WeChall - Training: Crypto - Digraphs (Crypto, Training)

原创

m0_38134842 于 2021-11-18 15:53:09 发布 9 收藏

分类专栏： CTF 文章标签： CTF入门 安全

本文链接：https://blog.csdn.net/m0_38134842/article/details/121402827

Digraphs

1. Training: Crypto - Digraphs (Crypto, Training)

   题中采用两个字符代替一个字符，理论上可以加密26*26个字符。

   -》替代密码斜体样式加密，可采用穷举或密码分析，但26*26字母表较大，先将密文还原为单字符替代的形式。

```
input = """jmlgzcbnnhxkwikauhxkwifilgzcgzri odlgka jcladcnhcsnkwilajc wiomfigz salagzgzxkbnla gzkadcdclagzgzcuka
uhuhcsri jpxkgz zclgwi wilglg jcficucufidckauhwi lafiwiomlanhib wsxkgz fiwitn jplauhuhib bnlglgjc qflgrnri kezcw
ilanh wiomfigz rblacswslgnhjc xkgz gzlguhkawifilgzcya nkficunkuhlalaomuhnhsaomri"""
s = input.split(' ')

key = []
result = ''
for i in range(len(s)):
    j = s[i]
    for k in range(0, len(s[i]), 2):
        if j[k:k+2] not in key:
            key.append(j[k:k+2])
        result += chr(key.index(j[k:k+2]) + ord('a'))
    result += " "
print(result)
```

还原出单字符替代的形式后，放到quipqiup进行观察

> quipqiup是Edwin Olson的快速自动密码求解器。它可以解决报纸上经常出现的**简单替代密码**，包括诸如密码窃听器（保留单词边界）之类
>
> 的难题和爱国主义者（如密码迷）之类的难题。

根据明密文对应关系建立部分明密文字典，进一步进行分析。

根据找出的message、difficult、either、decrypted等

```
input = """jmlgzcbnnhxkwikauhxkwifilgzcgzri odlgka jcladcnhcsnkwilajc wiomfigz salagzgzxkbnla gzkadcdclagzgzcuka uhuhcsri jpxkgz zclgwi wilglg jcficucufidckauhwi lafiwiomlanhib wsxkgz fiwitn jplauhuhib bnlglgjc qflgrnri kezcw ilanh wiomfigz rblacswslgnhjc xkgz gzlguhkawifilgzcya nkficunkuhlalaomuhnhsaomri"""
s = input.split(" ")

dic = {'od': 'y', 'lg': 'o', 'ka': 'u', 'jc': 'd', 'la': 'e', 'dc': 'c', 'nh': 'r', 'cs': 'y', 'nk': 'p', 'wi': 't',
       'om': 'h', 'fi': 'i', 'gz': 's', 'sa': 'm', 'bn': 'g', 'xk': 'a', 'uh': 'l', 'zc': 'n', 'cu': 'f', 'rb': 'k',
       'ws': 'w', 'ke': 'e', 'jm': 'C'}
# dic = {'ie':'C','rk':'o','vk':'n','ll':'g','hw':'r','ha':'a','rd':'t','jz':'u','aa':'l','sx':'i','nq':'s','cx':'y'}
for i in s:
    a = []
    for j in range(0, len(i), 2):
        a.append(i[j:j + 2])  # 密文字符串每两个分开
    print(a)
    b = []
    for k in a:
        if k in dic:
            b.append(dic[k])  # 解密
        else:
            b.append('_')  # 解不出来的用_填充
    txt = ''.join(b)
    print(txt)
```

注意，其中包含标点符号。

最后得出flag：pifpleehlrmh

```
['jm', 'lg', 'zc', 'bn', 'nh', 'xk', 'wi', 'ka', 'uh', 'xk', 'wi', 'fi', 'lg', 'zc', 'gz', 'ri']
Congratulations_
['od', 'lg', 'ka']
you
['jc', 'la', 'dc', 'nh', 'cs', 'nk', 'wi', 'la', 'jc']
decrypted
['wi', 'om', 'fi', 'gz']
this
['sa', 'la', 'gz', 'gz', 'xk', 'bn', 'la']
message
['gz', 'ka', 'dc', 'dc', 'la', 'gz', 'gz', 'cu', 'ka', 'uh', 'uh', 'cs', 'ri']
successfully_
['jp', 'xk', 'gz']
_as
['zc', 'lg', 'wi']
not
['wi', 'lg', 'lg']
too
['jc', 'fi', 'cu', 'cu', 'fi', 'dc', 'ka', 'uh', 'wi']
difficult
['la', 'fi', 'wi', 'om', 'la', 'nh', 'ib']
either_
['ws', 'xk', 'gz']
was
['fi', 'wi', 'tn']
its
['jp', 'la', 'uh', 'uh', 'ib']
_ell_
['bn', 'lg', 'lg', 'jc']
good
['qf', 'lg', 'rn', 'ri']
_o__
['ke', 'zc', 'wi', 'la', 'nh']
enter
['wi', 'om', 'fi', 'gz']
this
['rb', 'la', 'cs', 'ws', 'lg', 'nh', 'jc']
keyword
['xk', 'gz']
as
['gz', 'lg', 'uh', 'ka', 'wi', 'fi', 'lg', 'zc', 'ya']
solution_
['nk', 'fi', 'cu', 'nk', 'uh', 'la', 'la', 'om', 'uh', 'nh', 'sa', 'om', 'ri']
pifpleehlrmh_
```