Hack-me.org 16번

Polyalphabetic substitution

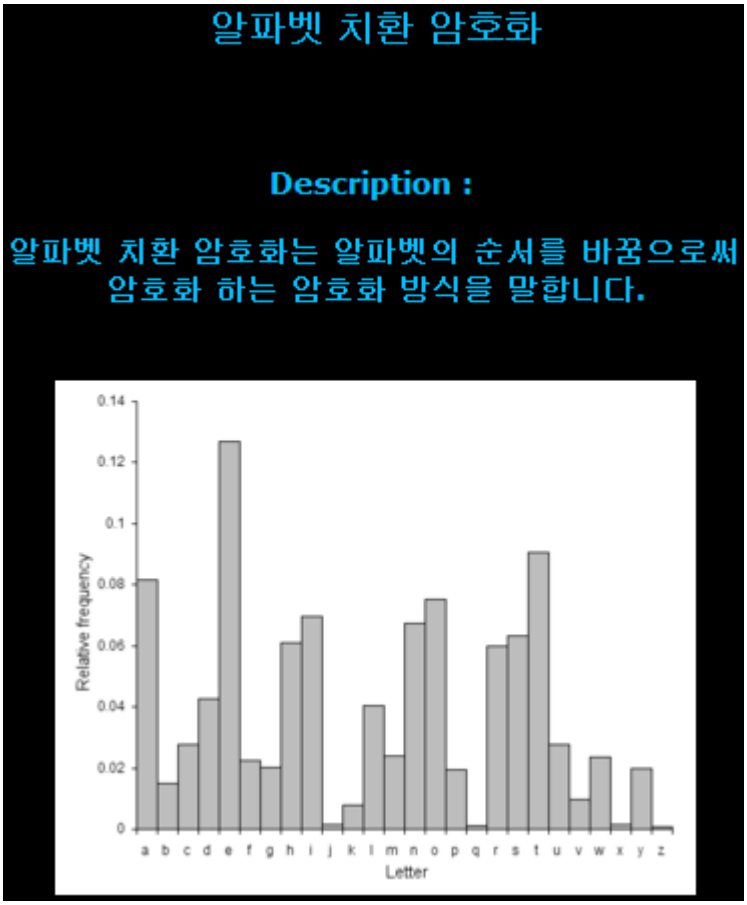
Xero

박준혁 (한국디지털미디어고등학교)

2011-10-21

wnsgurzxc@nate.com

다음과 같이 알파벳 치환 암호화 문제이다.
알파벳 빈도 그래프와 함께 엄청나게 긴 암호문이 같이 나타나 있다.



g qh tqkkn dx ixgy mgdt nxj dxfqn gy mtqd mgoo zx fxmy gy tgudxvn qu dte zveqdeud fehxyudvqdgxy
wxv wveefxh gy dte tgudxvn xw xjv yqdgxy. ... 중간생략 ...

알파벳 치환 암호화이기 때문에 우선 암호문의 알파벳 빈도수를 조사하였다.
다음과 같이 파이썬으로 코딩하였다.
a부터 z까지 for문을 돌리기 위해 a의 아스키 값부터 z의 아스키 값까지 range를 지정하였다.
그리고 chr() 함수를 이용해 알파벳으로 바꿔서 빈도 수를 조사하였다.

```
for i in range(97,123):
        lKey.append(chr(i))
        lVal.append(sProb.count(chr(i)))
```

위의 소스를 돌리자 다음과 같은 값이 나왔다.

```
>>> lKey
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
>>> lVal
[81, 5, 6, 678, 893, 271, 572, 188, 21, 178, 100, 7, 166, 131, 344, 184, 570, 51
, 117, 386, 436, 416, 224, 612, 487, 176]
>>>
```

선택 정렬을 이용해 빈도 수가 높은 순서대로 알파벳을 배열해 보았다.

```
>>> lKey
['e', 'd', 'x', 'g', 'q', 'y', 'u', 'v', 't', 'o', 'f', 'w', 'h', 'p', 'j', 'z',
 'm', 'n', 's', 'k', 'a', 'r', 'i', 'l', 'c', 'b']
>>> lVal
[893, 678, 612, 572, 570, 487, 436, 416, 386, 344, 271, 224, 188, 184, 178, 176,
 166, 131, 117, 100, 81, 51, 21, 7, 6, 5]
>>>
```

다음의 소스를 짜서 디코드하였다.

```
import string
sKey2='etoiansrhldfmcugwybpvkjqzx'
f=file('C:/Users/Sonic/Desktop/input.txt','r')
sProb=f.read()
f.close()
lKey=[]
lVal=[]
for i in range(97,123):
        lKey.append(chr(i))
        lVal.append(sProb.count(chr(i)))
for i in range(26):
        for j in range(i,26):
                if(lVal[i]<lVal[j]):
                        aTemp1=lKey[i]
                        lKey[i]=lKey[j]
                        lKey[j]=aTemp1
                        aTemp2=lVal[i]
                        lVal[i]=lVal[j]
                        lVal[j]=aTemp2
sKey1=''.join(lKey)
trans=string.maketrans(sKey1,sKey2)
sAnswer=sProb.translate(trans)
f=file('C:/Users/Sonic/Desktop/output.txt','w')
f.write(sAnswer)
f.close()
```

그러자 output에 다음과 같은 값이 나타났다.

password is decrypting_polyalphabetic_substitution_is_too_easy

Key : decrypting_polyalphabetic_substitution_is_too_easy