度分别为4位的分组，具体转换操作如下：

   - S-盒是一类选择函数，用于二进制**6-4转换**。Feistel轮函数使用8个S-盒 $S_1,\cdots,S_8$ ，每个S-盒是一个4行(编号十进制数 0-3)、16列(编号十进制数 0-15) 的二维表，表中每个元素是一个十进制数，取值在 0-15 之间，用于表示一个4位二进制数。
   - 假设Si 的6位二进制输入为 $b_1b_2b_3b_4b_5b_6$ ，则由 $n=(b_1b_6)_{10}$ 确定行号，由 $m=(b_2b_3b_4b_5)_{10}$ 确定列号，$S_i[n,m]$ 元素的值 的二进制形式即为所要的 S_i 的输出。

4. 将第3步得到的分组结果顺序连接得到长度为32位的串

5. 将上一步的结果经过**P-置换**，得到的结果作为轮函数 $f(R_{i-1},K_i)$ 的最终32位输出。

S-盒如图：



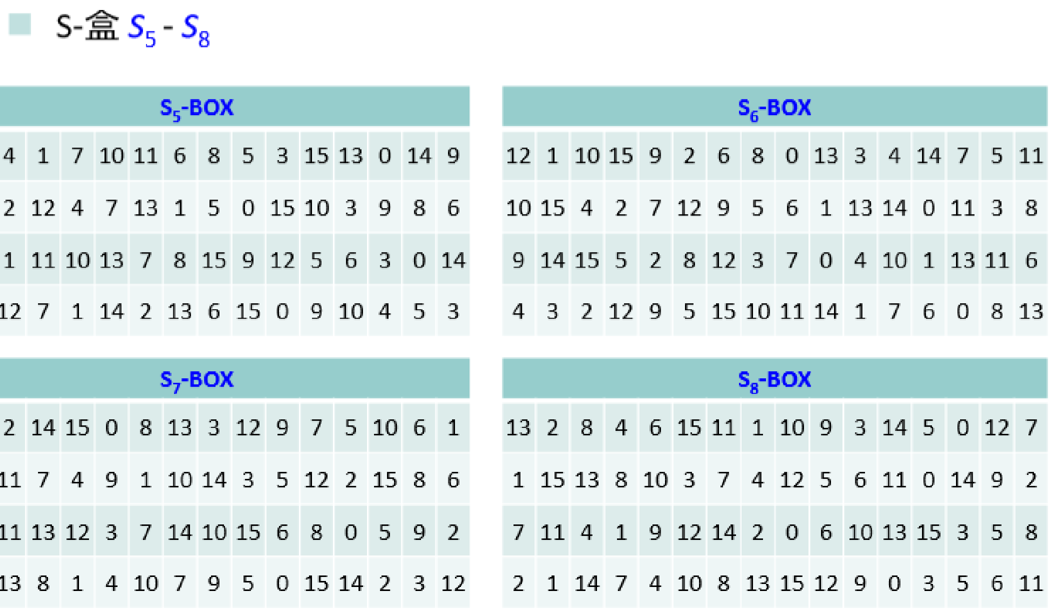
S-盒 S_1 - S_4

S_1-BOX															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2-BOX															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3-BOX															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4-BOX															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14



S-盒 S_5 - S_8

S_5-BOX															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6-BOX															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7-BOX															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

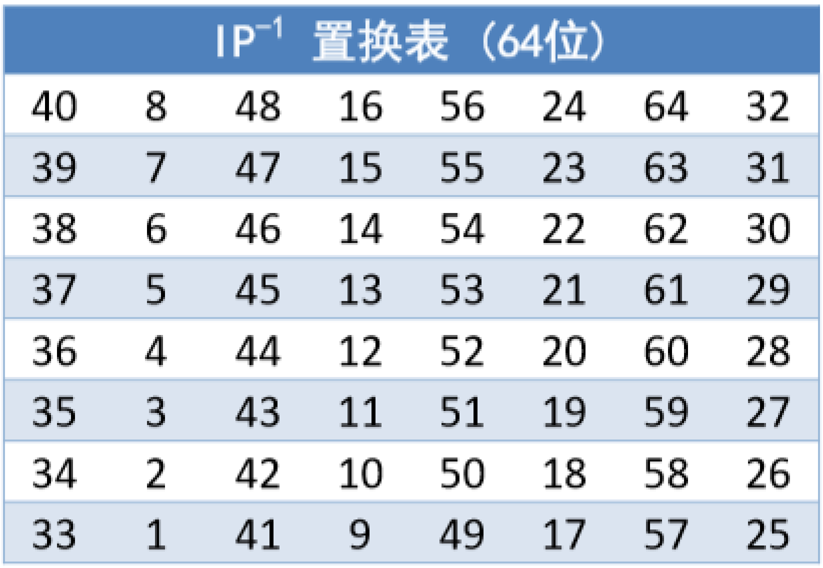
S_8-BOX															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

交换置换

将迭代结果得到的 $L_{16}R_{16}$ 进行交换，即得到结果 $R_{16}L_{16}$

IP 逆置换

根据 IP 逆置换表（由 IP 置换表变换而来），进行置换，得到加密结果，逆置换表如下：



IP^{-1} 置换表（64位）

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

解密

> 解密没有补全，其余基本和加密一样，剩下的唯一区别是：
>
> 使用轮函数进行迭代时，是倒序使用子密钥，即从 K_{16} 到 K_1 进行引用

总体架构

- 主函数：
  - 获取密钥，并且生成子密钥
  - 根据参数加载功能：
    - 加密功能（需要设置一个 flag 判断是否已经补全）：
      1. 以8字节为单位，进行文件块读取，如果需要补全，则 `flag = true`
      2. 对上面获取的块进行加密：
         1. 初始置换
         2. 使用轮函数进行16次迭代
         3. 交换置换
         4. IP逆置换
         5. 输出
      3. 如果 `flag == false` ，那么需要进行新增一个空块，进行补全，并按照第2步进行加密，输出；反之则不用
    - 解密功能：
      1. 以8字节为单位，进行文件块读取
      2. 对上面读取的块进行解密：
         1. 初始置换
         2. 使用轮函数进行16次迭代
         3. 交换置换
         4. IP逆置换
         5. 判断填充用于确定输出

数据结构设计

相关数据类型定义如下：

```
#define BLOCK64 64         // 01位块长度
#define BLOCK8 9           // 8字节明文块长度，由于字符串限制，必须+1
#define EEXTAND 48         // E-拓展串
#define SUBKEYLEN 48       // 子密钥长度
#define SUBKEYNUM 16       // 子密钥数量
#define KEYLEN 64          // 密钥长度
#define NOCHECKDIGITLEN 56 // 非校验位长度

typedef bool des1_t;
typedef unsigned char des8_t;

des8_t block8[BLOCK8];           // 明文
des8_t encodedBlock8[BLOCK8];    // 加密后的明文
des1_t block64[BLOCK64];         // 二进制明文
des1_t encodedBlock64[BLOCK64];  // 加密后的二进制明文
des1_t encodingBlock64[BLOCK64]; // 加密中的二进制明文
des1_t decodedBlock64[BLOCK64];  // 解密后的二进制明文
des8_t decodedBlock8[BLOCK8];    // 解密后的明文
des1_t decodingBlock64[BLOCK64]; // 解密中的二进制明文

char InitKey[KEYLEN / 4 + 1];        // 16进制的输入
des1_t Key[BLOCK64];                 // 密钥
des1_t Subkey[SUBKEYNUM][SUBKEYLEN]; // 子密钥

FILE *readFile; // 读取的文件
```

模块分解

具体函数如下：

```
/**
 * 通过密钥生成子密钥，总共生成16个
 * @param K des1_t* 密钥
*/
void getSubkey(des1_t *K);

/**
 * 8字节 转换成 64位
 * @param from des8_t* 源数组
 * @param to des1_t* 目标数组
*/
void block8ToBlock64(des8_t *from, des1_t *to);

/**
 * 64位 转换为 8字节
 * @param from des1_t* 源数组
 * @param to des8_t* 目标数组
*/
void block64ToBlock8(des1_t *from, des8_t *to);

```

```c
/**
 * 通过初始获取的密钥进行转换
*/
void getKey();

/**
 * 轮函数
 * @param Ri des1_t*
 * @param iterationNum int 迭代次数
 * @return 一个32位数组指针
*/
des1_t *Feistel(des1_t *Ri, int iteraionNum);

/**
 * 块加密
*/
void encodeBlock();

/**
 * 块解密
*/
void decodeBlock();

/**
 * 加密
*/
void encode();

/**
 * 解密
*/
void decode();

int main(); // 主函数
```

C语言代码

完整代码如下：

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <ctype.h>
#include <string.h>

#define BLOCK64 64           // 01位块长度
#define BLOCK8 9             // 8字节明文块长度，由于字符串限制，必须+1
#define EEXTAND 48           // E-拓展串
#define SUBKEYLEN 48         // 子密钥长度
#define SUBKEYNUM 16         // 子密钥数量
#define KEYLEN 64            // 密钥长度
#define NOCHECKDIGITLEN 56 // 非校验位长度

typedef bool des1_t;
typedef unsigned char des8_t;
```

```

// IP 置换表
const int IP_TABLE[BLOCK64] = {
    58, 50, 42, 34, 26, 18, 10, 2,
    60, 52, 44, 36, 28, 20, 12, 4,
    62, 54, 46, 38, 30, 22, 14, 6,
    64, 56, 48, 40, 32, 24, 16, 8,
    57, 49, 41, 33, 25, 17, 9, 1,
    59, 51, 43, 35, 27, 19, 11, 3,
    61, 53, 45, 37, 29, 21, 13, 5,
    63, 55, 47, 39, 31, 23, 15, 7};

// IP逆 置换表
const int IP_TABLE_REVERSE[BLOCK64] = {
    40, 8, 48, 16, 56, 24, 64, 32,
    39, 7, 47, 15, 55, 23, 63, 31,
    38, 6, 46, 14, 54, 22, 62, 30,
    37, 5, 45, 13, 53, 21, 61, 29,
    36, 4, 44, 12, 52, 20, 60, 28,
    35, 3, 43, 11, 51, 19, 59, 27,
    34, 2, 42, 10, 50, 18, 58, 26,
    33, 1, 41, 9, 49, 17, 57, 25};

// P-置换
const int P_TABLE[BLOCK64 / 2] = {
    16, 7, 20, 21,
    29, 12, 28, 17,
    1, 15, 23, 26,
    5, 18, 31, 10,
    2, 8, 24, 14,
    32, 27, 3, 9,
    19, 13, 30, 6,
    22, 11, 4, 25};

// PC-1 置换表
const int PC_1_TABLE[NOCHECKDIGITLEN] = {
    // C0
    57, 49, 41, 33, 25, 17, 9,
    11, 58, 50, 42, 34, 26, 18,
    10, 2, 59, 51, 43, 35, 27,
    19, 11, 3, 60, 52, 44, 36,

    // D0
    63, 55, 47, 39, 31, 23, 15,
    7, 62, 54, 46, 38, 30, 22,
    14, 6, 61, 53, 45, 37, 29,
    21, 13, 5, 28, 20, 12, 4};

// PC-2 置换表
const int PC_2_TABLE[SUBKEYLEN] = {
    14, 17, 11, 24, 1, 5,
    3, 28, 15, 6, 21, 10,
    23, 19, 12, 4, 26, 8,
    16, 7, 27, 20, 13, 2,

    41, 52, 31, 37, 47, 55,
    30, 40, 51, 45, 33, 48,
    44, 49, 39, 56, 34, 53,
    46, 42, 50, 36, 29, 32};

// S 盒
const int S_BOX[][BLOCK64] = {
    {14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7,
     0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8,
     4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0,
     15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13},

    {15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10,
     3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5,
     0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15,
     13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9},

    {10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8,
     13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1,
     13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7,
     1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12},

    {7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15,
     13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9,
     10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4,
     3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14},

    {2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9,
     14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6,
     4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14,
     11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3},

    {12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11,
     10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8,
     9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6,
     4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13},

    {4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1,
     13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6,
     1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2,
     6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12},

    {13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7,
     1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2,
     7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8,
     2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11}};

// E-拓规则（比特-选择表）
const int E_EXTAND[SUBKEYLEN] = {
    32, 1, 2, 3, 4, 5,
    4, 5, 6, 7, 8, 9,
    8, 9, 10, 11, 12, 13,
    12, 13, 14, 15, 16, 17,
    16, 17, 18, 19, 20, 21,
    20, 21, 22, 23, 24, 25,
    24, 25, 26, 27, 28, 29,
    28, 29, 30, 31, 32, 1};

des8_t block8[BLOCK8];           // 明文
des8_t encodedBlock8[BLOCK8];    // 加密后的明文
des1_t block64[BLOCK64];         // 二进制明文
```

```c
des1_t encodedBlock64[BLOCK64];  // 加密后的二进制明文
des1_t encodingBlock64[BLOCK64]; // 加密中的二进制明文
des1_t decodedBlock64[BLOCK64];  // 解密后的二进制明文
des8_t decodedBlock8[BLOCK8];    // 解密后的明文
des1_t decodingBlock64[BLOCK64]; // 解密中的二进制明文

char InitKey[KEYLEN / 4 + 1];        // 16进制的输入
des1_t Key[BLOCK64];                 // 密钥
des1_t Subkey[SUBKEYNUM][SUBKEYLEN]; // 子密钥

FILE *readFile; // 读取的文件

/**
 * 通过密钥生成子密钥，总共生成16个
 * @param K des1_t* 密钥
*/
void getSubkey(des1_t *K);

/**
 * 8字节 转换成 64位
 * @param from des8_t* 源数组
 * @param to des1_t* 目标数组
*/
void block8ToBlock64(des8_t *from, des1_t *to);

/**
 * 64位 转换为 8字节
 * @param from des1_t* 源数组
 * @param to des8_t* 目标数组
*/
void block64ToBlock8(des1_t *from, des8_t *to);

/**
 * 通过初始获取的密钥进行转换
*/
void getKey();

/**
 * 轮函数
 * @param Ri des1_t*
 * @param iterationNum int 迭代次数
 * @return 一个32位数组指针
*/
des1_t *Feistel(des1_t *Ri, int iteraionNum);

/**
 * 块加密
*/
void encodeBlock();

/**
 * 块解密
*/
void decodeBlock();

/**
 * 加密
*/
```

```
void encode();

/**
 * 解密
*/
void decode();

int main(char argc, char **argv)
{
  if (argc != 4)
  {
    printf("usage: ./out [enc | dec] key filename\n");
    return 0;
  }
  else
  {
    strcpy(InitKey, argv[2]); // 获取密钥
    getKey();
    getSubkey(Key);
    readFile = fopen(argv[3], "r"); // 打开文件
    if (strcmp(argv[1], "enc") == 0)
    {
      encode();
    }
    else if (strcmp(argv[1], "dec") == 0)
    {
      decode();
    }
    fclose(readFile); // 关闭文件
  }
}

void decode()
{
  int len = 0;
  while ((len = fread(encodedBlock8, 1, 8, readFile)) != 0)
  {
    encodedBlock8[len] = 0;
    // printf("%s", encodedBlock8);
    block8ToBlock64(encodedBlock8, encodedBlock64);
    decodeBlock();
    block64ToBlock8(decodedBlock64, decodedBlock8);

    // 去除填充
    decodedBlock8[8] = 0;
    int tail = decodedBlock8[7]; // 看末尾那位是否是填充的
    bool isPadding = true;
    for (int i = 8 - tail; i < BLOCK8 - 1; i++)
    {
      if (decodedBlock8[i] != tail) // 不是填充
      {
        isPadding = false;
        break;
      }
    }

    if (isPadding)
    {
      decodedBlock8[8 - tail] = 0;
    }
    printf("%s", decodedBlock8);
  }
}

void encode()
{
  bool padding = false; // 判定是否已经补全
  int len = 0;
  while (!feof(readFile))
  {
    len = fread(block8, 1, 8, readFile);
    block8[len] = 0;
    if (len < 8)
    {
      for (int i = len; i < 8; i++)
      {
        block8[i] = 8 - len; // 填充
      }
      block8[8] = 0;
      padding = true;
    }
    block8ToBlock64(block8, block64);
    encodeBlock();
    block64ToBlock8(encodedBlock64, encodedBlock8);
    for (int i = 0; i < 8; i++)
    {
      putchar(encodedBlock8[i]);
    }
  }

  // 如果刚好输入完成，那么需要补一个块
  if (!padding)
  {
    for (int i = 0; i < 8; i++)
    {
      block8[i] = 0x08;
    }
    block8[8] = 0;
    block8ToBlock64(block8, block64);
    encodeBlock();
    block64ToBlock8(encodedBlock64, encodedBlock8);
    for (int i = 0; i < 8; i++)
    {
      putchar(encodedBlock8[i]);
    }
  }
}

void block8ToBlock64(des8_t *from, des1_t *to)
{
  for (int i = 0; i < 8; i++)
  {
    des8_t tmp = from[i];
    for (int j = 0; j < 8; j++)
    {
      to[i * 8 + j] = (tmp >> (7 - j)) & 1;
    }
  }
}

void block64ToBlock8(des1_t *from, des8_t *to)
{
  for (int i = 0; i < 8; i++)
  {
    des8_t tmp = 0;
    for (int j = 0; j < 8; j++)
    {
      tmp = (tmp << 1) + from[i * 8 + j];
    }
    to[i] = tmp;
  }
}

void encodeBlock()
{
  // 初始置换 IP
  for (int i = 0; i < BLOCK64; i++)
  {
    encodingBlock64[i] = block64[IP_TABLE[i] - 1];
  }
  // 16次迭代
  des1_t *Li = encodingBlock64;               // 初始化 L0
  des1_t *Ri = encodingBlock64 + BLOCK64 / 2; // 初始化 R0

  for (int i = 0; i < BLOCK64 / 4; i++)
  {
    des1_t *tmp = Feistel(Ri, i); // 轮函数结果
    des1_t L_tmp, R_tmp;
    for (int j = 0; j < BLOCK64 / 2; j++)
    {
      L_tmp = Ri[j];
      R_tmp = Li[j] ^ tmp[j];

      Li[j] = L_tmp;
      Ri[j] = R_tmp;
    }
  }

  // 交换置换
  for (int i = 0; i < BLOCK64 / 2; i++)
  {
    des1_t tmp = Li[i];
    Li[i] = Ri[i];
    Ri[i] = tmp;
  }
  for (int i = 0; i < BLOCK64; i++)
  {
    encodedBlock64[i] = encodingBlock64[IP_TABLE_REVERSE[i] - 1];
  }
}

des1_t *Feistel(des1_t *Ri, int iteraionNum)
{
  // E 拓展
  des1_t e_extand[48]; // E 拓展结果
  for (int i = 0; i < EEXTAND; i++)
  {
    e_extand[i] = Ri[E_EXTAND[i] - 1];
  }

  des1_t xorList[48]; // 异或的结果
  for (int i = 0; i < EEXTAND; i++)
  {
    xorList[i] = e_extand[i] ^ Subkey[iteraionNum][i];
  }

  // S 盒压缩
  des1_t s_box_res[32]; // S 盒压缩结果
  for (int i = 0; i < 8; i++)
  {
    int n = (xorList[i * 6] << 1) + xorList[i * 6 + 5]; // 确定行号
    int m = (xorList[i * 6 + 1] << 3) + (xorList[i * 6 + 2] << 2) + (xorList[i * 6 + 3] << 1) + xorList[i * 6 + 4]; // 获取列号

    des8_t res = S_BOX[i][n * BLOCK64 / 4 + m];

    for (int j = 0; j < 4; j++)
    {
      s_box_res[i * 4 + j] = (res >> (3 - j)) & 1;
    }
  }

  static des1_t p_res[BLOCK64 / 2]; // P 置换的结果
  for (int i = 0; i < BLOCK64 / 2; i++)
  {
    p_res[i] = s_box_res[P_TABLE[i] - 1];
  }

  return p_res;
}

void getKey()
{
  for (int i = 0; i < 16; i++)
  {
    int moveBit = i % 2 == 0 ? 4 : 0;
    int tmp = InitKey[i] = tolower(InitKey[i]);
    if (isdigit(tmp))
    {
      tmp -= '0';
      InitKey[i] = tmp;
    }
    else
    {
      tmp -= ('a' - 10);
      InitKey[i] = tmp;
    }

    for (int j = 0; j < 4; j++)
    {
      Key[i * 4 + j] = (tmp >> (3 - j)) & 1;
```

```c
    }
  }
}

void getSubkey(des1_t *K)
{

  // 进行初始的 PC-1 置换
  des1_t CD[NOCHECKDIGITLEN];
  for (int i = 0; i < NOCHECKDIGITLEN; i++)
  {
    CD[i] = K[PC_1_TABLE[i] - 1];
  }

  // 循环生成
  for (int i = 0; i < SUBKEYNUM; i++)
  {

    // 进行 LS 操作
    if (i == 0 || i == 1 || i == 8 || i == 15) // 需要循环左移1个位置
    {
      des1_t tmpC = CD[0];                   // 对 C
      des1_t tmpD = CD[NOCHECKDIGITLEN / 2]; // 对 D
      for (int j = 0; j < NOCHECKDIGITLEN / 2 - 1; j++)
      {
        CD[j] = CD[j + 1];
        CD[j + NOCHECKDIGITLEN / 2] = CD[j + NOCHECKDIGITLEN / 2 + 1];
      }
      CD[NOCHECKDIGITLEN / 2 - 1] = tmpC;
      CD[NOCHECKDIGITLEN - 1] = tmpD;
    }
    else // 否则循环左移2个位置
    {
      des1_t tmpC1 = CD[0], tmpC2 = CD[1];
      des1_t tmpD1 = CD[NOCHECKDIGITLEN / 2], tmpD2 = CD[NOCHECKDIGITLEN / 2 + 1];
      for (int j = 0; j < NOCHECKDIGITLEN / 2 - 2; j++)
      {
        CD[j] = CD[j + 2];
        CD[j + NOCHECKDIGITLEN / 2] = CD[j + NOCHECKDIGITLEN / 2 + 2];
      }
      CD[NOCHECKDIGITLEN / 2 - 2] = tmpC1;
      CD[NOCHECKDIGITLEN / 2 - 1] = tmpC2;
      CD[NOCHECKDIGITLEN - 2] = tmpD1;
      CD[NOCHECKDIGITLEN - 1] = tmpD2;
    }

    // PC-2 压缩置换
    for (int j = 0; j < SUBKEYLEN; j++)
    {
      Subkey[i][j] = CD[PC_2_TABLE[j] - 1];
    }
  }
}

void decodeBlock()
{
  // 初始置换 IP
  for (int i = 0; i < BLOCK64; i++)
  {
    decodingBlock64[i] = encodedBlock64[IP_TABLE[i] - 1];
  }
  // 16次迭代
  des1_t *Li = decodingBlock64;               // 初始化 L0
  des1_t *Ri = decodingBlock64 + BLOCK64 / 2; // 初始化 R0

  for (int i = BLOCK64 / 4 - 1; i >= 0; i--)
  {
    des1_t *tmp = Feistel(Ri, i); // 轮函数结果
    des1_t L_tmp, R_tmp;
    for (int j = 0; j < BLOCK64 / 2; j++)
    {
      L_tmp = Ri[j];
      R_tmp = Li[j] ^ tmp[j];

      Li[j] = L_tmp;
      Ri[j] = R_tmp;
    }
  }

  // 交换置换
  for (int i = 0; i < BLOCK64 / 2; i++)
  {
    des1_t tmp = Li[i];
    Li[i] = Ri[i];
    Ri[i] = tmp;
  }

  // 逆置换
  for (int i = 0; i < BLOCK64; i++)
  {
    decodedBlock64[i] = decodingBlock64[IP_TABLE_REVERSE[i] - 1];
  }
}
```

编译运行结果

编译运行环境为 WSL：

```
Linux LAPTOP-QTCGESHO 4.4.0-19041-Microsoft #488-Microsoft Mon Sep 01 13:43:00 PST 2020 x86_64 x86_64 x86_64 GNU/Linux
```

使用 makefile 设置了相关的命令，文件代码如下,使用 openssl 进行加密解密的对照：

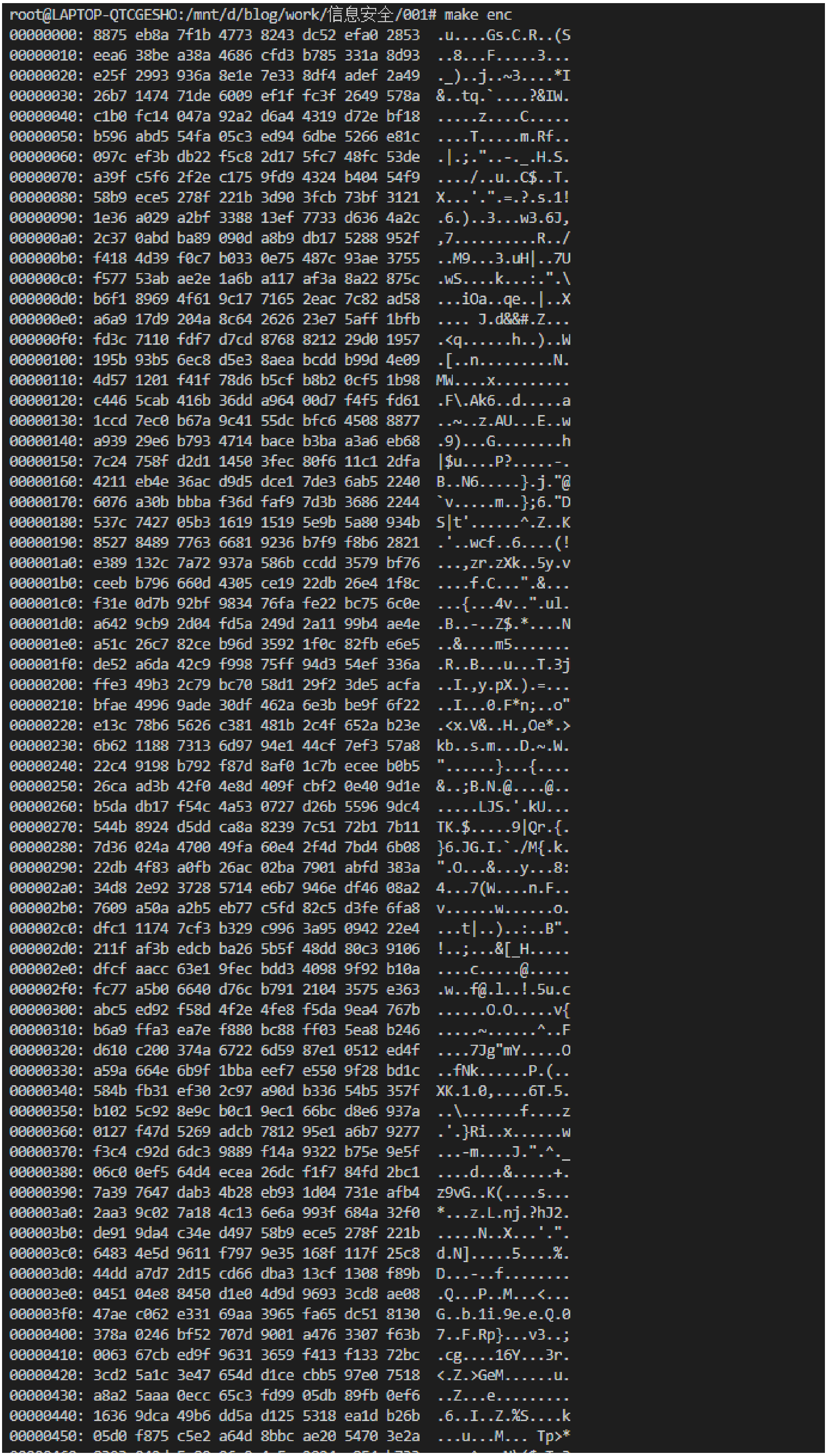
```
KEY = a1b2c3d4e5f6f7e8 # 密钥，请务必保证是64位
IN := ./in.txt # 输入的 txt 文件名

# openssl 相关，主要用于验证
SENC := ./senc.txt          # openssl 加密输出的文件名
SDEC := ./sdec.txt          # openssl 解密输出的文件名
ENCMODE := enc -e -des-ecb  # 加密模式
DECMODE := enc -d -des-ecb  # 解密模式

# C 代码相关
```

```
GCC := gcc                      # 编译器
INC := ./des.c              # 源代码
OUTC := ./des               # 编译出的程序
CENC := ./cenc.txt # 加密输出的文件名
CDEC := ./cdec.txt # 解密输出的文件名

# C 代码进行加密操作
enc:
    @${GCC} ${INC} -o ${OUTC}
    @${OUTC} enc ${KEY} ${IN} > ${CENC}
    @xxd ${CENC}

# C 代码进行解密操作
dec:
    @${GCC} ${INC} -o ${OUTC}
    @${OUTC} dec ${KEY} ${CENC} > ${CDEC}
    @xxd ${CDEC}

# 使用 openssl 进行加密操作
senc:
    @openssl ${ENCMODE} -K ${KEY} -in ${IN} -out ${SENC}
    @xxd ${SENC}

# 使用 openssl 进行解密操作
sdec:
    @openssl ${DECMODE} -K ${KEY} -in ${SENC} -out ${SDEC}
    @xxd ${SDEC}

# 比较 C 代码和 openssl 加密结果
enc-diff:
    @diff -y ${CENC} ${SENC} || exit 0
    @echo ''

# 比较 C 代码和 openssl 解密结果
dec-diff:
    @diff -y ${CDEC} ${SDEC} || exit 0
    @echo ''

# 清除
clean:
    @rm ${OUTC} || exit 0
```

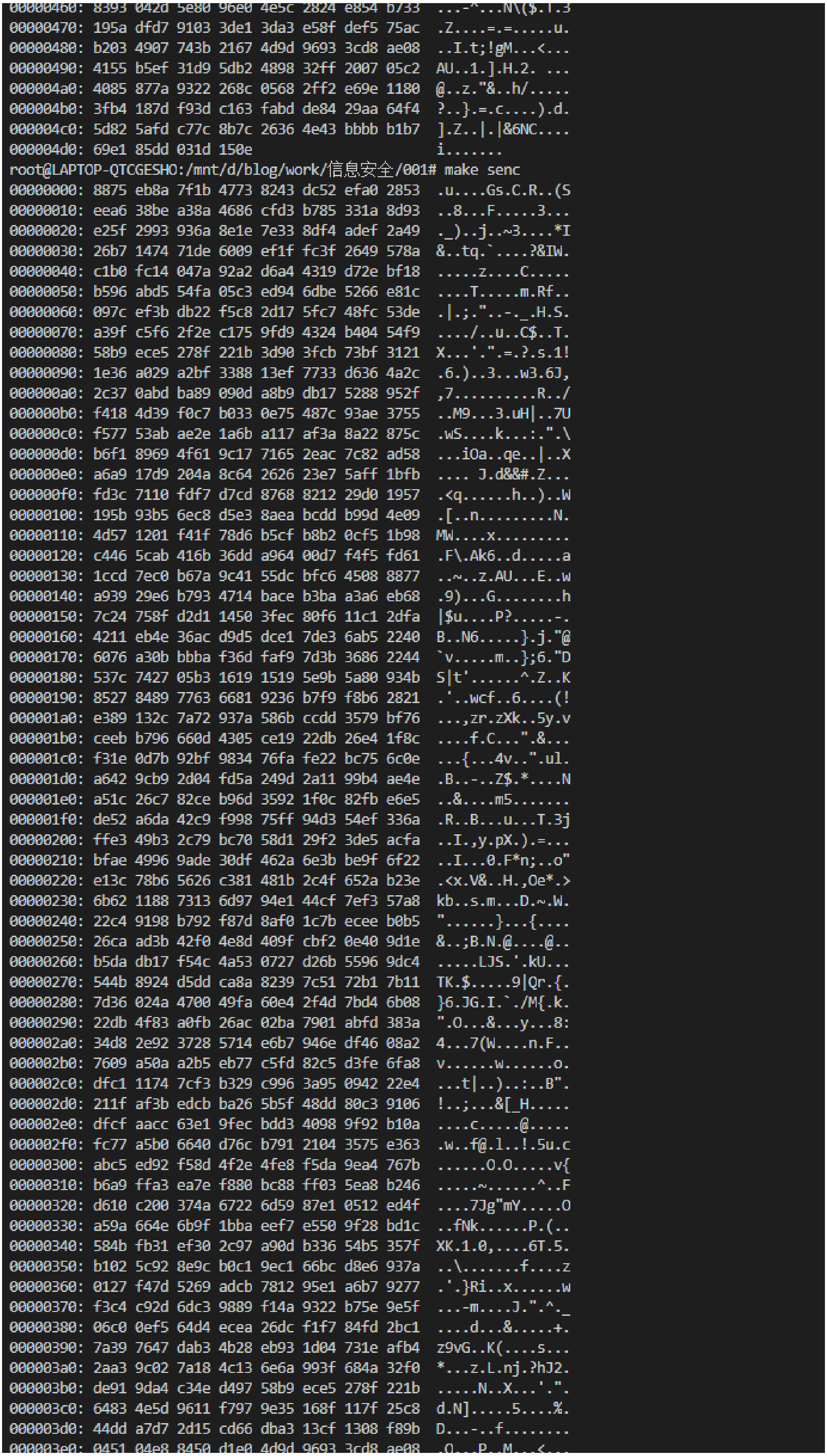
设置明文如下：

```
*Astronomy* in Elizabethan times was much closer to what we would nowadays term astrology.
It was not yet weighted down with knowledge of what the planets and stars actually are, as modern day astronomy is.
There was a widespread belief that the stars, in their various conjunctions, had an important and direct influence on the life of humans, both on individuals, and on social institutions.
See the sonnet by Sidney, given at the bottom of the page.
He calls those who consider the stars to shine merely to spangle the night 'dusty wits', for to him their importance was much greater.
They were an importance influence in human lives.
Although his sonnet, like this one, by its conclusion is somewhat tongue in cheek.
(Note that Sidney uses the term astrology. He also reads Stellas's eyes as if they were stars).
The poet here claims to 'have Astronomy', i.e he understands it as a science, and then he proceeds to tell us how his knowledge differs from that of the traditional astrologer (lines 3-8).
We tend to think of ourselves as a more rational age, but a recent president of the United States, Ronald Reagan, relied on his wife's astrologer to forecast for him propitious days for work and policy decisions.
```

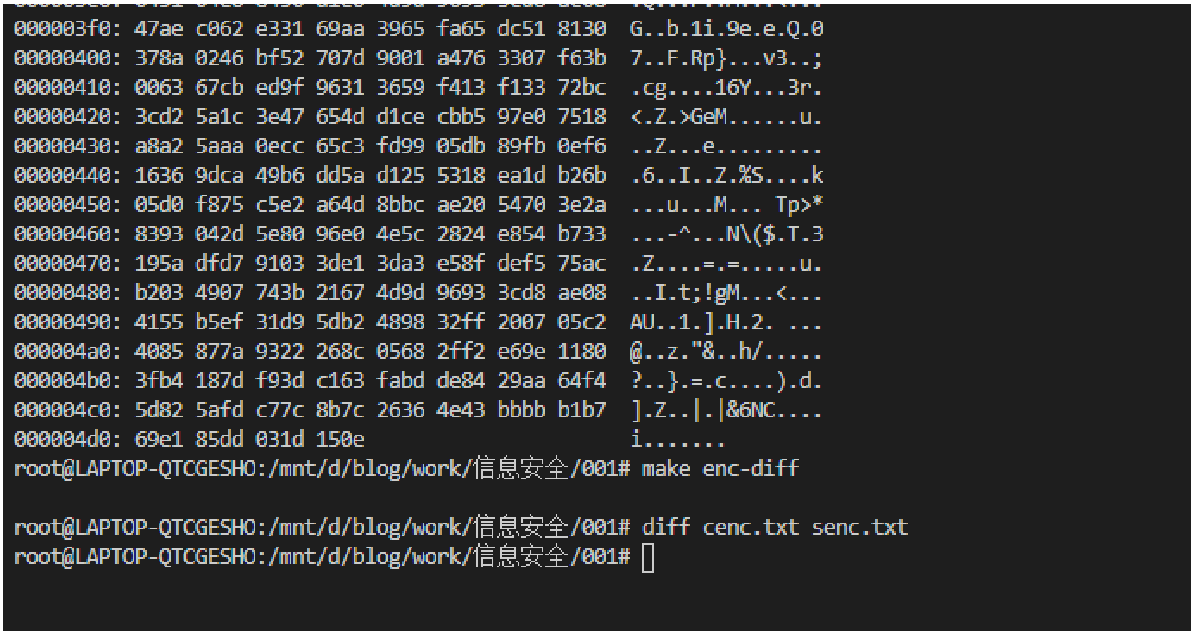
进行加密测试，运行结果如下：



```
root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# make enc
00000000: 8875 eb8a 7f1b 4773 8243 dc52 efa0 2853  .u....Gs.C.R..(S
00000010: eea6 38be a38a 4686 cfd3 b785 331a 8d93  ..8...F.....3...
00000020: e25f 2993 936a 8e1e 7e33 8df4 adef 2a49  ._)..j..~3....*I
00000030: 26b7 1474 71de 6009 ef1f fc3f 2649 578a  &..tq.`....?&IW.
00000040: c1b0 fc14 047a 92a2 d6a4 4319 d72e bf18  .....z....C.....
00000050: b596 abd5 54fa 05c3 ed94 6dbe 5266 e81c  ....T.....m.Rf..
00000060: 097c ef3b db22 f5c8 2d17 5fc7 48fc 53de  .|.;."..-._.H.S.
00000070: a39f c5f6 2f2e c175 9fd9 4324 b404 54f9  ..../..u..C$..T.
00000080: 58b9 ece5 278f 221b 3d90 3fcb 73bf 3121  X...'.".=.?.s.1!
00000090: 1e36 a029 a2bf 3388 13ef 7733 d636 4a2c  .6.)..3...w3.6J,
000000a0: 2c37 0abd ba89 090d a8b9 db17 5288 952f  ,7..........R../
000000b0: f418 4d39 f0c7 b033 0e75 487c 93ae 3755  ..M9...3.uH|..7U
000000c0: f577 53ab ae2e 1a6b a117 af3a 8a22 875c  .wS....k...:.".\
000000d0: b6f1 8969 4f61 9c17 7165 2eac 7c82 ad58  ...iOa..qe..|..X
000000e0: a6a9 17d9 204a 8c64 2626 23e7 5aff 1bfb  .... J.d&&#.Z...
000000f0: fd3c 7110 fdf7 d7cd 8768 8212 29d0 1957  .<q......h..)..W
00000100: 195b 93b5 6ec8 d5e3 8aea bcdd b99d 4e09  .[..n.........N.
00000110: 4d57 1201 f41f 78d6 b5cf b8b2 0cf5 1b98  MW....x.........
00000120: c446 5cab 416b 36dd a964 00d7 f4f5 fd61  .F\.Ak6..d.....a
00000130: 1ccd 7ec0 b67a 9c41 55dc bfc6 4508 8877  ..~..z.AU...E..w
00000140: a939 29e6 b793 4714 bace b3ba a3a6 eb68  .9)...G........h
00000150: 7c24 758f d2d1 1450 3fec 80f6 11c1 2dfa  |$u....P?.....-.
00000160: 4211 eb4e 36ac d9d5 dce1 7de3 6ab5 2240  B..N6.....}.j."@
00000170: 6076 a30b bbba f36d faf9 7d3b 3686 2244  `v.....m..};6."D
00000180: 537c 7427 05b3 1619 1519 5e9b 5a80 934b  S|t'......^.Z..K
00000190: 8527 8489 7763 6681 9236 b7f9 f8b6 2821  .'..wcf..6....(!
000001a0: e389 132c 7a72 937a 586b ccdd 3579 bf76  ...,zr.zXk..5y.v
000001b0: ceeb b796 660d 4305 ce19 22db 26e4 1f8c  ....f.C...".&...
000001c0: f31e 0d7b 92bf 9834 76fa fe22 bc75 6c0e  ...{...4v..".ul.
000001d0: a642 9cb9 2d04 fd5a 249d 2a11 99b4 ae4e  .B..-..Z$.*....N
000001e0: a51c 26c7 82ce b96d 3592 1f0c 82fb e6e5  ..&....m5.......
000001f0: de52 a6da 42c9 f998 75ff 94d3 54ef 336a  .R..B...u...T.3j
00000200: ffe3 49b3 2c79 bc70 58d1 29f2 3de5 acfa  ..I.,y.pX.).=...
00000210: bfae 4996 9ade 30df 462a 6e3b be9f 6f22  ..I...0.F*n;..o"
00000220: e13c 78b6 5626 c381 481b 2c4f 652a b23e  .<x.V&..H.,Oe*.>
00000230: 6b62 1188 7313 6d97 94e1 44cf 7ef3 57a8  kb..s.m...D.~.W.
00000240: 22c4 9198 b792 f87d 8af0 1c7b ecee b0b5  "......}...{....
00000250: 26ca ad3b 42f0 4e8d 409f cbf2 0e40 9d1e  &..;B.N.@....@..
00000260: b5da db17 f54c 4a53 0727 d26b 5596 9dc4  .....LJS.'.kU...
00000270: 544b 8924 d5dd ca8a 8239 7c51 72b1 7b11  TK.$.....9|Qr.{.
00000280: 7d36 024a 4700 49fa 60e4 2f4d 7bd4 6b08  }6.JG.I.`./M{.k.
00000290: 22db 4f83 a0fb 26ac 02ba 7901 abfd 383a  ".O...&...y...8:
000002a0: 34d8 2e92 3728 5714 e6b7 946e df46 08a2  4...7(W....n.F..
000002b0: 7609 a50a a2b5 eb77 c5fd 82c5 d3fe 6fa8  v......w......o.
000002c0: dfc1 1174 7cf3 b329 c996 3a95 0942 22e4  ...t|..)..:..B".
000002d0: 211f af3b edcb ba26 5b5f 48dd 80c3 9106  !..;...&[_H.....
000002e0: dfcf aacc 63e1 9fec bdd3 4098 9f92 b10a  ....c.....@.....
000002f0: fc77 a5b0 6640 d76c b791 2104 3575 e363  .w..f@.l..!.5u.c
00000300: abc5 ed92 f58d 4f2e 4fe8 f5da 9ea4 767b  ......O.O.....v{
00000310: b6a9 ffa3 ea7e f880 bc88 ff03 5ea8 b246  .....~......^..F
00000320: d610 c200 374a 6722 6d59 87e1 0512 ed4f  ....7Jg"mY.....O
00000330: a59a 664e 6b9f 1bba eef7 e550 9f28 bd1c  ..fNk......P.(..
00000340: 584b fb31 ef30 2c97 a90d b336 54b5 357f  XK.1.0,.....6T.5.
00000350: b102 5c92 8e9c b0c1 9ec1 66bc d8e6 937a  ..\.......f....z
00000360: 0127 f47d 5269 adcb 7812 95e1 a6b7 9277  .'.}Ri..x......w
00000370: f3c4 c92d 6dc3 9889 f14a 9322 b75e 9e5f  ...-m....J.".^._
00000380: 06c0 0ef5 64d4 ecea 26dc f1f7 84fd 2bc1  ....d...&.....+.
00000390: 7a39 7647 dab3 4b28 eb93 1d04 731e afb4  z9vG..K(....s...
000003a0: 2aa3 9c02 7a18 4c13 6e6a 993f 684a 32f0  *...z.L.nj.?hJ2.
000003b0: de91 9da4 c34e d497 58b9 ece5 278f 221b  .....N..X...'.".
000003c0: 6483 4e5d 9611 f797 9e35 168f 117f 25c8  d.N].....5....%.
000003d0: 44dd a7d7 2d15 cd66 dba3 13cf 1308 f89b  D...-..f........
000003e0: 0451 04e8 8450 d1e0 4d9d 9693 3cd8 ae08  .Q...P..M...<...
000003f0: 47ae c062 e331 69aa 3965 fa65 dc51 8130  G..b.1i.9e.e.Q.0
00000400: 378a 0246 bf52 707d 9001 a476 3307 f63b  7..F.Rp}...v3..;
00000410: 0063 67cb ed9f 9631 3659 f413 f133 72bc  .cg....16Y...3r.
00000420: 3cd2 5a1c 3e47 654d d1ce cbb5 97e0 7518  <.Z.>GeM......u.
00000430: a8a2 5aaa 0ecc 65c3 fd99 05db 89fb 0ef6  ..Z...e.........
00000440: 1636 9dca 49b6 dd5a d125 5318 ea1d b26b  .6..I..Z.%S....k
00000450: 05d0 f875 c5e2 a64d 8bbc ae20 5470 3e2a  ...u...M... Tp>*
```

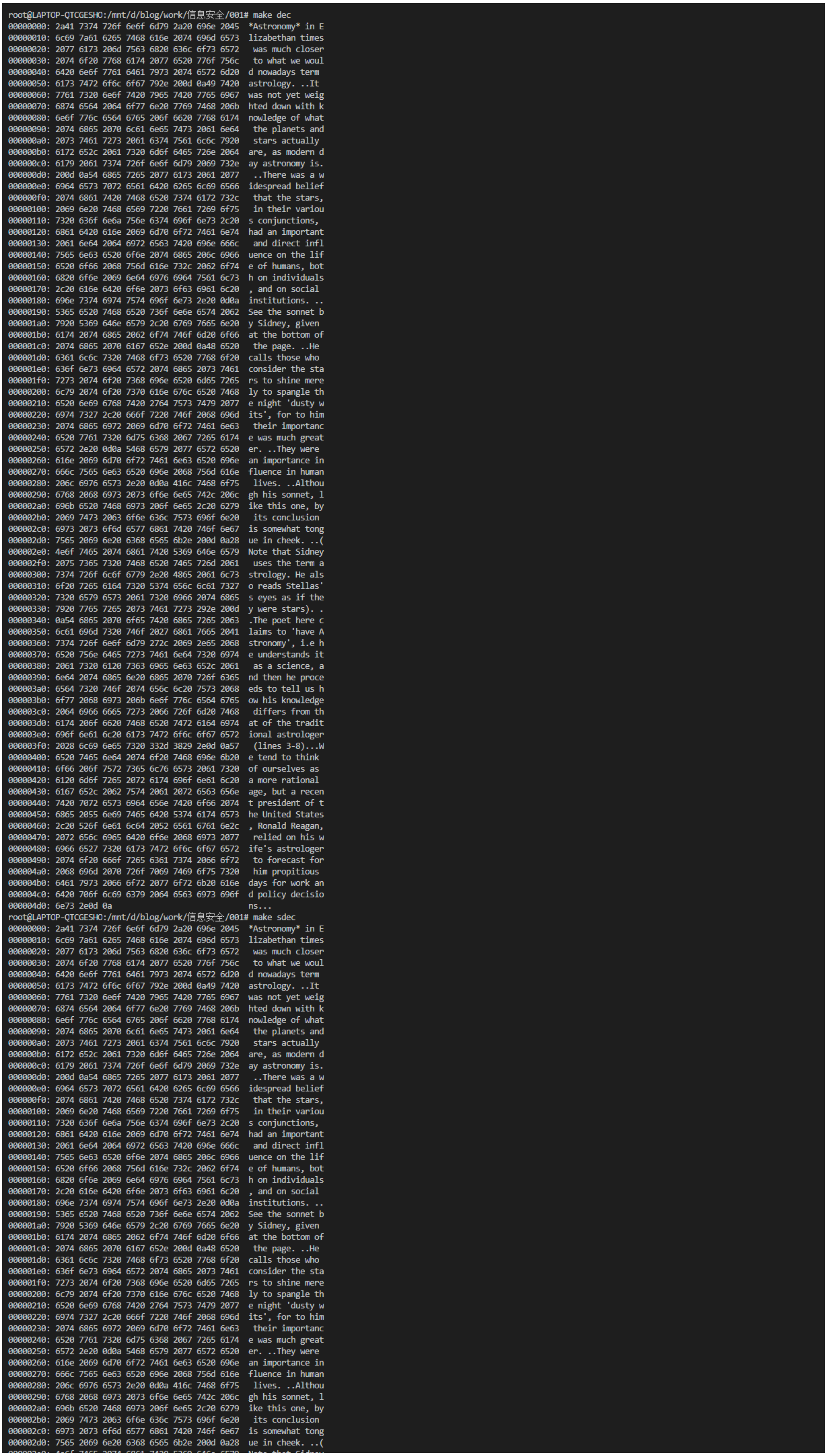


```
00000460: 8393 042d 5e80 96e0 4e5c 2824 e854 b733  ...-^...N\($.T.3
00000470: 195a dfd7 9103 3de1 3da3 e58f def5 75ac  .Z....=.=.....u.
00000480: b203 4907 743b 2167 4d9d 9693 3cd8 ae08  ..I.t;!gM...<...
00000490: 4155 b5ef 31d9 5db2 4898 32ff 2007 05c2  AU..1.].H.2. ...
000004a0: 4085 877a 9322 268c 0568 2ff2 e69e 1180  @..z."&..h/.....
000004b0: 3fb4 187d f93d c163 fabd de84 29aa 64f4  ?..}.=.c....).d.
000004c0: 5d82 5afd c77c 8b7c 2636 4e43 bbbb b1b7  ].Z..|.|&6NC....
000004d0: 69e1 85dd 031d 150e                      i.......
root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# make senc
00000000: 8875 eb8a 7f1b 4773 8243 dc52 efa0 2853  .u....Gs.C.R..(S
00000010: eea6 38be a38a 4686 cfd3 b785 331a 8d93  ..8...F.....3...
00000020: e25f 2993 936a 8e1e 7e33 8df4 adef 2a49  ._)..j..~3....*I
00000030: 26b7 1474 71de 6009 ef1f fc3f 2649 578a  &..tq.`....?&IW.
00000040: c1b0 fc14 047a 92a2 d6a4 4319 d72e bf18  .....z....C.....
00000050: b596 abd5 54fa 05c3 ed94 6dbe 5266 e81c  ....T.....m.Rf..
00000060: 097c ef3b db22 f5c8 2d17 5fc7 48fc 53de  .|.;."..-._.H.S.
00000070: a39f c5f6 2f2e c175 9fd9 4324 b404 54f9  ..../..u..C$..T.
00000080: 58b9 ece5 278f 221b 3d90 3fcb 73bf 3121  X...'.".=.?.s.1!
00000090: 1e36 a029 a2bf 3388 13ef 7733 d636 4a2c  .6.)..3...w3.6J,
000000a0: 2c37 0abd ba89 090d a8b9 db17 5288 952f  ,7..........R../
000000b0: f418 4d39 f0c7 b033 0e75 487c 93ae 3755  ..M9...3.uH|..7U
000000c0: f577 53ab ae2e 1a6b a117 af3a 8a22 875c  .wS....k...:.".\
000000d0: b6f1 8969 4f61 9c17 7165 2eac 7c82 ad58  ...iOa..qe..|..X
000000e0: a6a9 17d9 204a 8c64 2626 23e7 5aff 1bfb  .... J.d&&#.Z...
000000f0: fd3c 7110 fdf7 d7cd 8768 8212 29d0 1957  .<q......h..)..W
00000100: 195b 93b5 6ec8 d5e3 8aea bcdd b99d 4e09  .[..n.........N.
00000110: 4d57 1201 f41f 78d6 b5cf b8b2 0cf5 1b98  MW....x.........
00000120: c446 5cab 416b 36dd a964 00d7 f4f5 fd61  .F\.Ak6..d.....a
00000130: 1ccd 7ec0 b67a 9c41 55dc bfc6 4508 8877  ..~..z.AU...E..w
00000140: a939 29e6 b793 4714 bace b3ba a3a6 eb68  .9)...G........h
00000150: 7c24 758f d2d1 1450 3fec 80f6 11c1 2dfa  |$u....P?.....-.
00000160: 4211 eb4e 36ac d9d5 dce1 7de3 6ab5 2240  B..N6.....}.j."@
00000170: 6076 a30b bbba f36d faf9 7d3b 3686 2244  `v.....m..};6."D
00000180: 537c 7427 05b3 1619 1519 5e9b 5a80 934b  S|t'......^.Z..K
00000190: 8527 8489 7763 6681 9236 b7f9 f8b6 2821  .'..wcf..6....(!
000001a0: e389 132c 7a72 937a 586b ccdd 3579 bf76  ...,zr.zXk..5y.v
000001b0: ceeb b796 660d 4305 ce19 22db 26e4 1f8c  ....f.C...".&...
000001c0: f31e 0d7b 92bf 9834 76fa fe22 bc75 6c0e  ...{...4v..".ul.
000001d0: a642 9cb9 2d04 fd5a 249d 2a11 99b4 ae4e  .B..-..Z$.*....N
000001e0: a51c 26c7 82ce b96d 3592 1f0c 82fb e6e5  ..&....m5.......
000001f0: de52 a6da 42c9 f998 75ff 94d3 54ef 336a  .R..B...u...T.3j
00000200: ffe3 49b3 2c79 bc70 58d1 29f2 3de5 acfa  ..I.,y.pX.).=...
00000210: bfae 4996 9ade 30df 462a 6e3b be9f 6f22  ..I...0.F*n;..o"
00000220: e13c 78b6 5626 c381 481b 2c4f 652a b23e  .<x.V&..H.,Oe*.>
00000230: 6b62 1188 7313 6d97 94e1 44cf 7ef3 57a8  kb..s.m...D.~.W.
00000240: 22c4 9198 b792 f87d 8af0 1c7b ecee b0b5  "......}...{....
00000250: 26ca ad3b 42f0 4e8d 409f cbf2 0e40 9d1e  &..;B.N.@....@..
00000260: b5da db17 f54c 4a53 0727 d26b 5596 9dc4  .....LJS.'.kU...
00000270: 544b 8924 d5dd ca8a 8239 7c51 72b1 7b11  TK.$.....9|Qr.{.
00000280: 7d36 024a 4700 49fa 60e4 2f4d 7bd4 6b08  }6.JG.I.`./M{.k.
00000290: 22db 4f83 a0fb 26ac 02ba 7901 abfd 383a  ".O...&...y...8:
000002a0: 34d8 2e92 3728 5714 e6b7 946e df46 08a2  4...7(W....n.F..
000002b0: 7609 a50a a2b5 eb77 c5fd 82c5 d3fe 6fa8  v......w......o.
000002c0: dfc1 1174 7cf3 b329 c996 3a95 0942 22e4  ...t|..)..:..B".
000002d0: 211f af3b edcb ba26 5b5f 48dd 80c3 9106  !..;...&[_H.....
000002e0: dfcf aacc 63e1 9fec bdd3 4098 9f92 b10a  ....c.....@.....
000002f0: fc77 a5b0 6640 d76c b791 2104 3575 e363  .w..f@.l..!.5u.c
00000300: abc5 ed92 f58d 4f2e 4fe8 f5da 9ea4 767b  ......O.O.....v{
00000310: b6a9 ffa3 ea7e f880 bc88 ff03 5ea8 b246  .....~......^..F
00000320: d610 c200 374a 6722 6d59 87e1 0512 ed4f  ....7Jg"mY.....O
00000330: a59a 664e 6b9f 1bba eef7 e550 9f28 bd1c  ..fNk......P.(..
00000340: 584b fb31 ef30 2c97 a90d b336 54b5 357f  XK.1.0,....6T.5.
00000350: b102 5c92 8e9c b0c1 9ec1 66bc d8e6 937a  ..\.......f....z
00000360: 0127 f47d 5269 adcb 7812 95e1 a6b7 9277  .'.}Ri..x......w
00000370: f3c4 c92d 6dc3 9889 f14a 9322 b75e 9e5f  ...-m....J.".^._
00000380: 06c0 0ef5 64d4 ecea 26dc f1f7 84fd 2bc1  ....d...&.....+.
00000390: 7a39 7647 dab3 4b28 eb93 1d04 731e afb4  z9vG..K(....s...
000003a0: 2aa3 9c02 7a18 4c13 6e6a 993f 684a 32f0  *...z.L.nj.?hJ2.
000003b0: de91 9da4 c34e d497 58b9 ece5 278f 221b  .....N..X...'.".
000003c0: 6483 4e5d 9611 f797 9e35 168f 117f 25c8  d.N].....5....%.
000003d0: 44dd a7d7 2d15 cd66 dba3 13cf 1308 f89b  D...-..f........
000003e0: 0451 04e8 8450 d1e0 4d9d 9693 3cd8 ae08  .Q...P..M...<...
```

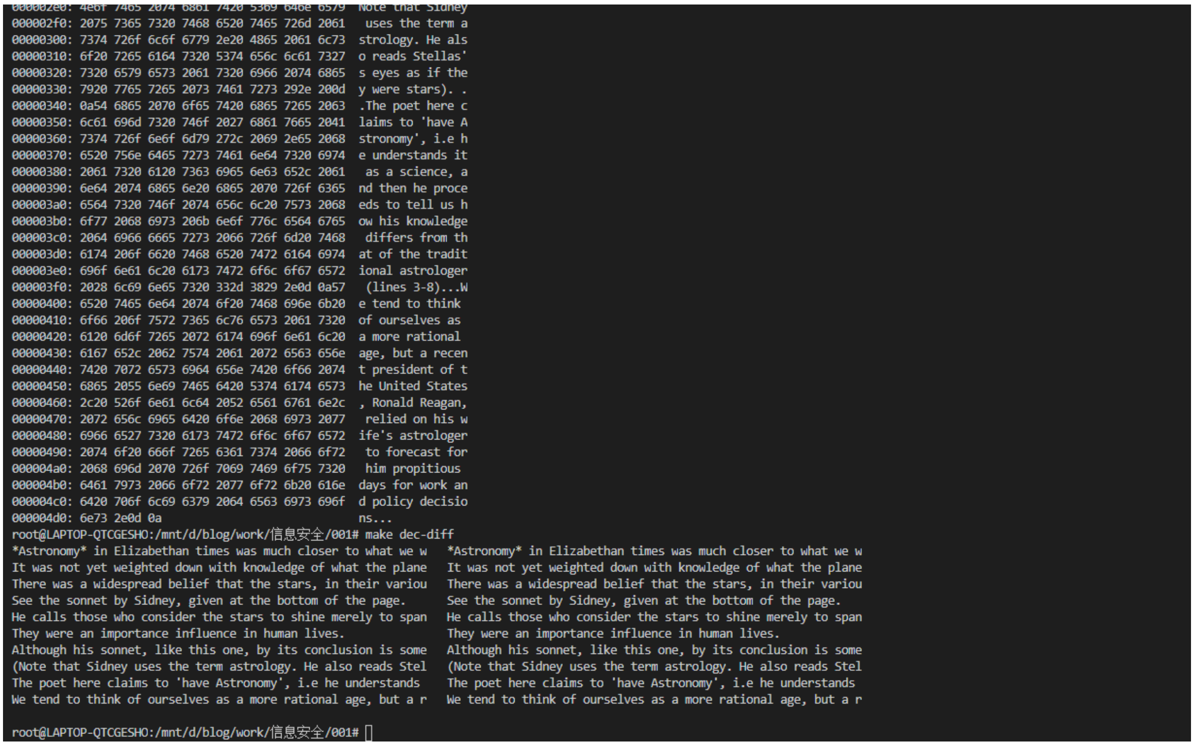


```
000003f0: 47ae c062 e331 69aa 3965 fa65 dc51 8130  G..b.1i.9e.e.Q.0
00000400: 378a 0246 bf52 707d 9001 a476 3307 f63b  7..F.Rp}...v3..;
00000410: 0063 67cb ed9f 9631 3659 f413 f133 72bc  .cg....16Y...3r.
00000420: 3cd2 5a1c 3e47 654d d1ce cbb5 97e0 7518  <.Z.>GeM......u.
00000430: a8a2 5aaa 0ecc 65c3 fd99 05db 89fb 0ef6  ..Z...e.........
00000440: 1636 9dca 49b6 dd5a d125 5318 ea1d b26b  .6..I..Z.%S....k
00000450: 05d0 f875 c5e2 a64d 8bbc ae20 5470 3e2a  ...u...M... Tp>*
00000460: 8393 042d 5e80 96e0 4e5c 2824 e854 b733  ...-^...N\($.T.3
00000470: 195a dfd7 9103 3de1 3da3 e58f def5 75ac  .Z....=.=.....u.
00000480: b203 4907 743b 2167 4d9d 9693 3cd8 ae08  ..I.t;!gM...<...
00000490: 4155 b5ef 31d9 5db2 4898 32ff 2007 05c2  AU..1.].H.2. ...
000004a0: 4085 877a 9322 268c 0568 2ff2 e69e 1180  @..z."&..h/.....
000004b0: 3fb4 187d f93d c163 fabd de84 29aa 64f4  ?..}.=.c....).d.
000004c0: 5d82 5afd c77c 8b7c 2636 4e43 bbbb b1b7  ].Z..|.|&6NC....
000004d0: 69e1 85dd 031d 150e                      i.......
root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# make enc-diff

root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# diff cenc.txt senc.txt
root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# 
```

使用上述加密后的文件，进行解密测试，运行结果如下：



```
root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# make dec
00000000: 2a41 7374 726f 6e6f 6d79 2a20 696e 2045  *Astronomy* in E
00000010: 6c69 7a61 6265 7468 616e 2074 696d 6573  lizabethan times
00000020: 2077 6173 206d 7563 6820 636c 6f73 6572   was much closer
00000030: 2074 6f20 7768 6174 2077 6520 776f 756c   to what we woul
00000040: 6420 6e6f 7761 6461 7973 2074 6572 6d20  d nowadays term
00000050: 6173 7472 6f6c 6f67 792e 200d 0a49 7420  astrology. ..It
00000060: 7761 7320 6e6f 7420 7965 7420 7765 6967  was not yet weig
00000070: 6874 6564 2064 6f77 6e20 7769 7468 206b  hted down with k
00000080: 6e6f 776c 6564 6765 206f 6620 7768 6174  nowledge of what
00000090: 2074 6865 2070 6c61 6e65 7473 2061 6e64   the planets and
000000a0: 2073 7461 7273 2061 6374 7561 6c6c 7920   stars actually
000000b0: 6172 652c 2061 7320 6d6f 6465 726e 2064  are, as modern d
000000c0: 6179 2061 7374 726f 6e6f 6d79 2069 732e  ay astronomy is.
000000d0: 200d 0a54 6865 7265 2077 6173 2061 2077   ..There was a w
000000e0: 6964 6573 7072 6561 6420 6265 6c69 6566  idespread belief
000000f0: 2074 6861 7420 7468 6520 7374 6172 732c   that the stars,
00000100: 2069 6e20 7468 6569 7220 7661 7269 6f75   in their variou
00000110: 7320 636f 6e6a 756e 6374 696f 6e73 2c20  s conjunctions,
00000120: 6861 6420 616e 2069 6d70 6f72 7461 6e74  had an important
00000130: 2061 6e64 2064 6972 6563 7420 696e 666c   and direct infl
00000140: 7565 6e63 6520 6f6e 2074 6865 206c 6966  uence on the lif
00000150: 6520 6f66 2068 756d 616e 732c 2062 6f74  e of humans, bot
00000160: 6820 6f6e 2069 6e64 6976 6964 7561 6c73  h on individuals
00000170: 2c20 616e 6420 6f6e 2073 6f63 6961 6c20  , and on social
00000180: 696e 7374 6974 7574 696f 6e73 2e20 0d0a  institutions. ..
00000190: 5365 6520 7468 6520 736f 6e6e 6574 2062  See the sonnet b
000001a0: 7920 5369 646e 6579 2c20 6769 7665 6e20  y Sidney, given
000001b0: 6174 2074 6865 2062 6f74 746f 6d20 6f66  at the bottom of
000001c0: 2074 6865 2070 6167 652e 200d 0a48 6520   the page. ..He
000001d0: 6361 6c6c 7320 7468 6f73 6520 7768 6f20  calls those who
000001e0: 636f 6e73 6964 6572 2074 6865 2073 7461  consider the sta
000001f0: 7273 2074 6f20 7368 696e 6520 6d65 7265  rs to shine mere
00000200: 6c79 2074 6f20 7370 616e 676c 6520 7468  ly to spangle th
00000210: 6520 6e69 6768 7420 2764 7573 7479 2077  e night 'dusty w
00000220: 6974 7327 2c20 666f 7220 746f 2068 696d  its', for to him
00000230: 2074 6865 6972 2069 6d70 6f72 7461 6e63   their importanc
00000240: 6520 7761 7320 6d75 6368 2067 7265 6174  e was much great
00000250: 6572 2e20 0d0a 5468 6579 2077 6572 6520  er. ..They were
00000260: 616e 2069 6d70 6f72 7461 6e63 6520 696e  an importance in
00000270: 666c 7565 6e63 6520 696e 2068 756d 616e  fluence in human
00000280: 206c 6976 6573 2e20 0d0a 416c 7468 6f75   lives. ..Althou
00000290: 6768 2068 6973 2073 6f6e 6e65 742c 206c  gh his sonnet, l
000002a0: 696b 6520 7468 6973 206f 6e65 2c20 6279  ike this one, by
000002b0: 2069 7473 2063 6f6e 636c 7573 696f 6e20   its conclusion
000002c0: 6973 2073 6f6d 6577 6861 7420 746f 6e67  is somewhat tong
000002d0: 7565 2069 6e20 6368 6565 6b2e 200d 0a28  ue in cheek. ..(
000002e0: 4e6f 7465 2074 6861 7420 5369 646e 6579  Note that Sidney
000002f0: 2075 7365 7320 7468 6520 7465 726d 2061   uses the term a
00000300: 7374 726f 6c6f 6779 2e20 4865 2061 6c73  strology. He als
00000310: 6f20 7265 6164 7320 5374 656c 6c61 7327  o reads Stellas'
00000320: 7320 6579 6573 2061 7320 6966 2074 6865  s eyes as if the
00000330: 7920 7765 7265 2073 7461 7273 292e 200d  y were stars). .
00000340: 0a54 6865 2070 6f65 7420 6865 7265 2063  .The poet here c
00000350: 6c61 696d 7320 746f 2027 6861 7665 2041  laims to 'have A
00000360: 7374 726f 6e6f 6d79 272c 2069 2e65 2068  stronomy', i.e h
00000370: 6520 756e 6465 7273 7461 6e64 7320 6974  e understands it
00000380: 2061 7320 6120 7363 6965 6e63 652c 2061   as a science, a
00000390: 6e64 2074 6865 6e20 6865 2070 726f 6365  nd then he proce
000003a0: 6564 7320 746f 2074 656c 6c20 7573 2068  eds to tell us h
000003b0: 6f77 2068 6973 206b 6e6f 776c 6564 6765  ow his knowledge
000003c0: 2064 6966 6665 7273 2066 726f 6d20 7468   differs from th
000003d0: 6174 206f 6620 7468 6520 7472 6164 6974  at of the tradit
000003e0: 696f 6e61 6c20 6173 7472 6f6c 6f67 6572  ional astrologer
000003f0: 2028 6c69 6e65 7320 332d 3829 2e0d 0a57   (lines 3-8)...W
00000400: 6520 7465 6e64 2074 6f20 7468 696e 6b20  e tend to think
00000410: 6f66 206f 7572 7365 6c76 6573 2061 7320  of ourselves as
00000420: 6120 6d6f 7265 2072 6174 696f 6e61 6c20  a more rational
00000430: 6167 652c 2062 7574 2061 2072 6563 656e  age, but a recen
00000440: 7420 7072 6573 6964 656e 7420 6f66 2074  t president of t
00000450: 6865 2055 6e69 7465 6420 5374 6174 6573  he United States
00000460: 2c20 526f 6e61 6c64 2052 6561 6761 6e2c  , Ronald Reagan,
00000470: 2072 656c 6965 6420 6f6e 2068 6973 2077   relied on his w
00000480: 6966 6527 7320 6173 7472 6f6c 6f67 6572  ife's astrologer
00000490: 2074 6f20 666f 7265 6361 7374 2066 6f72   to forecast for
000004a0: 2068 696d 2070 726f 7069 7469 6f75 7320   him propitious
000004b0: 6461 7973 2066 6f72 2077 6f72 6b20 616e  days for work an
000004c0: 6420 706f 6c69 6379 2064 6563 6973 696f  d policy decisio
000004d0: 6e73 2e0d 0a                             ns...
root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# make sdec
00000000: 2a41 7374 726f 6e6f 6d79 2a20 696e 2045  *Astronomy* in E
00000010: 6c69 7a61 6265 7468 616e 2074 696d 6573  lizabethan times
00000020: 2077 6173 206d 7563 6820 636c 6f73 6572   was much closer
00000030: 2074 6f20 7768 6174 2077 6520 776f 756c   to what we woul
00000040: 6420 6e6f 7761 6461 7973 2074 6572 6d20  d nowadays term
00000050: 6173 7472 6f6c 6f67 792e 200d 0a49 7420  astrology. ..It
00000060: 7761 7320 6e6f 7420 7965 7420 7765 6967  was not yet weig
00000070: 6874 6564 2064 6f77 6e20 7769 7468 206b  hted down with k
00000080: 6e6f 776c 6564 6765 206f 6620 7768 6174  nowledge of what
00000090: 2074 6865 2070 6c61 6e65 7473 2061 6e64   the planets and
000000a0: 2073 7461 7273 2061 6374 7561 6c6c 7920   stars actually
000000b0: 6172 652c 2061 7320 6d6f 6465 726e 2064  are, as modern d
000000c0: 6179 2061 7374 726f 6e6f 6d79 2069 732e  ay astronomy is.
000000d0: 200d 0a54 6865 7265 2077 6173 2061 2077   ..There was a w
000000e0: 6964 6573 7072 6561 6420 6265 6c69 6566  idespread belief
000000f0: 2074 6861 7420 7468 6520 7374 6172 732c   that the stars,
00000100: 2069 6e20 7468 6569 7220 7661 7269 6f75   in their variou
00000110: 7320 636f 6e6a 756e 6374 696f 6e73 2c20  s conjunctions,
00000120: 6861 6420 616e 2069 6d70 6f72 7461 6e74  had an important
00000130: 2061 6e64 2064 6972 6563 7420 696e 666c   and direct infl
00000140: 7565 6e63 6520 6f6e 2074 6865 206c 6966  uence on the lif
00000150: 6520 6f66 2068 756d 616e 732c 2062 6f74  e of humans, bot
00000160: 6820 6f6e 2069 6e64 6976 6964 7561 6c73  h on individuals
00000170: 2c20 616e 6420 6f6e 2073 6f63 6961 6c20  , and on social
00000180: 696e 7374 6974 7574 696f 6e73 2e20 0d0a  institutions. ..
00000190: 5365 6520 7468 6520 736f 6e6e 6574 2062  See the sonnet b
000001a0: 7920 5369 646e 6579 2c20 6769 7665 6e20  y Sidney, given
000001b0: 6174 2074 6865 2062 6f74 746f 6d20 6f66  at the bottom of
000001c0: 2074 6865 2070 6167 652e 200d 0a48 6520   the page. ..He
000001d0: 6361 6c6c 7320 7468 6f73 6520 7768 6f20  calls those who
000001e0: 636f 6e73 6964 6572 2074 6865 2073 7461  consider the sta
000001f0: 7273 2074 6f20 7368 696e 6520 6d65 7265  rs to shine mere
00000200: 6c79 2074 6f20 7370 616e 676c 6520 7468  ly to spangle th
00000210: 6520 6e69 6768 7420 2764 7573 7479 2077  e night 'dusty w
00000220: 6974 7327 2c20 666f 7220 746f 2068 696d  its', for to him
00000230: 2074 6865 6972 2069 6d70 6f72 7461 6e63   their importanc
00000240: 6520 7761 7320 6d75 6368 2067 7265 6174  e was much great
00000250: 6572 2e20 0d0a 5468 6579 2077 6572 6520  er. ..They were
00000260: 616e 2069 6d70 6f72 7461 6e63 6520 696e  an importance in
00000270: 666c 7565 6e63 6520 696e 2068 756d 616e  fluence in human
00000280: 206c 6976 6573 2e20 0d0a 416c 7468 6f75   lives. ..Althou
00000290: 6768 2068 6973 2073 6f6e 6e65 742c 206c  gh his sonnet, l
000002a0: 696b 6520 7468 6973 206f 6e65 2c20 6279  ike this one, by
000002b0: 2069 7473 2063 6f6e 636c 7573 696f 6e20   its conclusion
000002c0: 6973 2073 6f6d 6577 6861 7420 746f 6e67  is somewhat tong
000002d0: 7565 2069 6e20 6368 6565 6b2e 200d 0a28  ue in cheek. ..(
000002e0: 4e6f 7465 2074 6861 7420 5369 646e 6579  Note that Sidney
```



```
000002e0: 4e6f 7465 2074 6861 7420 5369 646e 6579  Note that Sidney
000002f0: 2075 7365 7320 7468 6520 7465 726d 2061   uses the term a
00000300: 7374 726f 6c6f 6779 2e20 4865 2061 6c73  strology. He als
00000310: 6f20 7265 6164 7320 5374 656c 6c61 7327  o reads Stellas'
00000320: 7320 6579 6573 2061 7320 6966 2074 6865  s eyes as if the
00000330: 7920 7765 7265 2073 7461 7273 292e 200d  y were stars). .
00000340: 0a54 6865 2070 6f65 7420 6865 7265 2063  .The poet here c
00000350: 6c61 696d 7320 746f 2027 6861 7665 2041  laims to 'have A
00000360: 7374 726f 6e6f 6d79 272c 2069 2e65 2068  stronomy', i.e h
00000370: 6520 756e 6465 7273 7461 6e64 7320 6974  e understands it
00000380: 2061 7320 6120 7363 6965 6e63 652c 2061   as a science, a
00000390: 6e64 2074 6865 6e20 6865 2070 726f 6365  nd then he proce
000003a0: 6564 7320 746f 2074 656c 6c20 7573 2068  eds to tell us h
000003b0: 6f77 2068 6973 206b 6e6f 776c 6564 6765  ow his knowledge
000003c0: 2064 6966 6665 7273 2066 726f 6d20 7468   differs from th
000003d0: 6174 206f 6620 7468 6520 7472 6164 6974  at of the tradit
000003e0: 696f 6e61 6c20 6173 7472 6f6c 6f67 6572  ional astrologer
000003f0: 2028 6c69 6e65 7320 332d 3829 2e0d 0a57   (lines 3-8)...W
00000400: 6520 7465 6e64 2074 6f20 7468 696e 6b20  e tend to think
00000410: 6f66 206f 7572 7365 6c76 6573 2061 7320  of ourselves as
00000420: 6120 6d6f 7265 2072 6174 696f 6e61 6c20  a more rational
00000430: 6167 652c 2062 7574 2061 2072 6563 656e  age, but a recen
00000440: 7420 7072 6573 6964 656e 7420 6f66 2074  t president of t
00000450: 6865 2055 6e69 7465 6420 5374 6174 6573  he United States
00000460: 2c20 526f 6e61 6c64 2052 6561 6761 6e2c  , Ronald Reagan,
00000470: 2072 656c 6965 6420 6f6e 2068 6973 2077   relied on his w
00000480: 6966 6527 7320 6173 7472 6f6c 6f67 6572  ife's astrologer
00000490: 2074 6f20 666f 7265 6361 7374 2066 6f72   to forecast for
000004a0: 2068 696d 2070 726f 7069 7469 6f75 7320   him propitious
000004b0: 6461 7973 2066 6f72 2077 6f72 6b20 616e  days for work an
000004c0: 6420 706f 6c69 6379 2064 6563 6973 696f  d policy decisio
000004d0: 6e73 2e0d 0a                             ns...
root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# make dec-diff
*Astronomy* in Elizabethan times was much closer to what we w   *Astronomy* in Elizabethan times was much closer to what we w
It was not yet weighted down with knowledge of what the plane   It was not yet weighted down with knowledge of what the plane
There was a widespread belief that the stars, in their variou   There was a widespread belief that the stars, in their variou
See the sonnet by Sidney, given at the bottom of the page.      See the sonnet by Sidney, given at the bottom of the page.
He calls those who consider the stars to shine merely to span   He calls those who consider the stars to shine merely to span
They were an importance influence in human lives.               They were an importance influence in human lives.
Although his sonnet, like this one, by its conclusion is some   Although his sonnet, like this one, by its conclusion is some
(Note that Sidney uses the term astrology. He also reads Stel   (Note that Sidney uses the term astrology. He also reads Stel
The poet here claims to 'have Astronomy', i.e he understands    The poet here claims to 'have Astronomy', i.e he understands
We tend to think of ourselves as a more rational age, but a r   We tend to think of ourselves as a more rational age, but a r

root@LAPTOP-QTCGESHO:/mnt/d/blog/work/信息安全/001# 
```

可以发现，两者结果相同