山东大学 软件 学院

Python密码学编程课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100150209 | 姓名： 杨佳庆 | | 班级： 21级网安班 |
| 实验题目：实验1 | | | |
| 实验学时：2 | | 实验日期： 2023.9.26 | |
| 实验目的：   * 熟悉Pycharm/ Python Idle开发 * 完成Casear密码的编程。 | | | |
| 硬件环境：  PC机 | | | |
| 软件环境：  PyCharm | | | |
| 实验步骤与内容：  1、安装Python、Pycharm；了解Idle的应用；学会pip安装命令，将Cryptography gmpy2库安装到python中。    2、编写Casear密码程序，扩展到仿射密码、以及Casear密码的破译程序。  Casear.py 第一版  def caesar\_encrypt(k):  message = input('请输入需要加密的字符串：')  ciphertext = ''  for i in message:  ciphertext += (chr(ord(i) + k)) # 将明文汉字转换为对应ASCLL数值或Unicode数值（ord函数），然后在此数值上+3，再将该值返回汉字（chr函数）  print(ciphertext) def caesar\_decrypt(k):  message = input('请输入需要解密的字符串：')  plaintext = ''  for i in message:  plaintext += (chr(ord(i) - k)) # 将明文汉字转换为对应ASCLL数值或Unicode数值（ord函数），然后在此数值上-3，再将该值返回汉字（chr函数）  print(plaintext)   if \_\_name\_\_ == '\_\_main\_\_':  k = 3  caesar\_encrypt(k)  caesar\_decrypt(k)  这一版写的逻辑很简单，但是没有写条件，如果key值很大可能所有字母的加密和解密都会失效，或者即使在key值很小的情况下对靠后的字母的加密或者解密仍会出错。  第二版：  def caesar\_encrypt(key,text = 'abcxyzABCXYZ'):  ciphertext = ''  for i in range(len(text)):  char = text[i]  if (char.isupper()):  ciphertext += chr((ord(char) + key - 65) % 26 + 65)  else:  ciphertext += chr((ord(char) + key - 97) % 26 + 97)  return ciphertext def caesar\_decrypt(key,text = 'abcxyzABCXYZ'):  key = -key  plaintext = ''  for i in range(len(text)):  char = text[i]  if (char.isupper()):  plaintext += chr((ord(char) + key - 65) % 26 + 65)  else:  plaintext += chr((ord(char) + key - 97) % 26 + 97)  return plaintext   if \_\_name\_\_ == '\_\_main\_\_':  k = 3  # text = input("请输入需要加密的字符串：")  print(caesar\_encrypt(k))  print(caesar\_decrypt(k))  这一版加上了条件判断，已经可以正确运行，但加密和解密重用率太高了，下一版会整合一下，写到一个函数里，通过一个参数来判定是加密还是解密。  第三版：  def caesar (mode, message, key):  if mode[0] == 'd':  key = -key  ciphertext = ''  for symbol in message:  if symbol.isalpha():  num = ord(symbol)+key  if symbol.isupper(): # 所有密文字母是大写  if num > ord('Z'):  num -= 26  elif num < ord('A'):  num += 26  elif symbol.islower():  if num > ord('z'):  num -= 26  elif num < ord('a'):  num += 26  ciphertext += chr(num)  else:  ciphertext += symbol  return ciphertext if \_\_name\_\_ == '\_\_main\_\_':  key = int(input("请输入偏移量：")) % 26  message = input('请输入需要加密的字符串：')  mode = input('请输入加密或解密(e/d)：')  print("加密或解密后的结果为：", caesar (mode, message, key))  这一版整合了加密和解密的算法，通过把Key取负的方式来选择加密还是解密。  看到网上的暴力破解凯撒密码算法，略微修改了一下  def caesar\_decrypt(message):  LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  letters = "abcdefghijklmnopqrstuvwxyz"  for key in range(26):  translated = ''  for symbol in message:  if symbol in LETTERS:  num = LETTERS.find(symbol)  num = num - key  if num < 0:  num = num + len(LETTERS)  translated = translated + LETTERS[num]  elif symbol in letters:  num = letters.find(symbol)  num = num - key  if num < 0:  num = num + len(letters)  translated = translated + letters[num]  else:  translated = translated + symbol  print('Hacking key #%s: %s' % (key, translated))  仿射密码  class Affine(object):  DIE = 128  KEY = (7, 3, 55)   def \_\_init\_\_(self):  pass   def encryptChar(self, char):  K1, K2, kI = self.KEY  return chr((K1 \* ord(char) + K2) % self.DIE)   def encrypt(self, string):  return "".join(map(self.encryptChar, string))   def decryptChar(self, char):  K1, K2, KI = self.KEY  return chr(KI \* (ord(char) - K2) % self.DIE)   def decrypt(self, string):  return "".join(map(self.decryptChar, string))  结果如下图：  补充结果截屏 | | | |
| 结论分析与体会：  在这个实验中，我们熟悉了Python开发环境，包括Pycharm和Python Idle。我们学习了如何安装Python和Pycharm，并了解了Idle的应用。我们还学会了使用pip安装命令，将Cryptography和gmpy2库安装到Python中。  在完成Casear密码的编程实现过程中，我们首先了解了Casear密码的原理。Casear密码是一种简单的替换密码，它将字母按照一定的偏移量进行替换，从而实现加密和解密的功能。  我们编写了Casear密码的程序，可以通过输入明文和偏移量来加密和解密文本。我们还扩展了程序，实现了仿射密码的功能。仿射密码是Casear密码的一种改进，它使用了一个加密函数，包括线性函数和模运算，使得加密更加复杂和安全。  此外，我们还编写了Casear密码的破译程序。破译程序通过尝试所有可能的偏移量，对密文进行解密，并根据解密后的文本的可读性进行评估，找到最有可能的明文。  通过完成这个实验，我们加深了对Python开发环境的理解，并学会了使用Python编写简单的密码学程序。我们还了解了密码学中的一些基本概念，如替换密码和破译方法。 | | | |