**实验报告** SA16011095 郑晓杰

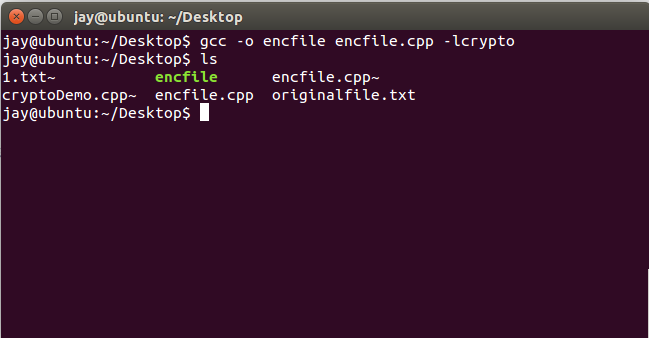
**实验目的**：修改例程cryptoDemo.cpp为encfile.cpp ，从命令行接受参数3个字符串类型的参数：参数1，参数2， 参数3。参数1=enc表示加密，参数1=dec表示解密；参数2为待 加密、解密的文件名；参数3为密码。

**实验环境**：vmware虚拟机 Ubuntu14.04

**实验过程**：

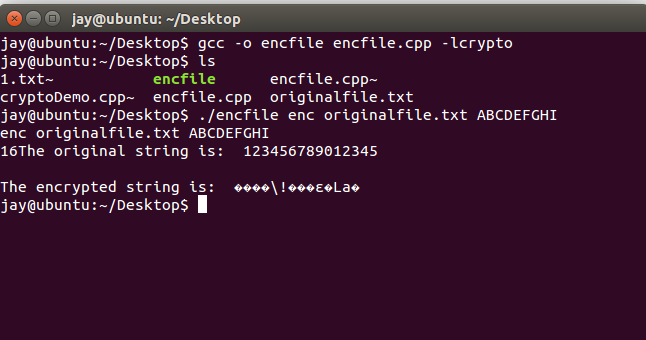
(1).修改代码，使其能够接受命令行中的三个参数，来执行相应的操作修改之后代码见附录

(2).编译代码，生成可执行文件encfile：

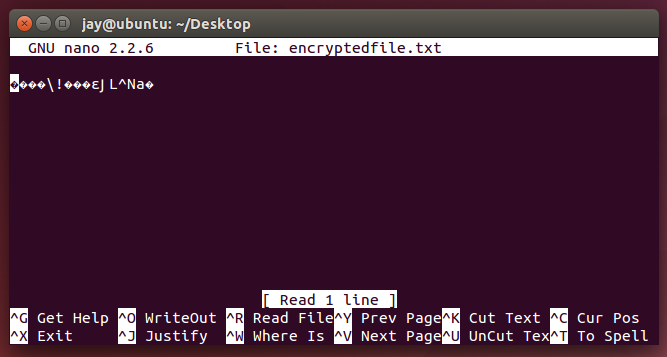


(3).开始执行：

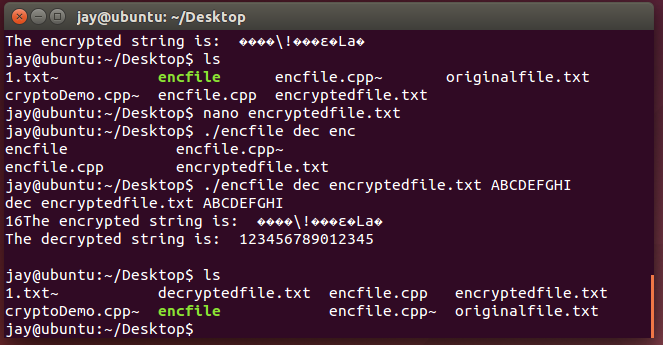
a.加密文件originalfile.txt 里面内容为123456789012345 密码为ABCDEFGHI



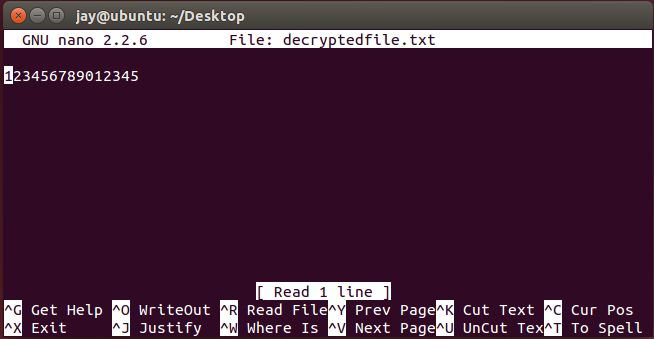
b.发现加密成功。生成一个encryptedfile.txt



c.开始解密encryptedfile.txt 密码为ABCDEFGHI



d．发现成功生成了解密之后的文件decryptedfile.txt,检查发现和originalfile.txt一致，实验取得成功。



**实验总结**：通过这次实验，使用到了信息安全课程上学习到的密码学的知识，成功的实现了使用openssl对文件进行加密和解密的过程，将知识用于实践，也通过实践来巩固知识，受益匪浅。

附录：encfile.cpp

// cryptoDemo.cpp : Defines the entry point for the console application.

// Windows: cl cryptoDemo.cpp

// Linux: gcc -o cryptoDemo cryptoDemo.cpp -lcrypto

#include <memory.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include "openssl/aes.h"

#pragma comment(lib,"libeay32.lib")

char\* readfile(char\* filename){

FILE \* pFile;

long lSize;

char \* buffer;

size\_t result;

/\* 若要一个byte不漏地读入整个文件，只能采用二进制方式打开 \*/

pFile = fopen (filename, "rb" );

if (pFile==NULL)

{

fputs ("File error",stderr);

exit (1);

}

/\* 获取文件大小 \*/

fseek (pFile , 0 , SEEK\_END);

lSize = ftell (pFile);

rewind (pFile);

/\* 分配内存存储整个文件 \*/

buffer = (char\*) malloc (sizeof(char)\*lSize);

if (buffer == NULL)

{

fputs ("Memory error",stderr);

exit (2);

}

/\* 将文件拷贝到buffer中 \*/

result = fread (buffer,1,lSize,pFile);

if (result != lSize)

{

fputs ("Reading error",stderr);

exit (3);

}

printf("%d",(int)lSize);

fclose (pFile);

return buffer;

}

void writefile(const char\* filename,char\* buffer){

FILE \* pFile;

pFile = fopen (filename, "wb" );

if (pFile==NULL)

{

fputs ("File error",stderr);

exit (1);

}

fwrite(buffer,strlen(buffer),1,pFile);

fclose (pFile);

return ;

}

void testAes(char\* way,char inString[], int inLen, char passwd[], int pwdLen)

{

int i,j, len, nLoop, nRes;

char enString[1024];

char deString[1024];

unsigned char buf[16];

unsigned char buf2[16];

unsigned char aes\_keybuf[32];

AES\_KEY aeskey;

if ( !strcmp(way,"enc") ) {

memset(aes\_keybuf,0x90,32);

if(pwdLen<32){ len=pwdLen; } else { len=32;}

for(i=0;i<len;i++) aes\_keybuf[i]=passwd[i];

nLoop=inLen/16; nRes = inLen%16;

AES\_set\_encrypt\_key(aes\_keybuf,256,&aeskey);

for(i=0;i<nLoop;i++){

memset(buf,0,16);

for(j=0;j<16;j++) buf[j]=inString[i\*16+j];

AES\_encrypt(buf,buf2,&aeskey);

for(j=0;j<16;j++) enString[i\*16+j]=buf2[j];

}

if(nRes>0){

memset(buf,0,16);

for(j=0;j<nRes;j++) buf[j]=inString[i\*16+j];

AES\_encrypt(buf,buf2,&aeskey);

for(j=0;j<16;j++) enString[i\*16+j]=buf2[j];

//puts("encrypt");

}

enString[i\*16+j]=0;

} else if (!strcmp(way,"dec")) {

memset(aes\_keybuf,0x90,32);

if(pwdLen<32){ len=pwdLen; } else { len=32;}

for(i=0;i<len;i++) aes\_keybuf[i]=passwd[i];

nLoop=inLen/16; nRes = inLen%16;

strcpy(enString,inString);

enString[i\*16+j]=0;

AES\_set\_decrypt\_key(aes\_keybuf,256,&aeskey);

for(i=0;i<nLoop;i++){

memset(buf,0,16);

for(j=0;j<16;j++) buf[j]=enString[i\*16+j];

AES\_decrypt(buf,buf2,&aeskey);

for(j=0;j<16;j++) deString[i\*16+j]=buf2[j];

}

if(nRes>0){

memset(buf,0,16);

for(j=0;j<16;j++) buf[j]=enString[i\*16+j];

AES\_decrypt(buf,buf2,&aeskey);

for(j=0;j<16;j++) deString[i\*16+j]=buf2[j];

//puts("decrypt");

}

deString[i\*16+nRes]=0;

}

if (!strcmp(way,"enc")){

printf("The original string is: %s \n", inString);

printf("The encrypted string is: %s \n", enString);

writefile("encryptedfile.txt",enString);

} else if (!strcmp(way,"dec")) {

printf("The encrypted string is: %s \n", enString);

printf("The decrypted string is: %s \n", deString);

writefile("decryptedfile.txt",deString);

}

}

int main(int argc, char\* argv[])

{

char \*way=argv[1];

char \*inputfile=argv[2];

char \*passwd = argv[3];

printf("%s %s %s \n",way,argv[2],argv[3]);

char \*inString=readfile(inputfile);

testAes(way,inString, strlen(inString), passwd, strlen(passwd));

return 0;

}