必修二18、数据加密

1、密码与密钥

明文（P）：原有的信息
密文（C）：明文经过加密变换后的形式
加密（E）：由明文变成密文的过程
解密（D）：由密文还原成明文的过程
C=Ek1(P)
P=Ek2(C)

2、替代加密法（凯撒密码）

```
#凯撒密码
def change(code,key):
    code=code.lower()
    m=ord(code)
    if m>97 and m<122:
        m=97+(m-97+key)%26
    return chr(m)
def encrypt(code,key):
    code_new=""
    for s in code:
        code_new+=change(s,key)
    return code_new
def decrypt(code,key):
    code_new=""
    for s in code:
        m=ord(s)-key
        if m<97:
            m+=26
        code_new+=chr(m)
    return code_new
p=input("请输入明文:")
key=int(input("请输入密码:"))
d=encrypt(p,key)
print("密文为:",d)
print("密文解密后:",decrypt(d,key))
```

```
请输入明文:hello
请输入密码:10
密文为: rovvy
密文解密后: hello
```

3、凯撒密码扩展举例

```python
#凯撒密码扩展
def encrypt(p,key):
    c=""
    for i in range(len(p)):
        if 'a'<=p[i]<='z' or 'A'<=p[i]<='Z':
            ik=i%(len(key))
            base=(ord(p[i])-65)//32*32+65   #当前大小写下第一个字母（a或者A）的ASCII
            c+=chr((ord(p[i])-base+int(key[ik]))%26+base)
        else:
            c+=p[i]
    return c
def decrypt(c,key):
    p=""
    for i in range(len(c)):
        if 'a'<=c[i]<='z' or 'A'<=c[i]<='Z':
            ik=i%(len(key))
            if ord(c[i])-int(key[ik])>=(ord(c[i])-65)//32*32+65:
                p+=chr(ord(c[i])-int(key[ik]))
            else:
                p+=chr(ord(c[i])-int(key[ik])+26)
        else:
            p+=c[i]
    return p
p=input("请输入明文:")
key=input("请输入密码:")
d=encrypt(p,key)
print("密文为:",d)
print("密文解密后:",decrypt(d,key))
```

```
请输入明文:abcdA
请输入密码:123
密文为: bdfeC
密文解密后: abcdA
```

4、异或加密

```python
def str2bin(s):
    bcode=''
    for c in s:
        bstr=bin(ord(c)).replace("0b","")
        bstr="0"*(8-len(bstr))+bstr
        bcode+=bstr
    return bcode
def xorencrypt(code,key):
    bincode=str2bin(code)
    binkey=str2bin(key)
    res=""
    for i in range(len(bincode)):
        bk=int(binkey[i%len(binkey)])
        if bincode[i]==binkey[bk]:
            res+="0"
        else:
            res+="1"
    return res
def xordecrypt(code,key):
    binkey=str2bin(key)
    res=""
    for i in range(len(code)):
        bk=int(binkey[i%len(binkey)])
        if code[i]==binkey[bk]:
            res+="0"
        else:
            res+="1"
    strcode=""
    for i in range(len(res)//8):
        strcode+=chr(int(res[i*8:i*8+8],2))
    return strcode
p=input("请输入明文:")
key=input("请输入密码:")
d=xorencrypt(p,key)
print("密文为:",d)
print("密文解密后:",xordecrypt(d,key))
```

```
请输入明文:hello james
请输入密码:joan
密文为: 000000100000101000001101000000100000010101001111000010110000111100000111000010100001001 0
密文解密后: hello james
```

5、对称密码体系与非对称密码体系

在上述密码算法中，加密密钥与解密密钥是一样的这样的密码体系称为对称密码体系。

试想：

为了保证用户向服务器提交的数据不被他人截获（如输入的用户名密码等），客户端可以先对这些敏感数据进行加密。但是也同时需要把密码发送给服务器。不然服务器无法解密

但是发送密码同样面临泄密问题，别人拿到密码和密文就可以进行解密。

如果加密密钥和解密密钥不同，就称为非对称密码体系

首先服务器生成一对公钥PK和私钥SK，并把公钥发送给客户端。客户端用公钥加密数据，然后传输给服务器端，服务器端用私钥解密。

在这个过程中，公钥只能实现加密，却不能解密，因此其它人就算知道公钥和密文，也无法得到明文。
RSA是著名的非对称加密算法