DeconstruCT.F 2021 Crypto Writeups

原创

M@ku1i 于 2021-10-05 21:06:14 发布 71 收藏

文章标签： 密码学

本文链接：https://blog.csdn.net/weixin_56678592/article/details/120618143

DeconstruCT.F 2021 Crypto Writeups

#1 RSA-1

直接分解模数n，得到p和q后常规解密。

#2 RSA-2

简单的低加密指数攻击，套板子直接解密。

#3 Stars and Shapes

题目只给了一个gif一直闪烁各个形状，利用StegSolve逐帧查看后发现与盲文形式类似。都是这样2x3的矩阵，尝试盲文解密，利用https://www.boxentriq.com/code-breaking/braille-alphabet，其中有两帧是带有花括号的，将除了这两帧以外的全部解密后得到flag。

flag：dsc{d0-y0u-th1nk-h3-s4w-us7132}

#4 The Conspiracy

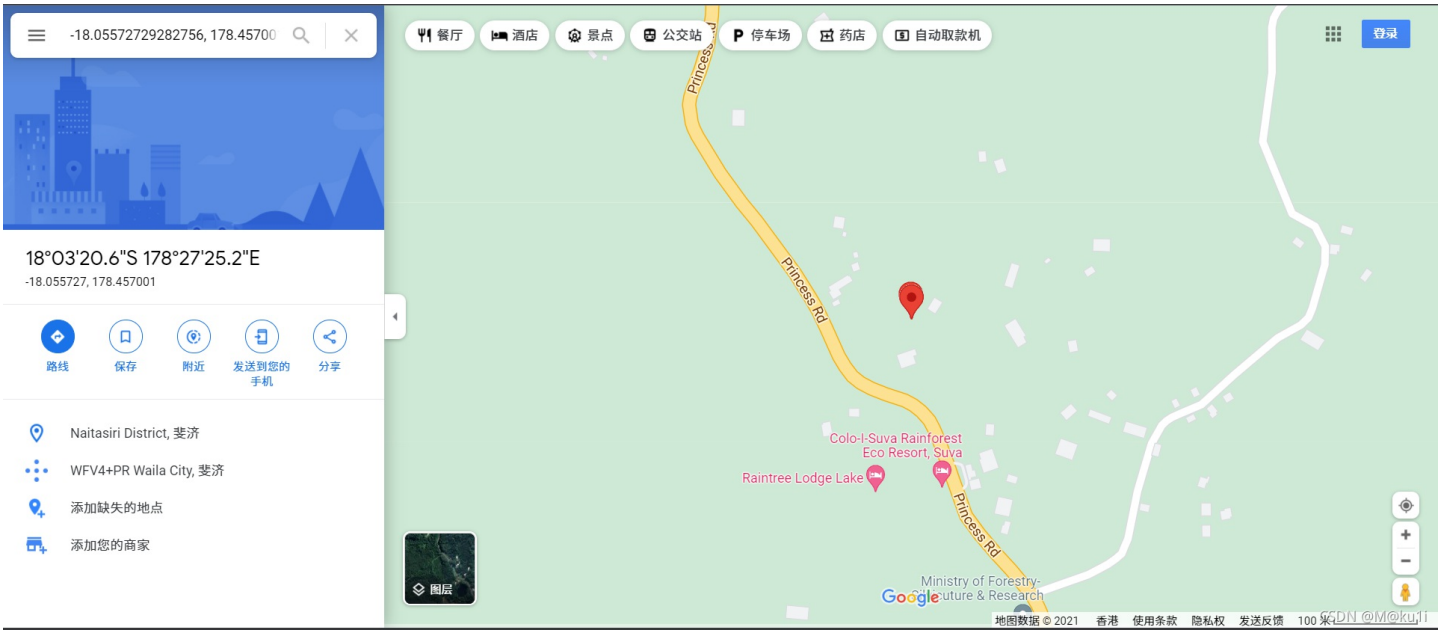
附件是一个名为diary的txt文件，并且题目中给了提示，提示说这个diary是一个航海家记录的，并且这个航海家去过很多国家。

> ZHNjeygtMTguMDU1Nzl3MjkyODI3NTYsIDE3OC40NTcwMDE0MzEzMTY3NCksKDE5LjAyODI4Mjg1NzUwNTM5MiwgMTAzLjE0NDI2MDcxMjA3MTA3KSwoNDIuNTM2NzA1OTkxMjY2MTQ2LCAxLjQ5MzAzNDQ2MTIyNzY5MzMpLCgzOC41ODkzNjk3MjE3MzU0LCA2OC44MTYzMjUyMzA1ODk2NyksXywoNTAuODUxNTE4OTQ4MjA2Nzk1LCA0LjM2MDE4MDg1MzU4MTk4NiksKDcuNjcxODYzNTM4NDUzMzg2LCAzNi44MzcyNjA5NTY5ODk1MiksXywoMzguNjE5NTA2NzQwNTg3MDM1LCAzNC44NTUxMjY0MjcxMTAwNCksKDQ3LjQyODA2MTU3MTIzNTQ1LCAxOC45OTg0MjExMDE5MDM2MDIpLCgzMC4xOTM4MzU4NzE1ODM1MzkyMSwgMzEuMTI0OTY2MTE2MTI1NzMpLF8sKC0wLjIzMTc3NTk0MjYxMTE1NTgsIC03OC41MDMzOTYzMDgzOTQ5KSwoLTEyLjg1NDQ2OTM2NzM0NTUsIDEzMi43OTI2MjQzMzg4MzI5OCksKDQ0LjQyNDIzMTI1MjU3NzQzNSwgMjQuMzUwMjQxMTY2Mjc5MjI2KSwoMjQuODE5NTY0NzQxMzkyMywgMTIwLjk3MjM2NzU0MDA1MDU4KSwoMTguNTg0NDU2OTc0NDAzNDg3LCAtNzIuMzE4MTIxODk2MTQ3MildfQ==

不难发现是base64串，解密后得

> dsc{(-18.05572729282756, 178.45700143131674),(19.028282857505392, 103.14426071207107),(42.536705991266146, 1.4930344612276933),(38.5893697217354, 68.81632523058967),,*(50.851518948206795, 4.360180853581986), (7.671863538453386, 36.83726095698952),*,(38.619506740587035, 34.85512642711004),(47.42806157123545, 18.998421101903602),(30.19387158353921, 31.12496611612573),_,(-0.23177559426111558, -78.5033963083949),(-12.85446936733455, 132.79262433883298),(44.424231252577435, 24.350241166279226),(24.8195647413923, 120.97236754005058),(18.584456974403487, -72.31812189614772)}

结合提示利用谷歌地图确定每个坐标的国家或地区，分别取它们的英文首字母，以第一个坐标为例。



可以看到该国家或地区是斐济，所以取f，以此类推得到flag。

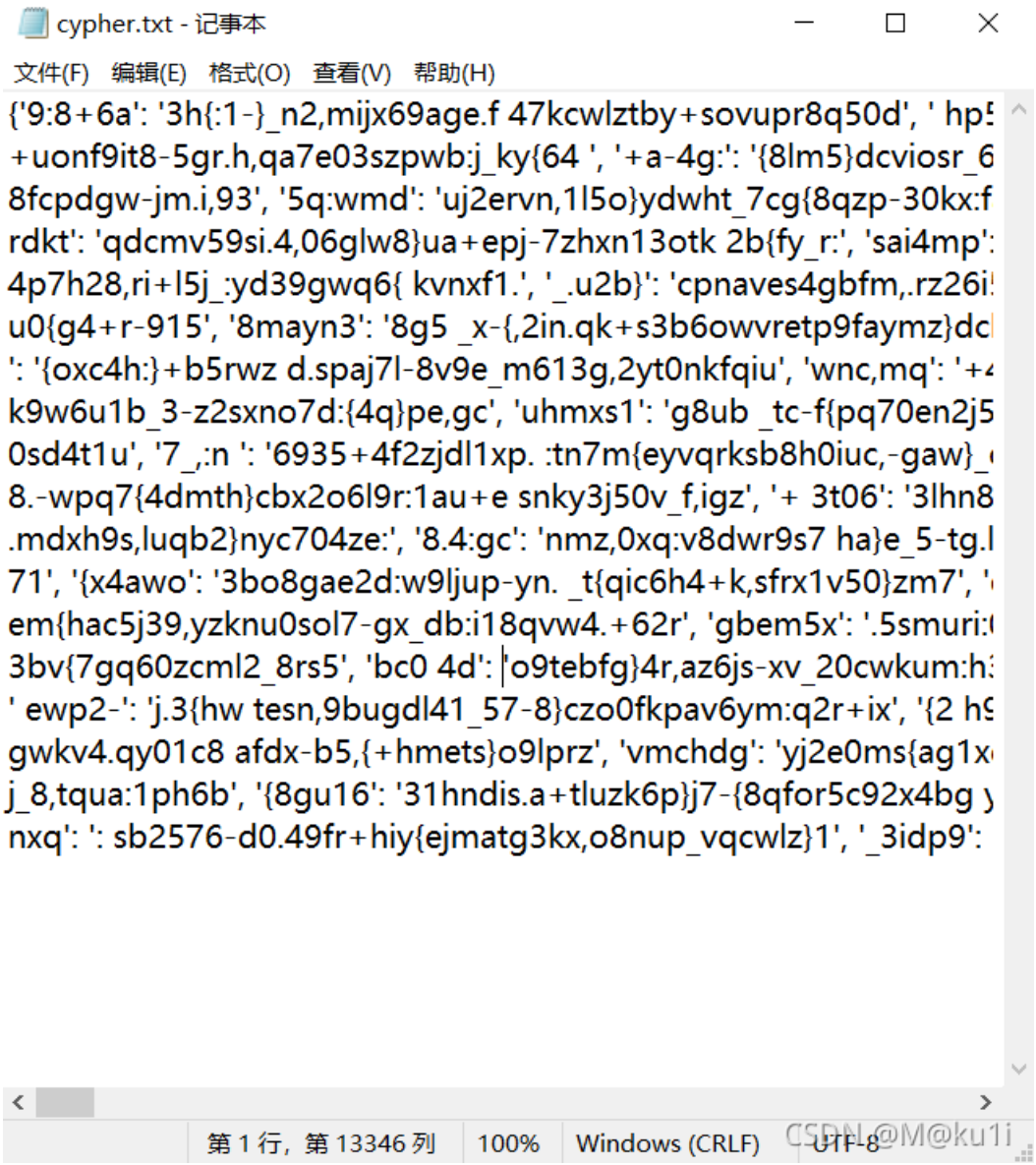
flag：dsc{flat_be_the_earth}

#5 Code Decode

题目描述：

大概就是出题人弄丢了解密脚本，让我们帮他写一个去解他给的密文。

附件：


cypher.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
{'9:8+6a': '3h{:1-}_n2,mijx69age.f 47kcwlztby+sovupr8q50d', ' hp5
+uonf9it8-5gr.h,qa7e03szpwb:j_ky{64 ', '+a-4g:': '{8lm5}dcviosr_6
8fcpdgw-jm.i,93', '5q:wmd': 'uj2ervn,1l5o}ydwht_7cg{8qzp-30kx:f
rdkt': 'qdcmv59si.4,06glw8}ua+epj-7zhxn13otk 2b{fy_r:', 'sai4mp':
4p7h28,ri+l5j_:yd39gwq6{ kvnxf1.', '_.u2b}': 'cpnaves4gbfm,.rz26i
u0{g4+r-915', '8mayn3': '8g5 _x-{,2in.qk+s3b6owvretp9faymz}dc
': '{oxc4h:}+b5rwz d.spaj7l-8v9e_m613g,2yt0nkfqiu', 'wnc,mq': '+
k9w6u1b_3-z2sxno7d:{4q}pe,gc', 'uhmxs1': 'g8ub _tc-f{pq70en2j5
0sd4t1u', '7_,:n ': '6935+4f2zjdl1xp. :tn7m{eyvqrksb8h0iuc,-gaw}_
8.-wpq7{4dmth}cbx2o6l9r:1au+e snky3j50v_f,igz', '+ 3t06': '3lhn8
.mdxh9s,luqb2}nyc704ze:', '8.4:gc': 'nmz,0xq:v8dwr9s7 ha}e_5-tg.l
71', '{x4awo': '3bo8gae2d:w9ljup-yn. _t{qic6h4+k,sfrx1v50}zm7', '
em{hac5j39,yzknu0sol7-gx_db:i18qvw4.+62r', 'gbem5x': '.5smuri:
3bv{7gq60zcml2_8rs5', 'bc0 4d': 'o9tebfg}4r,az6js-xv_20cwkum:h
' ewp2-': 'j.3{hw tesn,9bugdl41_57-8}czo0fkpav6ym:q2r+ix', '{2 h
gwkv4.qy01c8 afdx-b5,{+hmets}o9lprz', 'vmchdg': 'yj2e0ms{ag1x
j_8,tqua:1ph6b', '{8gu16': '31hndis.a+tluzk6p}j7-{8qfor5c92x4bg
nxq': ': sb2576-d0.49fr+hiy{ejmatg3kx,o8nup_vqcwlz}1', '_3idp9':
第 1 行，第 13346 列
100%
Windows (CRLF)
UTF-8
CSDN @M@ku1i

题目：

```
from random import choice

inputstring = input("Enter plaintext: ")


def read_encryption_details():
    with open("cypher.txt") as file:
        encrypt_text = eval(file.read())
        encrypt_key = choice(list(encrypt_text.keys()))
        character_key = encrypt_text[encrypt_key]
    return encrypt_key, character_key


def create_encryption(character_key):
    charstring = "abcdefghijklmnopqrstuvwxyz1234567890 _+{}-,.:"
    final_encryption = {}
    for i, j in zip(charstring, character_key):
        final_encryption[i] = j
    return final_encryption


def convert_plaintext_to_cypher(inputstring, final_encryption, encrypt_key):
    cypher_text = ""
    for i in inputstring:
        cypher_text += final_encryption[i]
    cypher_text = encrypt_key[:3] + cypher_text + encrypt_key[3:]
    return cypher_text


encrypt_key, character_key = read_encryption_details()
final_encryption = create_encryption(character_key)
cypher_text = convert_plaintext_to_cypher(
    inputstring, final_encryption, encrypt_key)

print(cypher_text)
```

思路：

可以看到最后一个convert_plaintext_to_cypher函数负责最后的加密工作，我们只要获得这一步的所有参数的内容就可以了。

利用read_encryption_details和create_encryption函数来获得final_encryption，再利用final_encryption写出对应的final_decryption解密就可以了。

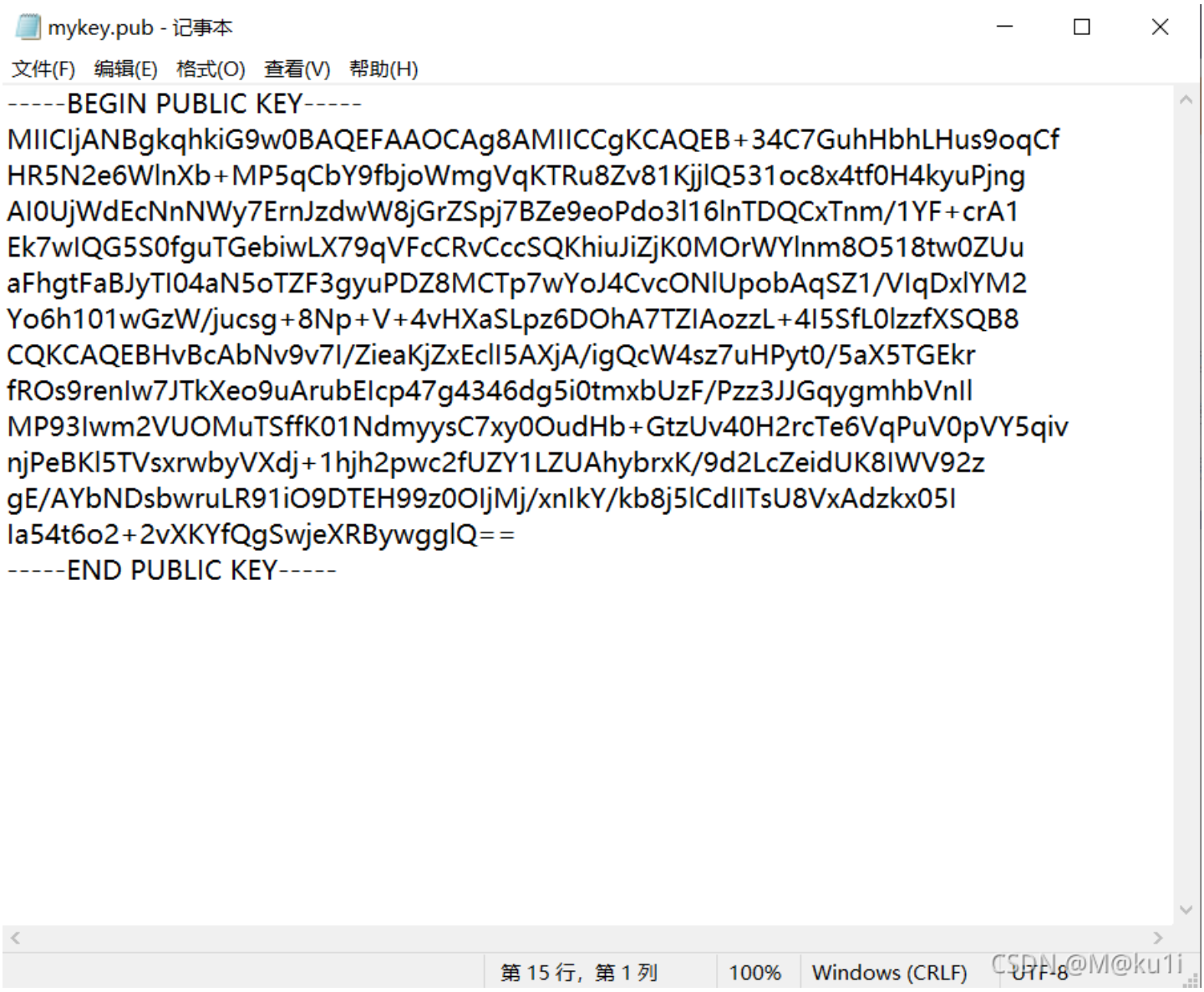
exp：

```
from random import choice

encrypt_key = '2nj-5x'
character_key = 'cxkl,_}o 4+tzrwe7ig9bfu5a-sy01.hpn628v3m{d:jq'
final_encryption = {'a': 'c', 'b': 'x', 'c': 'k', 'd': 'l', 'e': ',', 'f': '_', 'g': '}', 'h': 'o', 'i': ' ', 'j
': '4', 'k': '+', 'l': 't', 'm': 'z', 'n': 'r', 'o': 'w', 'p': 'e', 'q': '7', 'r': 'i', 's': 'g', 't': '9', 'u':
 'b', 'v': 'f', 'w': 'u', 'x': '5', 'y': 'a', 'z': '-', '1': 's', '2': 'y', '3': '0', '4': '1', '5': '.', '6': '
h', '7': 'p', '8': 'n', '9': '6', '0': '2', ' ': '8', '_': 'v', '+': '3', '{': 'm', '}': '{', '-': 'd', ',': ':'
, '.': 'j', ':': 'q'}
final_decryption = {'c': 'a', 'x': 'b', 'k': 'c', 'l': 'd', ',': 'e', '_': 'f', '}': 'g', 'o': 'h', ' ': 'i', '4
': 'j', '+': 'k', 't': 'l', 'z': 'm', 'r': 'n', 'w': 'o', 'e': 'p', '7': 'q', 'i': 'r', 'g': 's', '9': 't', 'b':
 'u', 'f': 'v', 'u': 'w', '5': 'x', 'a': 'y', '-': 'z', 's': '1', 'y': '2', '0': '3', '1': '4', '.': '5', 'h': '
6', 'p': '7', 'n': '8', '6': '9', '2': '0', '8': ' ', 'v': '_', '3': '+', 'm': '{', '{': '}', 'd': '-', ':': ','
, 'j': '.', 'q': ':'}
cypher_text = '2njlgkma2bv1i0v}221v19vuo19va2bvl2{-5x'
def read_encryption_details(encrypt_key):
    with open("cypher.txt") as file:
        encrypt_text = eval(file.read())
        character_key = encrypt_text[encrypt_key]
    return character_key
#print(read_encryption_details(encrypt_key))

def create_encryption(character_key):
    charstring = "abcdefghijklmnopqrstuvwxyz1234567890 _+{}-,.:"
    final_encryption = {}
    for i, j in zip(charstring, character_key):
        final_encryption[i] = j
    return final_encryption
#print(create_encryption(character_key))

def convert_cypher_to_plaintext(cypher_text, final_encryption):
    plain_text = ""
    llen = len(cypher_text)
    cypher_text = cypher_text[3:llen-3]
    for i in cypher_text:
        plain_text += final_decryption[i]
    return plain_text
print(convert_cypher_to_plaintext(cypher_text, final_decryption))
```

flag：dsc{y0u_4r3_g00d_4t_wh4t_y0u_d0}

#6 RSA-3

题目（附件）：



给了一个pub公钥文件，提取后发现e实在太大了，与n相近，尝试维纳攻击。

exp:

```
from Crypto.PublicKey import RSA
import owiener
key = RSA.import_key(open('./mykey.pub').read())
#print("n =",key.n)
#print("e =",key.e)
n = 640649591649238760648749454734070499855431199929927381192527492312531424642036475187774554751099725816847326
210729988980667283034333005852915275829794302763577876340268691160953915143111111742063951958176727373208372403 6
494460997984460198622146284536407039666572302990293265336894345265285417419707074763124210108426091228784928664 4
699582292473152660004035330616149016496957012948833038931711943984563035784805193474921164625068468842927905314 2
689421537200786809373453651211294043846330191830603471297782966405009353821868678504078933879204821412164983393 4
608110643314435248557179540571779304044123865992585719843943 3

e = 362226808584142561613758846021506408090629587181171413829230994943417330931725871171659200972855232763382747
505980224869760835111780913928499860393849757586093435975480391660240422646144965060875971140916639551337799561 7
694132543182268471698812827138441001047175532483313685965297824029712061845853430692355854617611005573723388312 9
780378153307730890915697357455996361736492022695824172516806204252765904924281272883818154621932085365817823019 7
738607836876667880950357904910063334322956981783785204448108138821178173298478745318095309293454307966008707287 3
667838947915932811932258764785627476226235888066458567521909 3

d = owiener.attack(key.e, key.n)

if d is None:
    print("没有找到私钥d")
else:
    print("dsc{"+str(d)+"}")
```

题目中提示flag格式为dsc{number}，所以解出来不做处理直接交。

flag：
dsc{6393313697836242618414301946448995659516429576261871356767102021920538052481829568588047189447471873340140537810769433878383029164089236876209147584435733}

#7 Behind Enemy Lines

附件：

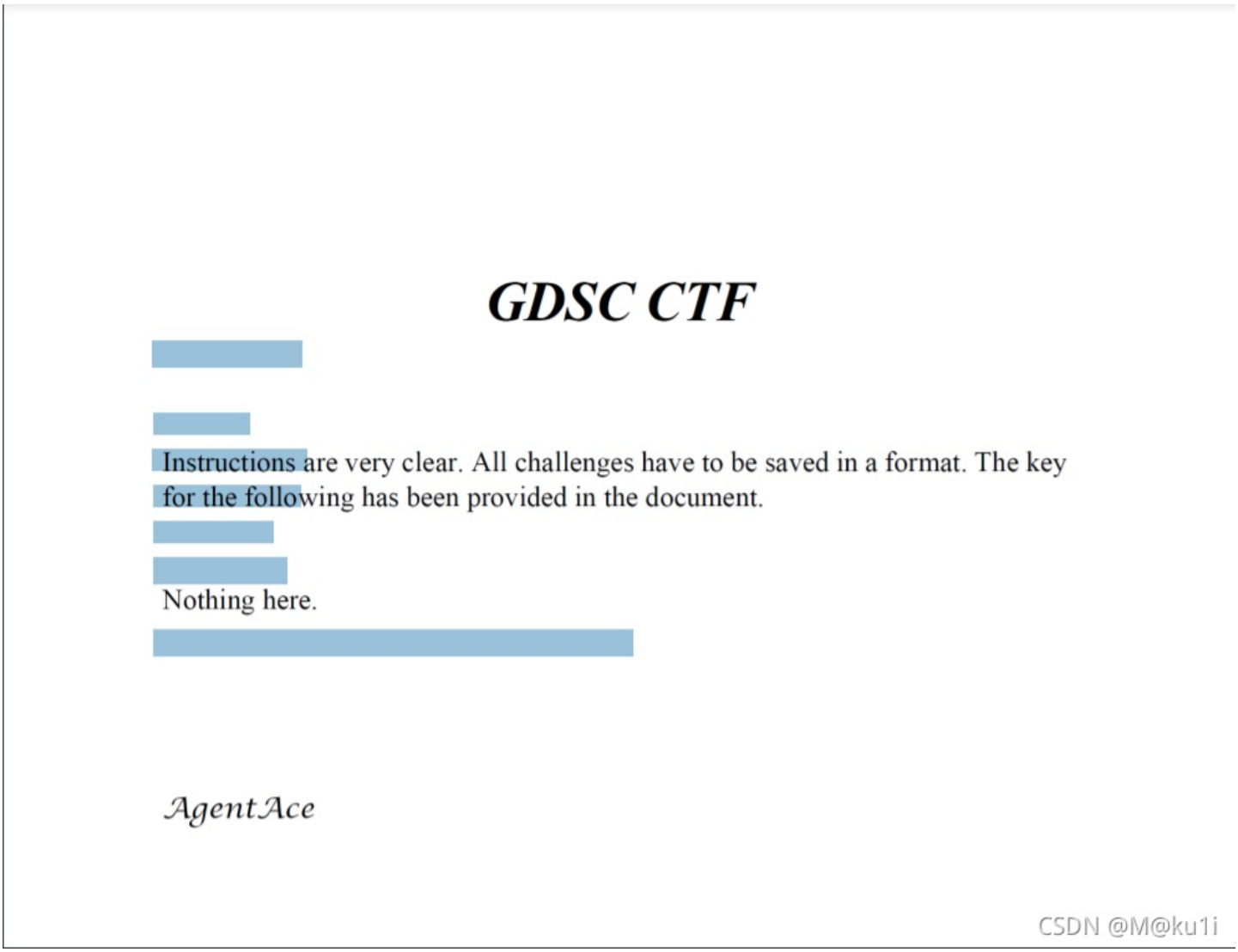
```
mckcu wiyqt jawul xedmi bclke ipdpp jdmvl gugks cggcq iiapb dkphr eymlv yrziv jzhmq lipab yrdbn suhpy wsqio tljo
t mrldl jzqmt qjmkn wahty oycpj ntvsh axhfn thrtu qtxfm ahiav xbqjt spuuf yyxcv qatli ewadf rksdd fhntl fbgjx ar
ngn scwmk kweba rropy uoohj nciho rjolj fozny bxpdu zdqzf ljrmv nopcw ismtz exjql axues ioypx amqms jbyeb pyssp
 ehfv iof jilazhmfpyotp yzdz whykv xjkrb ejvcx qvusj fqgkn mbwli oihvo caqkz mvfoh wcrmp xoujk wcirt slxlf bwbhg
 ecwin tanav zvvrq aoqgv yndwt ieuxf wwqig qafan zjnyj bppmg eegmp gbkqx xlqwv ombdc tlidr rjtvd oefvj cjvsg izl
qf szmpj qdmoe rrcyt pveaa emioj njtus nvcoc iyagm imjzx ljcph xaqrl tkpsc vpwwn jyjxr skqsd brknj radag omfzk w
juyy jyslo ygdpo cprkn dcpzy ynffg eunzh fzkzx hetck lbunm qsxpu zbzof xoakd dovna dmxna ethux ewzfj fjcle ivjbq
 axkbs nxjwx aaesx hmvon zhnuy fkgzn wvfrj jcihe hcknt ijfgw zidhn xlukp pwurl vyvpk idmck ybgfk velpb yomdz tiv
sy rdiyk kvggg jvwct sanep fuzfq j
```

一段密文和一个被加密的instructions.pdf文件。

解题：

利用john爆破pdf密码，跑了很久以后，得到密码ilovejohn。

打开pdf并没有看到任何有用的东西，但是当按下CTRL-A全选的时候会发现有隐藏内容。



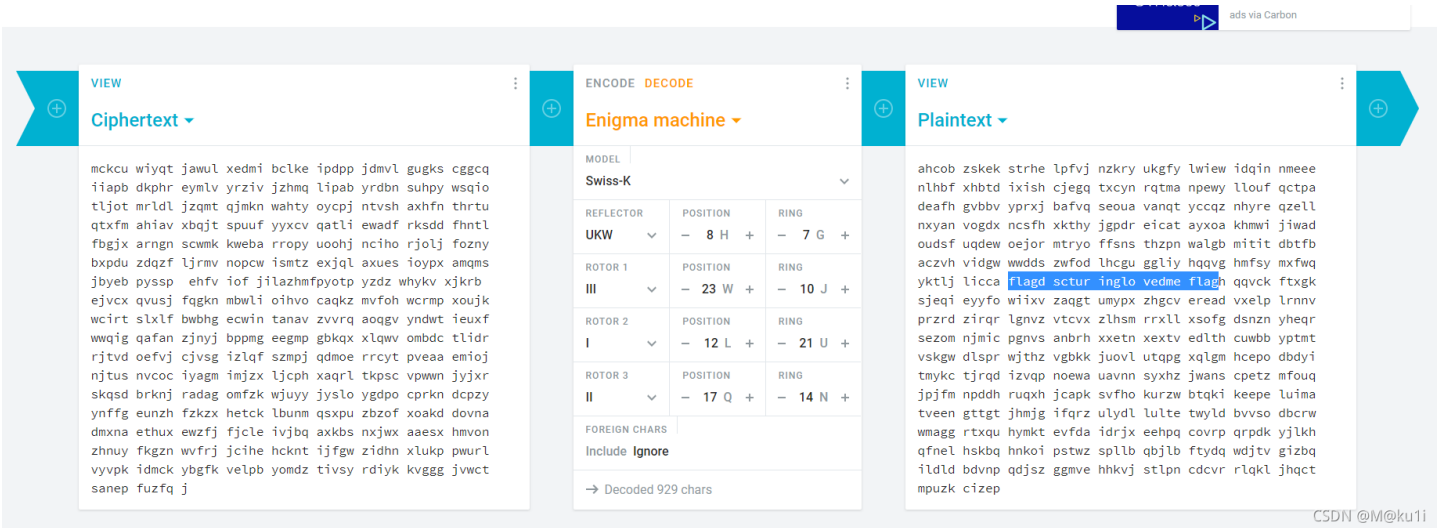
粘贴到文本中得

```
Hidden Key
Swiss K
UKW 8H 7G
III 23 W 10J
I 12L 21U
II 17Q 14N
-- you really thought, I wont hide it here?
```

题目有个提示

谷歌搜了一下纳粹德国使用的密码，查到了一个叫做Enigma的密码机，是当时二战时期纳粹德国使用的一系列相似的转子机械加解密机器的统称，它包括了许多不同的型号，为密码学对称加密算法的流加密。

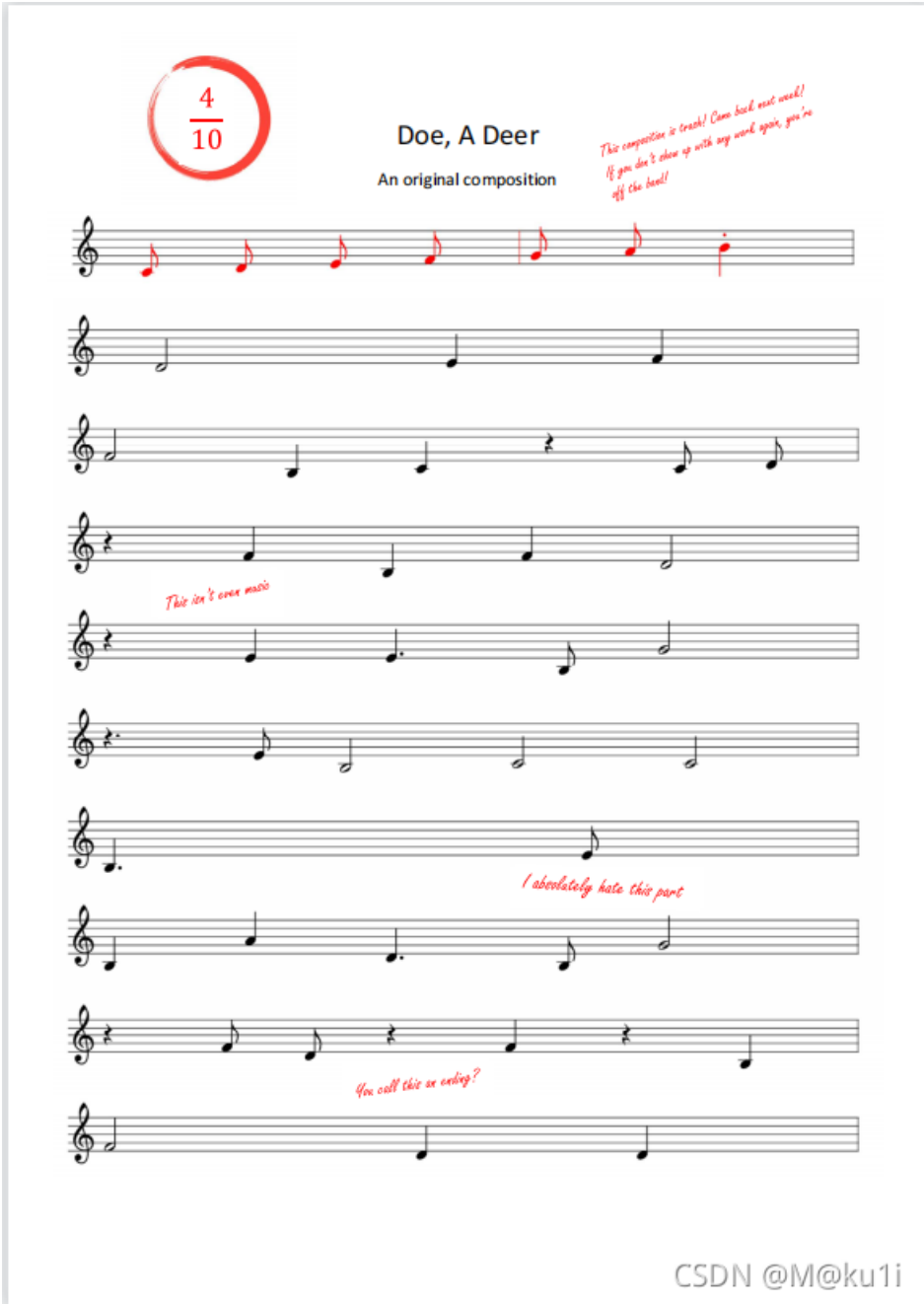
根据这些条件利用网站https://cryptii.com/pipes/enigma-decoder进行解密



flag：dsc{turinglovedme}

#8 Doe, a deer

附件：



一个关于乐理的密码名为Solfa Cipher。

解密器：https://wmich.edu/mus-theo/solfa-cipher/

关于乐理知识真是一窍不通，放一个国外大神的wp吧。

https://ctftime.org/writeup/30726