实验2 — 数字信封

姓名:陈扬

学号:17150011001

专业:17 政治学与行政学

课程:现代密码学 理论与实践

教师:林喜军

实验目的

通过实际编程了解数字信封的原理

实验要求

首先，学习实例代码 然后，利用数字信封原理，将VC6工程代码中的明文字符串input进行加密，将运算结果显示在屏幕上(以十六进制形式) 。然后将密文解密，并将解密结果(以字符串形式)显示在屏幕上。

实验步骤如下：

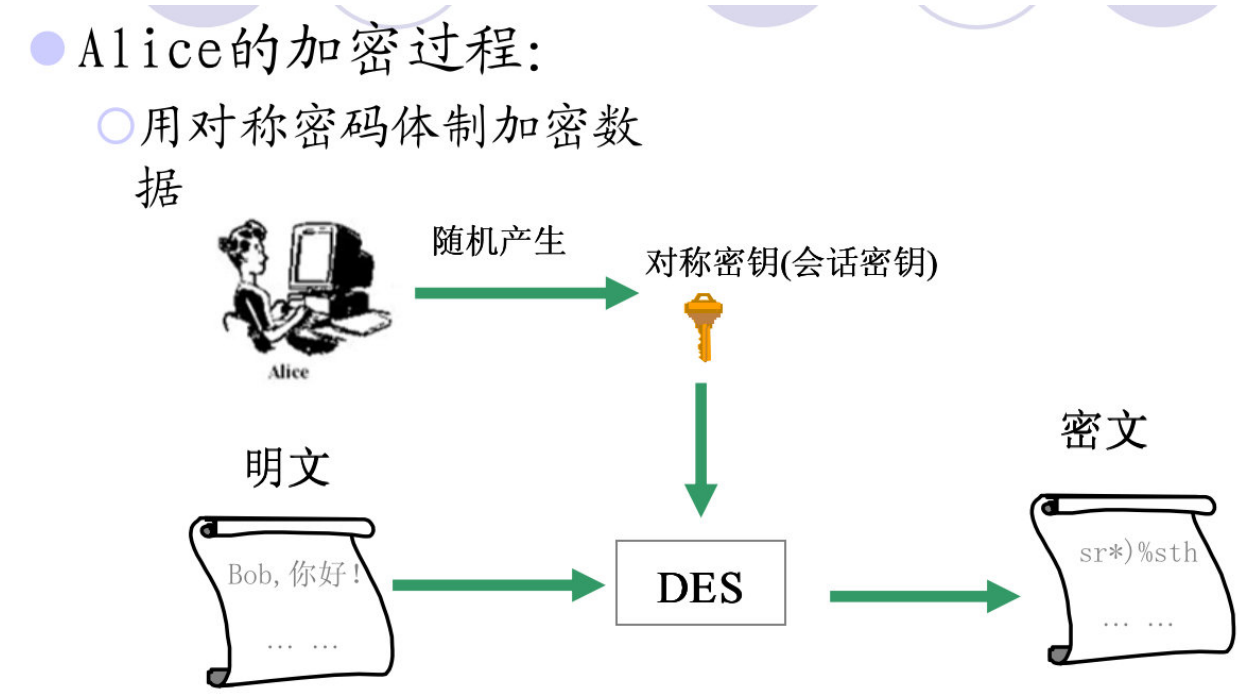
产生1024比特的随机RSA密钥(包括公钥和私钥，要求e=65537)

数字信封的封装(加密)

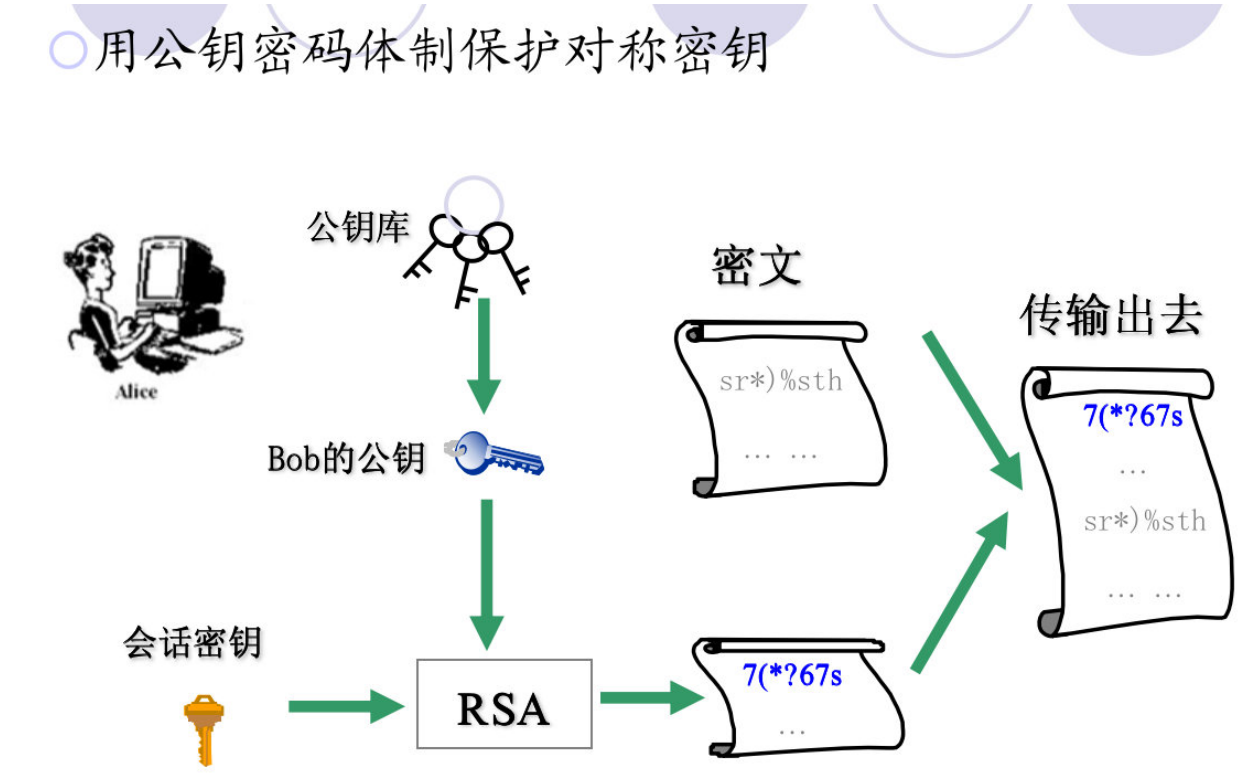
(1) 产生随机对称会话密钥

(2) 用公钥加密该会话密钥

(3) 用会话密钥，采用DES-CBC加密明文



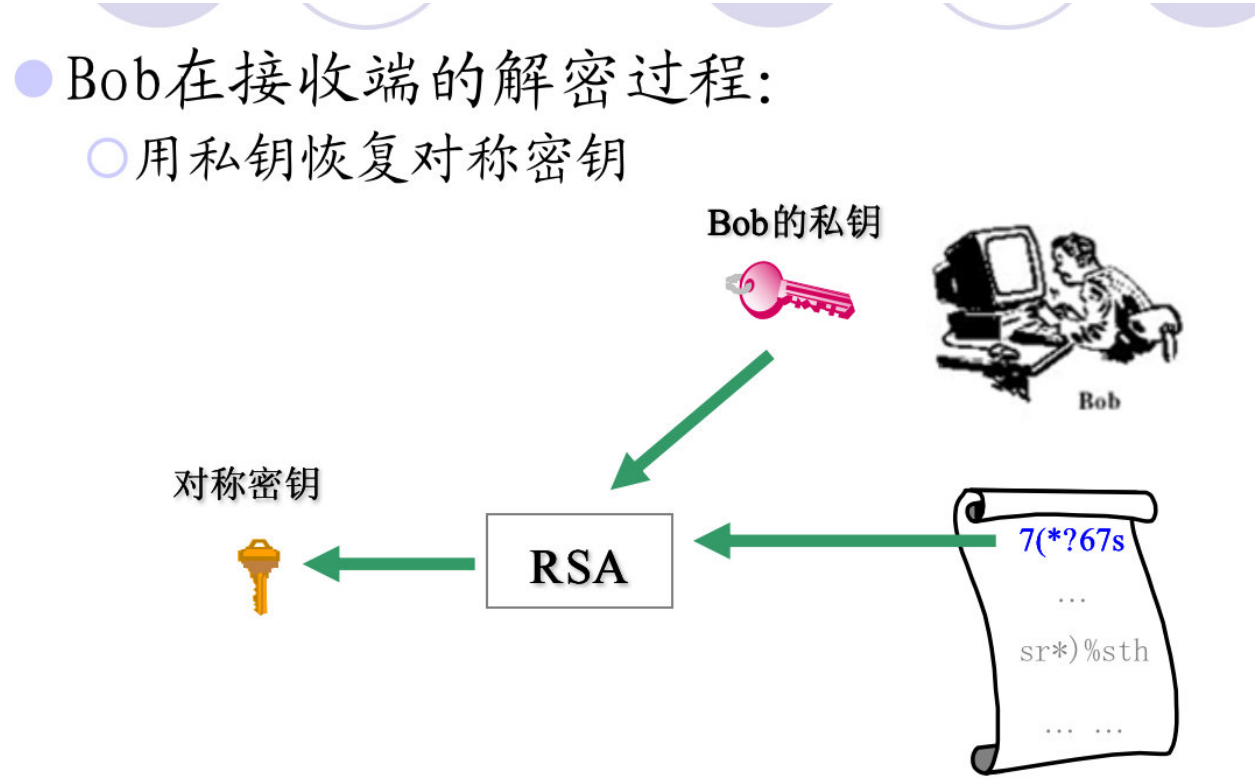
显示封装结果



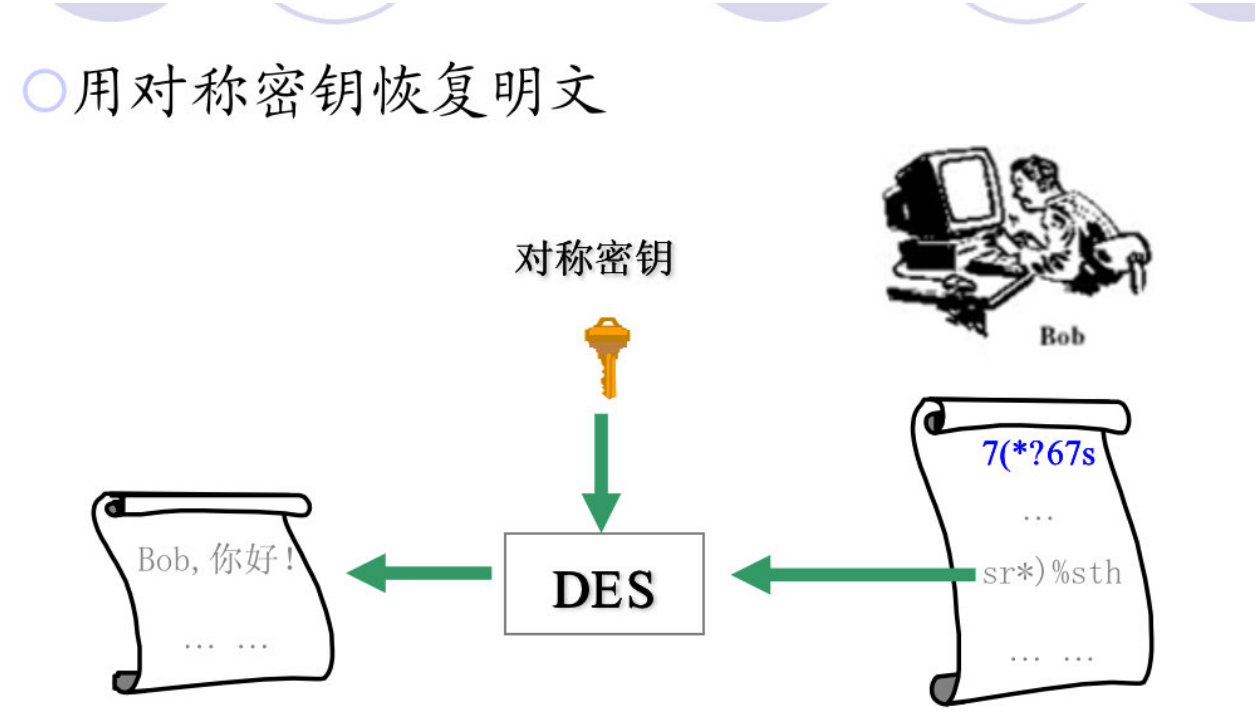
显示密文、加密后的对称密钥

数字信封的解封(解密)

(1) 用私钥解密出会话密钥



(2) 用会话密钥，采用DES-CBC解密密文



以字符串形式显示恢复出的明文

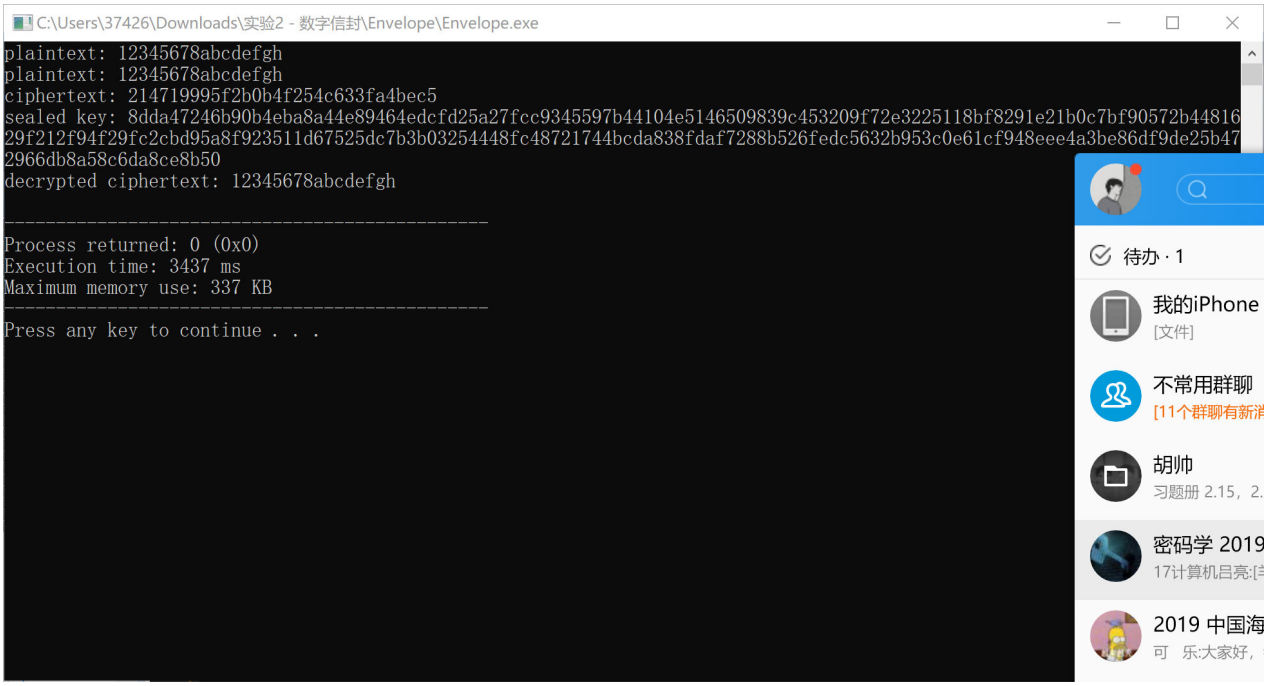
实验代码:

```cpp
// Envelope.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

#include <string.h>
#include <stdlib.h>
#include "R_STDLIB.C"
#include "R_RANDOM.C"
#include "NN.C"
#include "RSA.C"
#include "DIGIT.C"
#include "MD5C.C"
#include "PRIME.C"
#include "R_KEYGEN.C"
#include "DESC.C"

#define TEXT_LEN 16 //明密文长度
#define DES_LEN 8
//填充随机数变量
void seed_randomStruct(unsigned char *seed, R_RANDOM_STRUCT
*randomStruct)
{
    unsigned int bytesNeeded = 256; //结构体所需种子长度

    R_RandomInit(randomStruct);
    while (bytesNeeded > 0)
    {
        R_RandomUpdate(randomStruct, seed,
                       strlen((char *)seed));
        R_GetRandomBytesNeeded(&bytesNeeded, randomStruct);
    }
}

// 以十六进制形式显示output中的内容(参数len表示output的长度)
void shows(unsigned char *output, unsigned int len)
{
    for (unsigned int i = 0; i < len; i++)
        printf("%x", output[i]);
    printf("\n");
}

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

    R_RANDOM_STRUCT randomStruct;                              //保存随机数
    unsigned char seed[] = "3adqwe1212asd";                    // 种子
    unsigned char iv[8 + 1] = "13wedfgr";                      // IV
    unsigned char input[TEXT_LEN + 1] = "12345678abcdefgh"; // 明文
```

```
49
50      //seed_randomStruct (seed, &randomStruct);  // 用种子填充随机数变量
51
52      printf("plaintext: %s\n", input); // 显示明文
53
54      //*****请在下面每一步后面填写你的代码**************
55
56      //步骤1：产生随机RSA密钥（包括公钥和私钥）
57      unsigned char output[MAX_ENCRYPTED_KEY_LEN] = "";
58      unsigned char SK[MAX_ENCRYPTED_KEY_LEN] = "";
59      unsigned char output2[TEXT_LEN + 1] = "";
60      unsigned int outputlen, outputlen2, SK_LEN, des_len;
61      int flag;
62
63      R_RSA_PUBLIC_KEY publicKey;
64      R_RSA_PRIVATE_KEY privateKey;
65      R_RSA_PROTO_KEY protoKey;
66
67      protoKey.bits = 1024;
           //设定模数长度为1024
68      protoKey.useFermat4 = 1;
            //设定e=65537
69      seed_randomStruct(seed, &randomStruct);
           // 填充随机数结构体
70      flag = R_GeneratePEMKeys(&publicKey, &privateKey, &protoKey,
&randomStruct); // 产生随机RSA密钥
71      if (RE_MODULUS_LEN == flag)
72      {
73          printf("modulus length invalid\n");
74          exit(0);
75      }
76      else if (RE_NEED_RANDOM == flag)
77      {
78          printf("randomStruct is not seeded\n");
79          exit(0);
80      }
81      // 显示明文
82      printf("plaintext: %s\n", input);
83      // 显示密文
84
85      //步骤2：  数字信封的封装(加密)
86      // (1) 产生随机对称会话密钥
87      DES_CBC_CTX context;
88      //明文串input、密文串output、解密后的明文串output2
89      //密钥key,初始向量iv
90      unsigned char key[DES_LEN + 1] = "";
91      unsigned char key2[DES_LEN + 1] = "";
92      R_GenerateBytes(key, DES_LEN, &randomStruct);
93      //  shows(key,DES_LEN);
```

```
94
95      // (2) 用公钥加密该会话密钥
96      // 加密
97      RSAPublicEncrypt(SK, &SK_LEN, key, strlen((char *)key),
98                       &publicKey, &randomStruct);
99
100     // (3) 用会话密钥，采用DES-CBC加密明文（初始向量iv定义如上）
101
102     DES_CBCInit(&context, key, iv, 1);
103     DES_CBCUpdate(&context, output, input, TEXT_LEN);
104
105     //deleted key
106     //key 是随机生成的 DES 密钥,key2 是通过 RSA 解密得到的 DES 解密密钥
107     //步骤3：显示封装结果
108     // 调用函数shows显示密文
109     printf("ciphertext: ");
110     shows(output, TEXT_LEN);
111     // 调用函数shows显示加密后的对称密钥
112     printf("sealed key: ");
113     shows(SK, SK_LEN);
114
115     //步骤4： 数字信封的解封(解密)
116     //(1) 用私钥解密出会话密钥
117     RSAPrivateDecrypt(key2, &des_len, SK, SK_LEN,
118                       &privateKey);
119     //    shows(key,des_len);
120     //(2) 用会话密钥，采用DES-CBC解密密文（需使用同样的初始向量iv）
121     DES_CBCInit(&context, key2, iv, 0);
122     DES_CBCUpdate(&context, output2, output, TEXT_LEN);
123
124     //步骤5：以字符串形式显示恢复出的明文
125     printf("decrypted ciphertext: %s\n", output2);
126     //从内存中擦出随机数结构体中的信息
127     R_RandomFinal(&randomStruct);
128
129     return 0;
130 }
```

结果展示


C:\Users\37426\Downloads\实验2 - 数字信封\Envelope\Envelope.exe
plaintext: 12345678abcdefgh
plaintext: 12345678abcdefgh
ciphertext: 214719995f2b0b4f254c633fa4bec5
sealed key: 8dda47246b90b4eba8a44e89464edcfd25a27fcc9345597b44104e5146509839c453209f72e3225118bf8291e21b0c7bf90572b44816
29f212f94f29fc2cbd95a8f923511d67525dc7b3b03254448fc48721744bcda838fdaf7288b526fedc5632b953c0e61cf948eee4a3be86df9de25b47
2966db8a58c6da8ce8b50
decrypted ciphertext: 12345678abcdefgh
Process returned: 0 (0x0)
Execution time: 3437 ms
Maximum memory use: 337 KB
Press any key to continue . . .
待办 · 1
我的iPhone
[文件]
不常用群聊
胡帅
习题册 2.15，2.
密码学 2019
2019 中国海
可 乐:大家好，