3.2.1数据加密和安全

**【任务清单】**

1. 通过加、解密实践与体验，理解常用的加密算法，初步学会用学过的程序设计语言进行数据加、解密程序的设计。(计算思维)
2. 过对数据的安全防范技术的学习，提升信息安全意识，能主动保护自身数据的安全和尊重他人的数据安全。理解软件在信息系统中的作用。（信息社会责任）

**【启学单】**

1.数据加密与安全

(1)口令:指进行信息系统登录、无线网络连接、ATM机取款等操作时输人的用于认证用户身份的\_\_\_\_\_\_\_\_\_\_\_.

(2)密码:通常是指按特定编码规则，对通信双方的数据信息进行从明文到密文变换的符号。

(3)密钥:指在密码算法中引进\_\_\_\_\_\_\_\_\_, 对一个算法采用不同的参数值，其解密结果就不同。分为加密密钥(Ke)和解密密钥(Kd)。

(4)密码系统包括明文、\_\_\_\_\_、密钥和\_\_\_\_\_\_\_\_四个方面。

(5)原有的信息称为明文(P)，明文经过加密变换后的形式称为密文(C)。

(6)由明文变为密文的过程称为加密(E),通常由\_\_\_\_\_\_\_\_\_\_\_\_\_\_来实现;由密文还原成明文的过程称为解密(D)，通常由解密算法来实现。

(7)常见的加密函数形式:C=Ex(P);常见的解密函数形式:P=Dx2(C)。

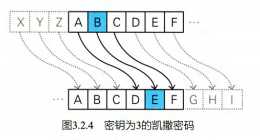
(8)三种简单加密算法:\_\_\_\_\_密码 换位密码，简单异或。

(9)若一种加密方法Ke=Kd, 则称为\_\_\_\_\_\_\_\_\_\_密码体制或单钥密码体制;若Ke≠Kd,则称为非对称密码体制或双钥密码体制。

**【厚学单】**

1. 替代密码，凯撒密码算法的实现。

凯撒密码是应用密钥加密数据的经典例子。凯撒密码是一种简单的置换密码，通过换位来实现加密。例如，密文字母表是由正常顺序的明文字母表右移3个字母得到的，如图3.2.4所示。加密过程可表示为Ci=Ek1(Pi)=(Pi+3) mod 26，这里的密钥为3。明文"hello world"由此可得到的密文是"khoor zruog”。解密时只要将密文中的每个字母左移3个字母即可得到明文，解密过程可表示为Pi=Dk1(Ci)=(Ci-3) mod 26。



1. 明文为”love”,密钥为5，则密文为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
2. 程序分为四个模块：字符转换change()，加密encrypt()，解密decrypt()，主函数调入。

①自定义函数change()，实现输入的字符串code ，根据密钥 key,转化为a~z之间的加密后的小写字母

思考：如何判断code为小写字母？如何将明文小写字母通过密码进行转换为密文字母？

def change(code,key):

code= code.lower() #将明文字符串code转化为小写字母

m = ord(code)

if \_\_\_\_\_\_\_\_(1)\_\_\_\_\_\_\_\_\_\_\_:

m =\_\_\_\_\_\_\_\_\_\_(2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

return \_\_\_\_\_\_\_(3)\_\_\_\_\_\_\_\_\_\_\_\_\_\_

②加密函数encrypt()和解密函数decryt()，根据输入的字符串code和key进行加密和解密。

|  |  |
| --- | --- |
| def encrypt(code,key):  code\_new =""  for s in \_\_\_(1)\_\_\_\_\_:  code\_new +=\_\_\_(2）\_\_\_\_\_\_\_\_  Print(code\_new)  return \_\_\_ （3）\_\_\_\_\_\_\_ | def decrypt(code,key):  code\_new =""  for s in code:  m=ord(s)  code\_new +\_\_\_\_(1)\_\_\_\_\_\_  print(code\_new)  return code\_new |

③自定义主程序main。运行main()。通过1，2 进行加密，解密选择

def main():

print("请选择数字1或2：")

print("1：加密")

print("2：解密")

select = input()

if select == "1":

code = input("请输入加密字符串：")

key = int(input("请输入偏移位数："))

encrypt(code,key)

elif select == "2":

code = input("请输入解密字符串：")

key = int(input("请输入偏移位数："))

decrypt(code,key)

else:

print("输入错误，请重试！")

if \_\_name\_\_ == '\_\_main\_\_':

main()

**【辨学单】**

1.换位密码算法的实现。换位密码的基本思想是将明文中的字母位置通过一定的规则重新排列。

最简单的换位就是逆序法，即将明文中的字母倒过来输出。例如:

明文: How are you?

密文: ?uoy era woH

(1)明文为”love”,密钥为2，则密文为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

(2)换位密码算法实现，补全下面代码：

def encrypt(code, key):

code\_new = ""

for i in range(len(code)):

code\_new += \_\_\_\_\_\_\_\_\_(1)\_\_\_\_\_\_\_\_\_\_\_\_\_\_

print(code\_new)

return code\_new

def main():

code = input("请输入加密字符串：")

key = int(input("请输入偏移位数："))

encrypt(code,key)

if \_\_name\_\_=='\_\_main\_\_':

main()

**【省学单】**

1. 简单异或法

异或运算，是一种逻辑运算，其数学符号为“⊕”。运算时要求把参与运算的数转换为二进制数再进行按位运算。如果两个值不相同，那么异或结果为1。如果两个值相同，那么异或结果为0。异或运算具有如下特点:0⊕0=0, 0⊕1=1, 1⊕0=1，1⊕1=0

对于任意一个字符，都可以用二进制编码形式来表示，字符的异或运算就是对每位进行运算。例如，字符串“Hello" (8位ASCII表示: 01001000 01 100101 01 101 100 01 101 10001101111 )可以按如下的方式用密钥101 10001进行加密:

01001000 01100101 01101100 01101100 01101111 (P-明文)

⊕ 10110001 10110001 10110001 10110001 10110001 (k-密钥)

= 11111001 11010100 11011101 11011101 11011110 (C-密文)

简单异或加密，就是将明文与密钥进行异或运算，解密则是对密文用同一密钥进行异或运算。即:

P⊕K=C C⊕K=P

算法代码实现： 注意运算符“^”能够实现异或运算

def encrypt(code, key):

code\_new = ""

for i in range(len(code)):

code\_new +=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_（1）

print(code\_new)

return code\_new

def main():

code = input("请输入加密字符串：")

key =input("请输入密钥字符串：")

encrypt(code,key)

if \_\_name\_\_=='\_\_main\_\_':

main()