CTFshow crypto

原创

梁冠希 于 2021-07-01 16:53:13 发布 520 收藏 1

文章标签： unctf

本文链接：https://blog.csdn.net/qq_52429984/article/details/117426346

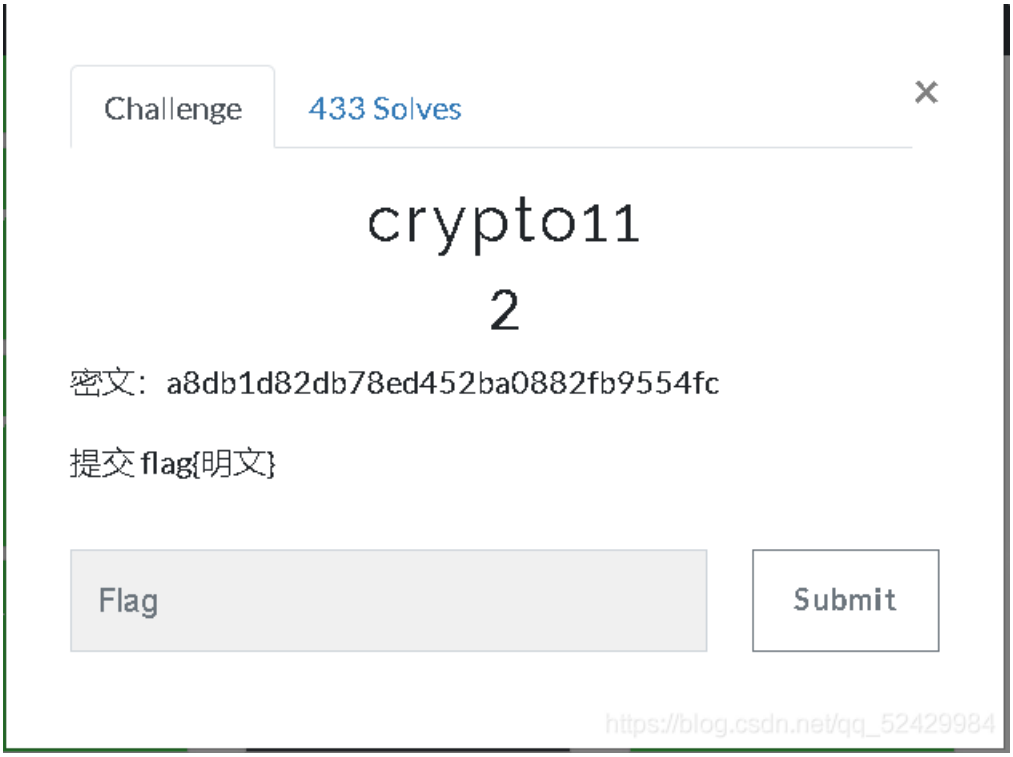
提示：文章写完后，目录可以自动生成，如何生成可参考右边的帮助文档

目录

- crypto11
- crypto0
- crypto12
- 萌新_密码5
- find the table
- babyrsa
- easyrsa1
- easyrsa2

---

crypto11



a8db1d82db78ed452ba0882fb9554fc
尝试md5，失败格式有问题，数一数这里只有31位
md5解密：https://www.cmd5.com/

crypto0

看着像凯撒密码，可以口算也可以使用工具

flag{hello_ctf}

crypto12

http://www.practicalcryptography.com/ciphers/classical-era/atbash-cipher/

萌新_密码5

群共享找 dumpcode.py

find the table

这个题目很有意思，你把他的属性打开你会得到一串数字，我开始的时候不知道什么意思使用了很多密码的解法都没有想到，后面是其他人告诉我数字是关于元素周期表的

babyrsa

直接用脚本

```
import gmpy2
import binascii
e = 65537
p = 10404683571266406477919473497427118563553892788988061192993193971100130156168227017793162297464278992091890256336129334543405576429361244688838391280714339400901980347181644892396963798067122111111796522740242963493548186870116652235057036472787328333237198686019424573942350856678366338061914243182086105117 9
q = 14017104807410798860577373167101890181392813058242288979773207152973309170384371085928226776378346173824295809861094912035449798794591102117084245755218288013364271130722707213381225334112983041615845049925821696787985758156538089078839506813003393118039592648243115029588092648008631773345739257393141022050 1
c = 47727589112047710280490206707783367995687789300728410840578098676080227326112953050960524306418815507811417764989040055898738309733018985236447449515453454045784661767250302904216493449369524802549029394172151482057357307548084673516399434748162809802304470974446824892230544995241979097198573005971574060750692043150227038944662261795076270708354282260865097677467593538223028093850477632928915436972770970684065129247964093932899827380710190473939729592289191158218628680570031454010725811159896806860736632597715874452506870602409912651439198579620477183440177418789258678004315563117856254690017713708524742921941
phi = (p-1)*(q-1)
d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c,d,p*q)

print(binascii.unhexlify(hex(m)[2:]))
```

easyrsa1

分解n后使用脚本

```
import gmpy2
import binascii

e = 65537
n = 145592552973435810546140653225991179080734761646499106530184 7
c = 693803710579142461926067606861522332256595033663193320650 09
q = 120114705943853078683536519456 7
p = 121211263707786291719219191384 1

L = (p-1)*(q-1)
d = gmpy2.invert(e,L)
m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```

easyrsa2

e一样，n和c不一样，求出两个n的最大公因数p，然后求出q和d，最后求出m

```
import gmpy2
import binascii
e = 65537
n1 = 236865639255375777530472290407542829533522217241544953906873588777753801476051524555379885634907169438725175932128583261468115111033118657530183291093146237022070738828842513725532259861120068271113515010449722392722006168717163252654161150388908051148293151119503191831895912838217932379990444278879345368358135267487596129631033778030899006625093995698197855714928281124373126592298798061687588436032488236298218510537754586519339521839884821639500392484872704538882884275403055428241799517344120449853648665321248037460081397630818867813614883046665754566804118065050949634254011755104168649296012205561585694437 47
c1 = 162748414223789761394460782826898119391141740806482454071194519203564908810413303814740022407058841033519066268223118999758008468042420949530307806120512284890464831921964658872099401924927986346298101532948372474782399151371417247888630670329004487178115839330414730105870600379335784692208699495276348599928274159520400866384796353942209634339146452706859904694627930903721285993130333550745514600139032655066853166549324529383900983246866839082028266498406639905140322799006803222638222217347807850588238749583237980643698405005689247922901342204142833875409505180847943212126302482358445768662608278731750064815
n2 = 222576053205255840781808890735232239739241929843538471371646051869566296759389295853863923276720655243381764024964140140838164465088605308877425833388803174788625123066330616015104049600951439413208471605620505240728602117725224784947422136438900274439921833626789704260467656309466443390931491391433887527949328069565898845035691752268504192710953367984562388990098831007935157445799458544814301948793607653462364180193846440952572428116293931644024982610660773393048752122508979184204278140001427512828059806320898671085253354880189400916986098909952524130070737258503960762720271834222976846675657120221990542897 11
c2 = 274260069544183655946955370283109837594864191540910697615784037797812391200739875362346111265979620991886698548047191139336279775362447953764680251042041503946183211801884903058067524981757692685836354168313577723932200274182014594428610917206625984376675579525591318990240364472113855493599143989385058967784963926308052859919759570592753543094246318489168941007805909047468269488642002223065766115799387593160093276382461877342007727361710629766019517992201887539917434686340471042016649701719642458611653591571296514714177502654987063632819569077425999018928666584464128910847483497371073042610504731895930799506 2

p = gmpy2.gcd(n1,n2)
q = n1 // p
phi = (p-1)*(q-1)
d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c1,d,n1)

print(binascii.unhexlify(hex(m)[2:]))
```