(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106529352 A
(43)申请公布日 2017.03.22

(21)申请号 201610899911.1

(22)申请日 2016.10.15

(71)申请人 北海益生源农贸有限责任公司
地址 536000 广西壮族自治区北海市海城区金海岸大道45号北部湾科技创业中心3幢0801号

(72)发明人 胡克荣

(74)专利代理机构 北海市佳旺专利代理事务所（普通合伙）45115
代理人 黄建中

(51)Int.Cl.
G06F 21/83(2013.01)
G06F 21/60(2013.01)

权利要求书2页 说明书7页

(54)发明名称

一种对计算机客户端信息安全输入的方法

(57)摘要

一种对计算机客户端信息安全输入的方法，包括计算机客户端，具体实现步骤为：(1)在计算机客户端上用户选择；(2)计算机客户端的键盘的按键值顺序布局或者乱序布局；(3)用户输入信息；(4)用户发送信息；(5)DES加密；(6)网络传输；(7)计算机服务器接受信息；(8)DES解密；(9)计算机服务器校验信息。本发明实现计算机用户信息输入的信息更加安全；可以应用各个行业中行业，是一种十分安全的用户信息输入；本发明提供了一种现实可行的解决方案，具有很好的推广使用价值。

CN 106529352 A

1.一种对计算机客户端信息安全输入的方法，包括计算机客户端，其特征在于，该方法的具体实现步骤为：(1)在计算机客户端上用户选择；(2)计算机客户端的键盘的按键值顺序布局或者乱序布局；(3)用户输入信息；(4)用户发送信息；(5)DES加密；(6)网络传输；(7)计算机服务器接受信息；(8)DES解密；(9)计算机服务器校验信息。

2.根据权利要求1中所述的一种对计算机客户端信息安全输入的方法，其特征在于，所述步骤(2)为键盘的按键值的位置每次在键盘上的排列顺序都是随机的或者顺序的，具体实现方法是：首先定义一个数组，根据电路板原理和数码管的显示数据，定义0到9和A到Z的显示数据编码，然后再定义两个数组，分别用于存放键盘值和数码管的显示编码；然后用随机函数rand()和srand()对以上数据进行从新排序；每一次随机的是第一个数据的下标；第一次产生是从0-9之间进行产生，这样产生的数据和最后一位进行交换，对产生的随机数据序列对应数码管的编码，就是最终的显示数据；最后根据具体显示的那个数据去对照数码管的编码，实现数据的显示，同时根据显示的数据在键盘值数组对应的位置设置对应的值。

3.根据权利要求1中所述的一种对计算机客户端信息安全输入的方法，其特征在于，所述DES加密由加密处理，加密变换和子密钥生成组成。

4.根据权利要求3中所述的一种对计算机客户端信息安全输入的方法，其特征在于，所述加密处理具体实现方法为：(1)首先对64位的明文按初始换位表IP进行变换，例如输入的第58位，在输出的时候被置换到第1位；输入的是第7位，在输出时被置换到第64位；

(2)对上述换位处理的输出经过16轮加密变换的加密处理，初始换位的64位的输出作为下一次的输入，将64位分为左、右两个32位，分别记为L0和R0，从L0、R0到L16、R16，共进行16轮加密变换；其中，经过n轮处理后的点左右32位分别为Ln和Rn，可做如下定义：

Ln＝Rn-1

Rn＝Ln-1

其中，kn是向第n轮输入的48位的子密钥，Ln-1和Rn-1分别是第n-1轮的输出，f是Mangler函数；

(3)进行16轮的加密变换之后，将L16和R16合成64位的数据，再进行IP-1的换位，得到64位的密文。

5.根据权利要求3中所述的一种对计算机客户端信息安全输入的方法，其特征在于，所述加密变换具体为：通过重复某些位将32位的右半部分按照扩展表3扩展换位表扩展为48位，而56位的密钥先移位然后通过选择其中的某些位减少至48位，48位的右半部分通过异或操作和48位的密钥结合，并分成6位的8个分组，通过8个S盒将这48位替代成新的32位数据，再将其置换一次。

6.根据权利要求3中所述的一种对计算机客户端信息安全输入的方法，其特征在于，所述子密钥生成具体为输入的64位密钥，通过压缩换位PC-1去掉每个字节的第8位，用作奇偶校验，减至密钥长度为56位，每层分成两部分，上部分28位为C0，下部分为D0，C0和D0依次进行循环左移操作生成了C1和D1，将C1和D1合成56位，再通过压缩换位PC-2输出48位的子密钥K1，再将C1和D1进行循环左移和PC-2压缩换位，得到子密钥K2.....以此类推，得到16个子密钥。

7.根据权利要求5中所述的一种对计算机客户端信息安全输入的方法，其特征在于，所

述S盒输入6位,输出4位,一个S盒中具有4种替换表(行号用0、1、2、3表示),通过输入的6位的开头和末尾两位选定行,然后按选定的替换表将输入的6位的中间4位进行替代。

8.根据权利要求1中所述的一种对计算机客户端信息安全输入的方法,其特征在于,所述DES解密具体为:数据传输到目标机之后再根据密钥key进行解密,DES解密算法和DES加密算法相同,密钥倒序即可,最后得到正确的输入信息。

一种对计算机客户端信息安全输入的方法

技术领域

[0001] 本发明涉及计算机客户端技术领域，具体涉及一种对计算机客户端信息安全输入的方法。

背景技术

[0002] 随着科技的发展，计算机在很多领域都得到广泛应用，由于目前计算机输入涉及到用户的隐私信息，在各种复杂的环境中无法保证用户信息的安全性；不法分子可以通过多种方式在用户输入信息和传输信息的过程中窃取用户的信息。因此，信息加密，尤其是一种对计算机输入信息的加密保护研究具有非常重要的意义。

发明内容

[0003] 本发明所要解决的技术问题是提供一种对计算机客户端信息安全输入的方法。

[0004] 为实现本发明的目的，本发明所采用的技术方案是：

[0005] 一种对计算机客户端信息安全输入的方法，包括计算机客户端，具体实现步骤为：(1)在计算机客户端上用户选择；(2)计算机客户端的键盘的按键值顺序布局或者乱序布局；(3)用户输入信息；(4)用户发送信息；(5)DES加密；(6)网络传输；(7)计算机服务器接受信息；(8)DES解密；(9)计算机服务器校验信息。

[0006] 进一步的，所述步骤(2)为键盘的按键值的位置每次在键盘上的排列顺序都是随机的或者顺序的，具体实现方法是：首先定义一个数组，根据电路板原理和数码管的显示数据，定义0到9和A到Z的显示数据编码，然后再定义两个数组，分别用于存放键盘值和数码管的显示编码；然后用随机函数rand()和srand()对以上数据进行从新排序；每一次随机的是第一个数据的下标；第一次产生是从0-9之间进行产生，这样产生的数据和最后一位进行交换，对产生的随机数据序列对应数码管的编码，就是最终的显示数据；最后根据具体显示的那个数据去对照数码管的编码，实现数据的显示，同时根据显示的数据在键盘值数组对应的位置设置对应的值。

[0007] 进一步的，所述DES加密由加密处理，加密变换和子密钥生成组成。

[0008] 进一步的，所述加密处理具体实现方法为：(1)首先对64位的明文按初始换位表IP进行变换，例如输入的第58位，在输出的时候被置换到第1位；输入的是第7位，在输出时被置换到第64位；(2)对上述换位处理的输出经过16轮加密变换的加密处理，初始换位的64位的输出作为下一次的输入，将64位分为左、右两个32位，分别记为L0和R0，从L0、R0到L16、R16，共进行16轮加密变换；其中，经过n轮处理后的点左右32位分别为Ln和Rn，可做如下定义：

[0009] Ln＝Rn-1

[0010] Rn＝Ln-1

[0011] 其中，kn是向第n轮输入的48位的子密钥，Ln-1和Rn-1分别是第n-1轮的输出，f是Mangler函数；

[0012] (3)进行16轮的加密变换之后,将L16和R16合成64位的数据,再进行IP-1的换位,得到64位的密文。

[0013] 加密变换具体为:通过重复某些位将32位的右半部分按照扩展表3扩展换位表扩展为48位,而56位的密钥先移位然后通过选择其中的某些位减少至48位,48位的右半部分通过异或操作和48位的密钥结合,并分成6位的8个分组,通过8个S盒将这48位替代成新的32位数据,再将其置换一次。

[0014] 子密钥生成具体为输入的64位密钥,通过压缩换位PC-1去掉每个字节的第8位,用作奇偶校验,减至密钥长度为56位,每层分成两部分,上部分28位为C0,下部分为D0,C0和D0依次进行循环左移操作生成了C1和D1,将C1和D1合成56位,再通过压缩换位PC-2输出48位的子密钥K1,再将C1和D1进行循环左移和PC-2压缩换位,得到子密钥K2.....以此类推,得到16个子密钥。

[0015] 进一步的,所述S盒输入6位,输出4位,一个S盒中具有4种替换表(行号用0、1、2、3表示),通过输入的6位的开头和末尾两位选定行,然后按选定的替换表将输入的6位的中间4位进行替代。

[0016] 所述DES解密具体为:数据传输到目标机之后再根据密钥key进行解密,DES解密算法和DES加密算法相同,密钥倒序即可,最后得到正确的输入信息。

[0017] DES算法加密的关键代码为:

```
using System;
using System.Security.Cryptography;
using System.IO;
using System.Text;
public class EncryptStringDES {
public static void Main(String[] args) {
```

[0018]

```
if (args.Length 1) {
Console.WriteLine("Usage: des_demo
encrypt", args[0]);
 return;
     }
// 使用 UTF8 函数加密输入参数
 UTF8Encoding utf8Encoding = new UTF8Encoding();
```

```
 byte[] inputByteArray =
utf8Encoding.GetBytes(args[0].ToCharArray());
//使用 DES_CSP()实现 DES 的实体
//DES_CSP DES = new DES_CSP();
// 初始化 DES 加密的密钥和一个随机的、8 比特的初始化向量(IV)
Byte[] key = {0x01, 0x23, 0x45, 0x67, 0x89, 0xab,0xcd, 0xef};
Byte[] IV = {0x12, 0x34, 0x56, 0x78, 0x90, 0xab,0xcd, 0xef};
des.Key = key;
des.IV = IV;
// 建立加密流
SymmetricStreamEncryptor sse = des.CreateEncryptor();
// 使用 CryptoMemoryStream 方法获取加密过程的输出
```

[0019]

```
CryptoMemoryStream cms = new CryptoMemoryStream();
// 将 SymmetricStreamEncryptor 流中的加密数据输出到
CryptoMemoryStream 中
sse.SetSink(cms);
// 加密完毕,将结果输出到控制台
sse.Write(inputByteArray);
sse.CloseStream();
// 获取加密数据
 byte[] encryptedData = cms.Data;
// 输出加密后结果
Console.WriteLine("加密结果:");
for (int i = 0; i encryptedData.Length; i++) {
Console.Write("{0:X2} ",encryptedData[i]);
```

```
        }
        Console.WriteLine();
```

加密的关键代码为：

```
SymmetricStreamDecryptor ssd = des.CreateDecryptor();
cms = new CryptoMemoryStream();
ssd.SetSink(cms);
ssd.Write(encryptedData);
ssd.CloseStream();
byte[] decryptedData = cms.Data;
char[] decryptedCharArray =
utf8Encoding.GetChars(decryptedData);
Console.WriteLine("解密后数据:");
Console.Write(decryptedCharArray);
Console.WriteLine(); }
    }。
```

[0021] 本发明提供的一种对计算机客户端信息安全输入的方法，本发明采用键盘顺序布局或乱序布局、DES加密方法和新的设计理念，来实现计算机用户信息输入的信息更加安全；可以应用各个行业中行业，是一种十分安全的用户信息输入；本发明提供了一种现实可行的解决方案，具有很好的推广使用价值。

具体实施方式

[0022] 下列对本发明的实施作进一步的说明，以便本领域的技术人员能够更好的理解并实施。

[0023] 实施例1

[0024] 一种对计算机客户端信息安全输入的方法，具体实现步骤为：(1)在计算机客户端上用户选择；(2)计算机客户端的键盘的按键值顺序布局或者乱序布局；(3)用户输入信息；(4)用户发送信息；(5)DES加密；(6)网络传输；(7)计算机服务器接受信息；(8)DES解密；(9)计算机服务器校验信息。

[0025] 步骤(2)为键盘的按键值的位置每次在键盘上的排列顺序都是随机的或者顺序的：首先定义一个数组，根据电路板原理和数码管的显示数据，定义0到9和A到Z的显示数据编码，然后再定义两个数组，分别用于存放键盘值和数码管的显示编码；然后用随机函数

rand()和srand()对以上数据进行从新排序;由于本身芯片的资源和速度以及本身产品要求显示的实时性,在这个排序过程中,对产生随机数进行了改造,每一次随机的是第一个数据的下标;第一次产生是从0-9之间进行产生,这样产生的数据和最后一位进行交换,这样每遍历一次就会少一个值,这样就不会产生重复数据的机会,也就保证了速度。对产生的随机数据序列对应数码管的编码,就是最终的显示数据;最后根据具体显示的那个数据去对照数码管的编码,实现数据的显示,同时根据显示的数据在键盘值数组对应的位置设置对应的值,以便在用户输入信息能够正确发送出去。

[0026] DES加密(DES(key1,加密)DES(key2,解密)DES(key1,加密))由加密处理,加密变换和子密钥生成组成。加密处理具体实现方法为:(1)首先对64位的明文按初始换位表IP进行变换,例如输入的第58位,在输出的时候被置换到第1位;输入的是第7位,在输出时被置换到第64位;(2)对上述换位处理的输出经过16轮加密变换的加密处理,初始换位的64位的输出作为下一次的输入,将64位分为左、右两个32位,分别记为L0和R0,从L0、R0到L16、R16,共进行16轮加密变换;其中,经过n轮处理后的点左右32位分别为Ln和Rn,可做如下定义:

[0027] Ln＝Rn-1

[0028] Rn＝Ln-1

[0029] 其中,kn是向第n轮输入的48位的子密钥,Ln-1和Rn-1分别是第n-1轮的输出,f是Mangler函数;

[0030] (3)进行16轮的加密变换之后,将L16和R16合成64位的数据,再进行IP-1的换位,得到64位的密文。

[0031] 加密变换具体为:通过重复某些位将32位的右半部分按照扩展表3扩展换位表扩展为48位,而56位的密钥先移位然后通过选择其中的某些位减少至48位,48位的右半部分通过异或操作和48位的密钥结合,并分成6位的8个分组,通过8个S盒将这48位替代成新的32位数据,再将其置换一次。S盒输入6位,输出4位,一个S盒中具有4种替换表(行号用0、1、2、3表示),通过输入的6位的开头和末尾两位选定行,然后按选定的替换表将输入的6位的中间4位进行替代。

[0032] 子密钥生成具体为输入的64位密钥,通过压缩换位PC-1去掉每个字节的第8位,用作奇偶校验,减至密钥长度为56位,每层分成两部分,上部分28位为C0,下部分为D0,C0和D0依次进行循环左移操作生成了C1和D1,将C1和D1合成56位,再通过压缩换位PC-2输出48位的子密钥K1,再将C1和D1进行循环左移和PC-2压缩换位,得到子密钥K2.....以此类推,得到16个子密钥。

[0033] DES解密具体为:数据传输到目标机之后再根据密钥key进行解密(DES(key1,解密)DES(key2,加密)DES(key1,解密)),DES解密算法和DES加密算法相同,密钥倒序即可,最后得到正确的输入信息。

[0034] DES算法加密的关键代码为:

```
using System;
using System.Security.Cryptography;
using System.IO;
using System.Text;
public class EncryptStringDES {
public static void Main(String[] args) {
if (args.Length 1) {
Console.WriteLine("Usage: des_demo
encrypt", args[0]);
 return;
     }
// 使用UTF8函数加密输入参数
```

[0035]

```
 UTF8Encoding utf8Encoding = new UTF8Encoding();
 byte[] inputByteArray =
utf8Encoding.GetBytes(args[0].ToCharArray());
//使用DES_CSP()实现DES的实体
//DES_CSP DES = new DES_CSP();
// 初始化DES加密的密钥和一个随机的、8比特的初始化向量(IV)
Byte[] key = {0x01, 0x23, 0x45, 0x67, 0x89, 0xab,0xcd, 0xef};
Byte[] IV = {0x12, 0x34, 0x56, 0x78, 0x90, 0xab,0xcd, 0xef};
des.Key = key;
des.IV = IV;
// 建立加密流
SymmetricStreamEncryptor sse = des.CreateEncryptor();
```

```
// 使用 CryptoMemoryStream 方法获取加密过程的输出
CryptoMemoryStream cms = new CryptoMemoryStream();
// 将 SymmetricStreamEncryptor 流中的加密数据输出到
CryptoMemoryStream 中
sse.SetSink(cms);
// 加密完毕,将结果输出到控制台
sse.Write(inputByteArray);
sse.CloseStream();
```

[0036]

```
// 获取加密数据
 byte[] encryptedData = cms.Data;
// 输出加密后结果
Console.WriteLine("加密结果:");
for (int i = 0; i encryptedData.Length; i++) {
Console.Write("{0:X2} ",encryptedData[i]);
 }
 Console.WriteLine();
```

[0037] 加密的关键代码为：

```
[0038] SymmetricStreamDecryptor ssd＝des.CreateDecryptor();
[0039] cms＝new CryptoMemoryStream();
[0040] ssd.SetSink(cms);
[0041] ssd.Write(encryptedData);
[0042] ssd.CloseStream();
[0043] byte[]decryptedData＝cms.Data;
[0044] char[]decryptedCharArray＝
[0045] utf8Encoding.GetChars(decryptedData);
[0046] Console.WriteLine("解密后数据:");
[0047] Console.Write(decryptedCharArray);
[0048] Console.WriteLine();}
[0049] }。
```