Tamevic's Ctf-Crypto writeup@实验一"维吉尼亚密码的实现和破解"

原创

TameVic 于 2019-07-04 08:43:50 发布 1268 收藏 4
分类专栏： Crypto 文章标签： Crypto

本文链接：https://blog.csdn.net/qq_43264421/article/details/94593669
版权

Crypto 专栏收录该内容
2 篇文章 0 订阅
订阅专栏
Tamevic’s Ctf-Crypto writeup@实验一"维吉尼亚密码的实现和破解"

> 应用密码学的第一个实验

分析

首先 关于维吉尼亚密码（`Vigenere`）：

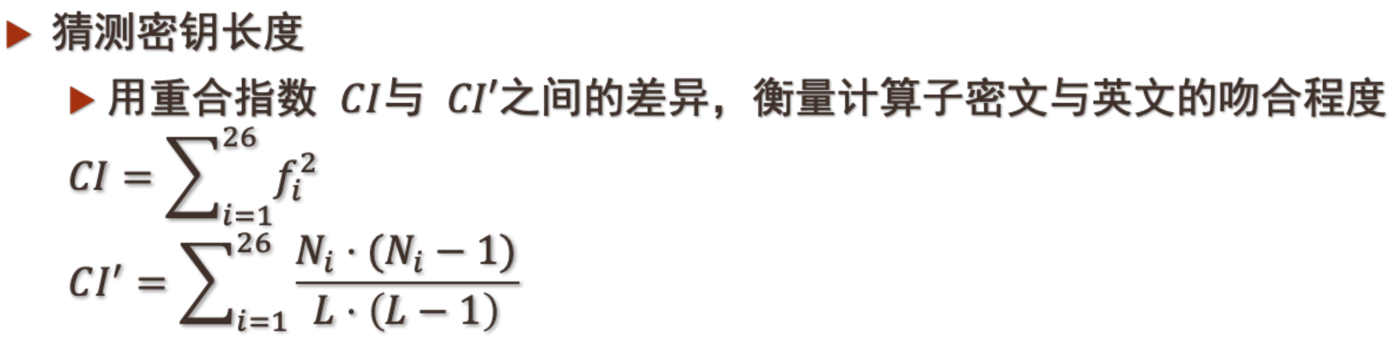
它是凯撒加密的一种升级版本，即明文的每一位都是用凯撒加密（移位），但所用的密钥（移动位数）却不是相同的。在实现的过程中需要考虑加密策略，主要是对空格和标点符号的处理。（遇到标点符号和空格时密钥是否跟着移一位。如`“hello world”`用`“nihao”`加密，如果遇到非字母时密钥也跳一位的情况下密文为`“umslc jwylr”`，而不跳一位的情况下密文为`“umslc evrzq”`）我们这里加密的时候使用的策略是遇到非字母字符密钥不往后顺延。

这样来看的话，对于攻击者来说复杂度就变为了两层，一层是密钥的长度，一层是密钥的内容。

其次 关于破解思路：

古典密码的一个最重要的问题在于没有办法隐蔽文章的统计特性，维吉尼亚密码也未能避免这个问题。所以拿到一串密文，根据之前分析到的两层结构，进行试验性爆破。

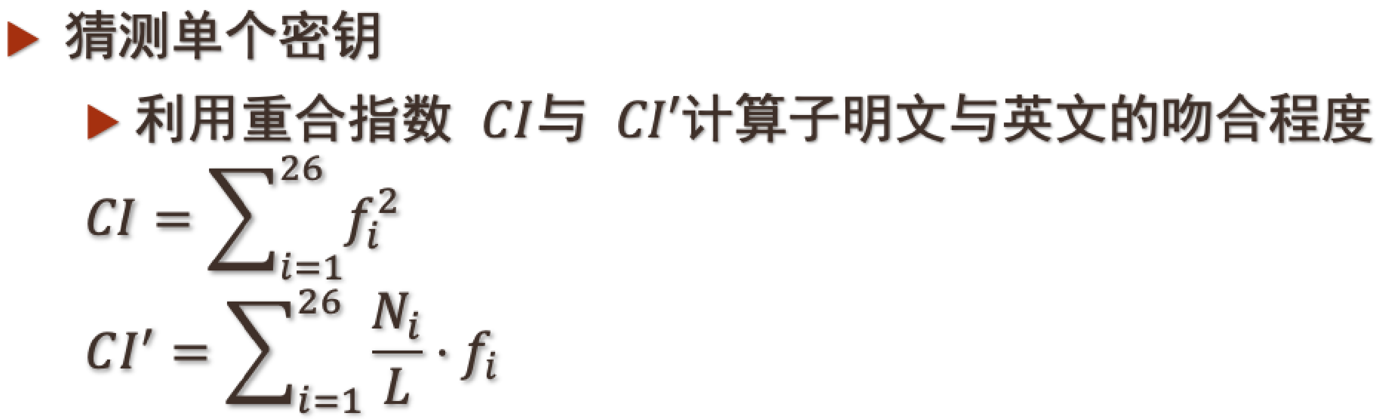
这里用到的主要是拟重合指数（CI）进行试验。



- 猜测密钥长度
  - 用重合指数 CI与 CI'之间的差异，衡量计算子密文与英文的吻合程度

$$CI = \sum_{i=1}^{26} f_i^2$$

$$CI' = \sum_{i=1}^{26} \frac{N_i \cdot (N_i - 1)}{L \cdot (L - 1)}$$



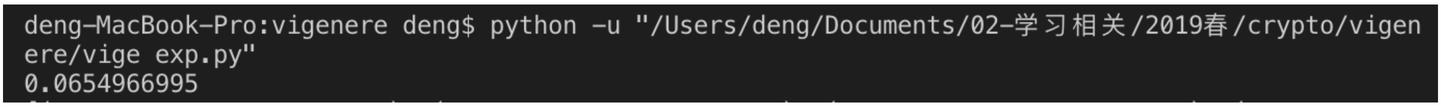
- 猜测单个密钥
  - 利用重合指数 CI与 CI'计算子明文与英文的吻合程度

$$CI = \sum_{i=1}^{26} f_i^2$$

$$CI' = \sum_{i=1}^{26} \frac{N_i}{L} \cdot f_i$$

CI指随机取出两个字符相同的概率

在一段正常的英文原文中CI的值是用所有的字母频率带入计算



```
deng-MacBook-Pro:vigenere deng$ python -u "/Users/deng/Documents/02-学习相关/2019春/crypto/vigen
ere/vige exp.py"
0.0654966995
```

CI≈0.065

通过和英文CI相比对，越接近这个值证明字母特性越符合英文原文。

破解时：
先猜测密钥长度，根据密钥长度对密文进行分组，把每一组中处于相同位置的密文放在一个分组里，这样一个分组内的密文都是用一个密钥进行加密的，他们的密文也会符合英文原文的重合指数。对一定范围内的密钥长度进行暴破，和CI值进行比对，求得最有可能的密钥长度。
再根据求得的密钥长度对密文进行分组，对每个分组的密钥进行暴破，具体思路和猜测密钥长度一致。看哪个密钥下重合指数比较符合英文原文，那么那个分组的密钥就是那个密钥。

最后再解密验证。

结果

加解密实现

```
# coding: utf-8
def encrypt(message,key):
    cipher=''
    j=0
    for i in range (len(message)):

        if key[j % len(key)].islower():
            offset = ord(key[j % len(key)]) - ord('a')
        else:
            if key[j % len(key)].isupper():
                offset = ord(key[j % len(key)]) - ord('A')
            else:
                offset = ord(key[j % len(key)]) - 48

        j+=1

        if message[i].isalpha():
            if message[i].islower():
                cipher += chr((ord(message[i]) - ord('a') + offset )%26 +ord('a'))
            else:
                cipher += chr((ord(message[i]) - ord('A') + offset )%26 +ord('A'))
        else:
            cipher += message[i]
            j-=1

    return cipher

def decrypt(cipher,key):
    explain=''
    j=0
    for i in range (len(cipher)):

        if key[j % len(key)].islower():
                offset = ord(key[j % len(key)]) - ord('a')
        else:
            if key[j % len(key)].isupper():
                offset = ord(key[j % len(key)]) - ord('A')
            else:
                offset = ord(key[j % len(key)]) - 48
        j+=1
        if cipher[i].isalpha():
            if cipher[i].islower():
                explain += chr((ord(cipher[i]) - ord('a') - offset )%26 +ord('a'))
            else:
                explain += chr((ord(cipher[i]) - ord('A') - offset )%26 +ord('A'))
        else:
            explain += cipher[i]
            j-=1
    return explain
```

猜测密钥长度

```
def takeapart(text,size):
    text=text.lower()
    part=['']*size
    for i in range(len(text)):
        if text[i].isalpha():
            part[i%size] += text[i]
    return part
    return part

p = [0.08167, 0.01492, 0.02782, 0.04253, 0.12702, 0.02228, 0.02015, 0.06094, 0.06966, 0.00153, 0.00772, 0.04025,
     0.02406, 0.06749, 0.07507, 0.01929, 0.00095, 0.05987, 0.06327, 0.09056, 0.02758, 0.00978, 0.02360, 0.00150,
     0.01974, 0.00074]

CI=0
m=0
for m in range(26):
    CI+=p[m]*p[m]
# CI=0.065
print CI

def CI1(part):
    count=[0]*26
    length=0
    i=0
    for i in range(len(part)):
        num=ord(part[i])-ord('a')
        if((num>=0) and (num<=25)):
            count[num]+=1
            length+=1
    # print count
    ci1=0.0000000
    j=0
    mot=float(length*(length-1))
    # print mot
    for j in range(25):
        son =0.0
        # print count[j]
        son =count[j]*(count[j]-1)
        # print son/mot
        ci1=ci1+float(son/mot)
    # print ci1
    return ci1

# print (ord('a')-ord('a'))
# count=[0]*26
# print count[0]+1


# part=[''] * 4
# print part

# print messa

# print cipher
dis1=[(0,0.0)]*10
# print dis1
# print takeapart(cipher,3)

def firstci(cipher):
    S=float(0.0)
    for n in range(10):
        part=takeapart(cipher,n+1)
        for i in range(n):
            S+=CI1(part[i])
        # print S/(n+1)
        dis1[n]=(n+1,(S/(n+1))-CI)
    sorted(dis1,key=lambda x:x[1])
    return dis1

dis1=firstci(cipher)
print dis1

def maxci(distance):
    for n in range(len(distance)):
        x=0
        y=0.0
        (x,y)=distance[n]
        if (y>0 and n>0):
            m=0
            p=0.0
            (m,p)=distance[n-1]
            if ( (0-p)>y ):
                return x
            else:
                return m
        else:
            if (n==0 and y>0):
                return x
correctlength = maxci(dis1)
# a=10
# print ~a
print "最有可能的分组长度是：",correctlength
```

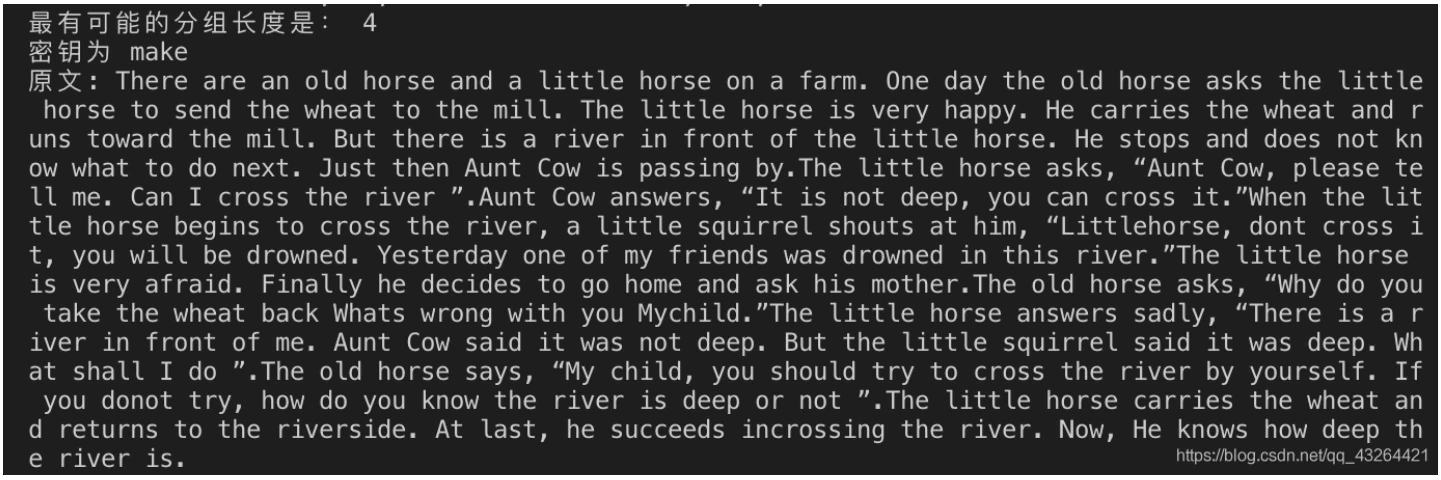
猜测密钥

```
def key_n ( b ):
    y = []
    e= []
    a=','.join (b)
   # x , _ = big_count_number(a)
  #  print (x)
    for k  in [chr(x) for x in range(97,123)]:
        d=temp.decrypt(a , k)
        e.append( (d , k ))
   # print (e)
    for i in range (len (e)):
        x = big_count_number(e[i][0])  , e[i][1]
        y.append(x)
    return(sorted( y ))



def guess_keys (key_len):
    y=[]
    b=seg_txt(key_len)
    for i in range (key_len):
        x=key_n(b[i])
        y.append(x)
    return y
```

End

用明文加密后进行测试

```
message='There are an old horse and a little horse on a farm. One day the old horse asks the little horse to send the wheat to the mill. The little horse is very happy. He carries the wheat and runs toward the mill. But there is a river in front of the little horse. He stops and does not know what to do next. Just then Aunt Cow is passing by.The little horse asks, “Aunt Cow, please tell me. Can I cross the river ”.Aunt Cow answers, “It is not deep, you can cross it.”When the little horse begins to cross the river, a little squirrel shouts at him, “Littlehorse, dont cross it, you will be drowned. Yesterday one of my friends was drowned in this river.”The little horse is very afraid. Finally he decides to go home and ask his mother.The old horse asks, “Why do you take the wheat back Whats wrong with you Mychild.”The little horse answers sadly, “There is a river in front of me. Aunt Cow said it was not deep. But the little squirrel said it was deep. What shall I do ”.The old horse says, “My child, you should try to cross the river by yourself. If you donot try, how do you know the river is deep or not ”.The little horse carries the wheat and returns to the riverside. At last, he succeeds incrossing the river. Now, He knows how deep the river is.'
key='make'
cipher=encrypt(message,key)
#messa=decrypt(cipher,key)
```



```
最有可能的分组长度是： 4
密钥为 make
原文： There are an old horse and a little horse on a farm. One day the old horse asks the little
 horse to send the wheat to the mill. The little horse is very happy. He carries the wheat and r
uns toward the mill. But there is a river in front of the little horse. He stops and does not kn
ow what to do next. Just then Aunt Cow is passing by.The little horse asks, “Aunt Cow, please te
ll me. Can I cross the river ”.Aunt Cow answers, “It is not deep, you can cross it.”When the lit
tle horse begins to cross the river, a little squirrel shouts at him, “Littlehorse, dont cross i
t, you will be drowned. Yesterday one of my friends was drowned in this river.”The little horse
is very afraid. Finally he decides to go home and ask his mother.The old horse asks, “Why do you
 take the wheat back Whats wrong with you Mychild.”The little horse answers sadly, “There is a r
iver in front of me. Aunt Cow said it was not deep. But the little squirrel said it was deep. Wh
at shall I do ”.The old horse says, “My child, you should try to cross the river by yourself. If
 you donot try, how do you know the river is deep or not ”.The little horse carries the wheat an
d returns to the riverside. At last, he succeeds incrossing the river. Now, He knows how deep th
e river is.
```

猜测成功。

统计特性的暴露也能让我们能清楚的认识到古典密码和现代密码的一个重要区别，对接下来的学习起到了比较好的作用！