来看看自己是怎么日自己的

呃。。。这题吧 我搜了一下`hctf{`还有`hgame{`就出现flag了 虽然觉得这解法一点都不正经 但是也懒得找正经解法了

```
try 'qqvvqhgame{sqlmap_Anddd_wireshark2333}qjkpq1' for key 'group_key'
```

正在前往翻车大道

看了一眼UA sqlmap 然后搜了搜`hctf{`和`hgame{` 发现什么都没有 看请求应该是爆破索性就把所有请求提取出来 放到了Linux里用strings把可见字符串提取了出来 URL解码一看 发现一段带有ctf.flag然后复制出来 贴上最重要的27个请求 前半段都是一个样子 就不复制上去了

```
1. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),1,1))>103 HTTP/1.1
2. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),2,1))>99 HTTP/1.1
3. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),3,1))>115 HTTP/1.1
4. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),4,1))>101 HTTP/1.1
5. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),5,1))>123 HTTP/1.1
6. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),6,1))>101 HTTP/1.1
7. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),7,1))>107 HTTP/1.1
8. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),8,1))>111 HTTP/1.1
9. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),9,1))>119 HTTP/1.1
10. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),10,1))>101 HTTP/1.1
11. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),11,1))>113 HTTP/1.1
12. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),12,1))>95 HTTP/1.1
13. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),13,1))>115 HTTP/1.1
14. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),14,1))>113 HTTP/1.1
15. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),15,1))>107 HTTP/1.1
16. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),16,1))>95 HTTP/1.1
17. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),17,1))>105 HTTP/1.1
18. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),18,1))>109 HTTP/1.1
19. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),19,1))>105 HTTP/1.1
20. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),20,1))>101 HTTP/1.1
21. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),21,1))>99 HTTP/1.1
22. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),22,1))>115 HTTP/1.1
23. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),23,1))>105 HTTP/1.1
24. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),24,1))>111 HTTP/1.1
25. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),25,1))>109 HTTP/1.1
26. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),26,1))>125 HTTP/1.1
27. CHAR),0x20) FROM ctf.flag ORDER BY flag LIMIT 0,1),27,1))>1 HTTP/1.1
```

对着ASCII表看一下再看一眼前面其他URL就出来了 hctf{flower_sql_injection}

进击的 Crypto [4]

一开始我并不知道这是什么东西 并且妄自想要以我自己的渣渣数学能力解出来
直到后来 一百度 怎么这么像DSA 照着网上的分析写脚本

```python
import gmpy2
import hashlib
p = 190716027447646792844189733681358379327343061199369341384474902557309769682923254674220284045951117393868651638379541511185212421611653965460100979887725583476037189369599999668327299529542494009290049992057157747007152994025186591269943960410319186487203043168774653327128061548663247131284489765017
q = 930788704028200015275140127068138499329817310955
g = 220237156062724657086413463831954488565703875839368639153162645824259735475648417436812689270045835305613201869418328189101417845912386944695747737570415063979393239590022180767575736652824918336764694014421673518178065244519367602124516716571334763354973717129155502536345229391073998560517516716958
k = 209569583662944399958807472421680653
x = 91403938517864943293870804762109355181345955543 5
r = 56875265362848301484954914290933136211525478820 6
sdata1 = 427262976273228083221871998313131945010029561209591706262118913937489577133576413685540380226864
sdata2 = 835940898148680488372488685713345793755099380413493862399556052721366535745667186387858109315383
s1 = 61815989378704830075259280288446715538875969669 8
s2 = 659836539307844663175437862395252943516139307036
ds = s2 - s1
dm = sdata2 - sdata1
k = gmpy2.mul(dm, gmpy2.invert(ds, q))
k = gmpy2.f_mod(k, q)
tmp = gmpy2.mul(k, s1) - sdata1
x = tmp * gmpy2.invert(r, q)
x = gmpy2.f_mod(x, q)
print int(k)
print int(x)
def data_to_int(s):
    return int(s.encode('hex'), 16)
def SHA1(data):
    return data_to_int(hashlib.sha1(data).hexdigest())
def encrypt(data, p, q, g, x, k):
    r = pow(g, k, p) % q
    s = (gmpy2.invert(k, q) * (SHA1(data) + x * r)) % q
    return (r, s)
def getflag(data):
    print 1
    if data == "getflag":
        (r, s) = encrypt(data, p, q, g, x, k)
        flag = "hctf{" + str(s % r) + "}"
        print flag
getflag("getflag")
```

flag : hctf{881691912314398184476813935100212817302692520 95}

进击的 Crypto [5]

低加密指数广播攻击 数学太差 中国剩余定理看了好久才知道怎么去写脚本 古代劳动人民的智慧我表示望尘莫及

```
import gmpy2
from gmpy2 import invert, iroot
n = [
17551188754807399016342420221734945766749930201727412345251590531404061
48074093299519906533298771918319719944833643585101554027215544148674602
73151078213409691056234515758725801709783686505938802740760255204539563
78539766228430429142117994616985318963125739076826635459215114946159348
67839826809938952541443161851774388931408571439043097278322327098568898
89952577703631172254725900844891209583971891309584627845326374534821827
63505791720641422686749616216521724012108378680177131928438795893866425
04048955125890261004757457989969283498499920952811084934204252982559406
18942383715919103545843051361543553651912154001 49,
21840437284422601584177601857355845296420300157767339109572377640408362
72667456124621040040076047418712124689371248010932689361416277947076841
52829007138009838155706022500686736950447499325326898513107700875525246
20857623442019428044482787965428075984951982104920901562456426672111590
90910073620915390585303512724772027697405040466384754709036278072907782
95946849959794573264149474490832194616173026433929045227338129031413013
32227357689517880801845510579518887472928083718826890369016944549160461
42871061574700127352125363876626714334181917024964838534112726370741764
29254648097889757153329386664173166959413474775 77,
30015914133986758133105015082922460910471726819000479872816812806794140
88720929439396306327337789120306986471146677620010817367242877929330832
04601164930405728269150206542939292418436867282963874001100620990745730
09645563520805301899962344680821396200256042478539540675850427377958553
32458137136211265275244482491960176782162203848836550103163655823612605
26542416627430706512133695668989988986344412618116550537893845887115476
85939927721116404429914329081432990913321296964957260691148704710581681
77456290492408081327498480994000896862951530485951862947328401833265183
15879164025393862424216302505370306881626757517 61,
18009718435825445649372629634867772247035138229493108362713630947680338
35422773557207088239037863244036516311709423341352010707058190822423921
09691720944027609240553345842596782765059194181916237629982497242483793
02786621689049107500760348303940112840664926231345646325964133281550765
45471962816160047514379256730994733013074786012555754705184362089962921
76369180029298464630978415394460140195798303473560840804885619174403568
48465812937246809018168582635191619622041079012450531811237168105634382
10863413588088709352561126318692521302791333745470691975492359371450999
80620734506797209055694893043986252551438015332 77,
27090736422393991189249636552945539144039087911497773160371557650625344
53325458076434462854051513288457673974659772907914615513089900923811010
16547113038003405665382987984329291365099231290899046470234091462644059
64139002638684510681372633714179570768685640570209727600215295330846361
75431497373175373439731293294743322573259752407656360572903799827280347
29530799128998672443180731445643553265202300786811067466708956434549397
14423661018216469020021429142336238301775948794784776906058601395026463
84207075554779319247065320407822282776895082374706165510690027657154768
04519535623143769135928964277304730913600513911 29,
22277916445389799876692954866506052125036892596099795492064670519272419
62152875983731825366529279127861537748967309681231938516330289846519214
66611382992050824219689223502301215265300809080532978790271585450401937
```

```
    746473504994158632111137767853998484727122935352891138103223981039835874233062826298483673287430891459405369133906646973215263060545796566004391400616672645160700035078416183521225673790383042949767254693052156824268987316990937949304173667692492569991773398430203643677697127762257439161121707872738912259262464072010802025938580633625659319425903491783610853967998901485059594372449834479348387920517932085100525470493395835481755297519058838396284382712222690496404831805324304982227704971550573576205153059294051400768713888255136990671178713343279247844746524140944914562547076686728054567331371087782749101096628587174763840131533861698678237064188161523843466768793640095562964262974898783999601155040815616231720113974645314887455860303475690229706683559274856201410003305073665035951249763320635087862042105884086417865415028745257336325154309073340438903724184386937954371980449964060054621605181257533507829220343507505674090735647650842356289315995804144395122093369406451027596426472360089455862342181990450121745680718194037861694569863678082853797244380310200176477643943644972463871360669070584682807703870684830948023158889780219716829353665600564791257912082043905122048004593803193375178269942973208249276102769671038752489438400731535367257356207098737711096953679231300720903502192144476519811856646400822239966219746455729409375771797064825552958482297097797911599808242875799342438899636328763766908154658738969614707377852147843953614178305416594819846812426828300689497046389665982817209315784313108617753052015326937047712513569322963102500122038505770232917473260057408981125469607573191926134043719437868156273,
12. 232574830423317810313200040663959730985398818704330344981806282921648254768456475961228897563969872207872634787786471990335232488610794879912560444736445159605596307921463940398140964084865415880676438110777164280150003100062278155638531616041537996670732628865064757929023928295714723546918751084975327216860769713715102381610605538588390910264512665217538300841432455174598613258809064312053285417369487847279095182590347814328253175880168292480467495542454226729231214524705523234388085911180500341083257545351342367277404601908492526475618262036756666014415060247806051943407778023899601805328318786894580960468 21,
13. 276370046223273380309881579061803246678299167513589776408327653286457186963854820917815131974720660521017041317511586272396574254739944411433856131055282002152751104662632899529624006642936021338568094752959703227673095632739993199261506483995225085925216856888968428332433240709875945881347765159781983649014557272920060586240156749141314472320320588754109561149399384601928761576124577403392212555327180940938118113466869668284758331469875933780281487616175133370182617665537212674671753322829218992864506411195982489567951747669955631394872481886090600677480894480298447751778839126014950482239627659345170711825 7
14. ]
15. C = [
16. 467518260554971134765329951477782623855904020052774781084690199143312702572336180721167220805286287025804728528594921251566427841049960382966326621396851149397476667407182522553666526704904461934037843853161598369640630584194575639657937179482626554992950304313466836005502727652236887919362093745788148650931171190572248318773465012779651080863789018995284097133076891394688452559462765245874593053221923825891160874411200179813639480600472647289083848188102671660574381267435095525800379459074825852392946044698602079444547807655963514518190909378661771647387060446 3
```

```
    3156961830504951435027626147674858819792175774 16,
17. 458603393001581497555348761432134129035807296053956484958975859380158782341439435186207409334325547963530319907856190526079364586994262213616461577682286727517288881298907101668765802300583877618143747020452833544441385256189560337795595930831839924696321396404498423597435282406062597058711958661277432238521681026481942645864185216600989910665503675591332228321973974102065681203434840409201834027702684411427269917032301405008728971540398967191876771261341434632022274532880009661594713866713905333948327831952120198388542349190517116767550156355656389082175689247499954963565048694401034655464160334323849057268917,
18. 120125753423664422109943686050325821296744853270060935529029838779572021727839380716848410319023961568993968791363886069968076082963167956289878773575472134503609537429477817005495064692265649759273617485965208785426516689631310212698519281746810073477715195699146909807449419650867622856300827810137818795116813531191172803541022714390572040664050723284220577218959835404921504023095526090350082723661766573841740812297987390995585390830744895003590620101643583000871244470379228447098704302180196739143057429399278888385996165203000119130227920603922606180468077883026437428748470241531981137926280466785787536329233,
19. 424900591132489863045872349115157681019861907833297291142016664512725859851995044850764090437934281748642626205488919936938525781252475976124558897365157821129134432661931755937736031950772772953508314329333579958819697624194921803783374783954529503025911527966806588656041787720405468449504718111707560346867229226883127373595696406762683797521853984066764188562142660794195030633019889718768375468737961184427847157279181639795063438890723331211792063480798973401113271833433485059211165163132313093343800700898348978260182732139616810466849333262655891183872199817314850899477147322405082747118647810689694941383 1,
20. 185571098538984059749247695501053456737030674575378136072521514699331407607983785742444395593721004289714000703511731403488236527558256014332688323635758961373642880699843141513461372061155183095973892688751505496963768132777785145915326157942946370446266839313675101816123836819144217022615922332654559500232524076806040410466296677032071331468536189891875977491326505140385842804177911344265968483694176260395337408275545521177275015658410650778353989518267478553935527314974389832742015997401897436309491746107221186590196309646699014386900280702137108333356225735264035887370928692739695816076375225727570152375 06,
21. 192215313427138012199714204550986663651248101695127110304440639423336945623641141723615521391765824761681342199373934683913465350979657637859164840126574010950668048579009414827726871594714839081076698924196260596801687621199540682887193035914985040506046286272996711602252763556365199615781184139130010587441645314706102791687234020094699593779413871520486223997356149479621362476935890672618676971168996280893782855339940317376203454897433596010770063891023165879563916790670077012074197441849519682783046254213008096555210241618051760039789706511378827909606679253380710418786867884885778073533474986581301968190 9,
22. 204386671853292730145412701752183594540172717376035248822051702849886177726280703656616518928573056937508340623421292158865409842129008826955135210183068842990360088069104660790239656677500569414831560561643771311542422359471018667784486727321844335047925327923196006184571189826246118597005700213852326295226083419116943876496475852687736144227766444859612945149641569577678187926673205792058333699976663393858265624060797375
```

```
    975602103248926238119076282508139261368981779276484440987711624022385237631140416496359504034717613949446131949528534860856258990167162006362068852424952370050014460455889927220256453529546623,
    901840645204111786792720467464922092916050994260429057230364418345123794491247023636730826631907934928962587344764720047459875586964299921314275882580952247481081760975118753334690266722503245269418716226268305644577545596399423720229610481238637463627018843352192750341550796347724091327087979414117056839302536789100392109001599394494620572360134914960125672743371615512188342027149826596723425906853517250144564375269186099748063610169934074026550576709192066821801800297663211125643065561906124729202565474088907701417925864708469822973316007130639002060363621073636319830240728919528695398052028503252861449292623,
    179786719197345952015241866188686596560367655466738839388193339115649485878588764958026002440784133124628139386820907615999616049596094230109041877207485397071728533525403702089919029290020367753815033651701968386263796857120507288751726226324478753681667437511597756116426294085348906841096526484848688767106972493518953551115112810374973477122412429085636997072728430327731074773687904191099072710963514575262404719751222964836553371501300008128130812391816533844191492929703209918920733523406492373874064980877347081620944760923686657943509689242892591928932893370382015040663101732847570448024496006660078215095536215040663101732847570448024496006660078215095536,
    859706788081245612366959497866013505866887583492420127992416276102013758266236770450498857661493496606232319166698294311365618678389424503526705175617833149088126462977463562017967800094777192885456623921349858607867087012792469645904129584847407387933650132543719456384541469936056457113668349242629276679912115504127616267342770099976105559260921360315857663721200377100399672274317054334288501128969284549359841396568894932567051974341307147905738260222205590259361415942718601779799221949030893011685008455303346131650945783443642082191609420601020622937027362268141317634718725295008742886829596552872832409316
]
N = 1
for i in n:
    N *= i
Ni = []
for i in n:
    Ni.append(N/i)
T = []
for i in xrange(10):
    T.append(long(invert(Ni[i], n[i])))
X = 0
for i in xrange(10):
    X += C[i]*Ni[i]*T[i]
m10 = X % N
m = iroot(m10,10)
print m
a = 446132467216522755969476211244757715855688689177985234184043021698656737262679647328602454718818107572358988636907247081112591957598531435271500424867817849875364197565207396857185829997005412827718155553981667749248837771875914914441321747210051854563126256338238698588297998296414456610979161719953152893577800386649140386124795211777334321653826403788113127081985630817569961006602125352075092710567132712905679043715514
```

```
    491874983609008809853
b = hex(a)
print b
```

出来的16进制长的 我都以为我做错了 结果是：When e are small and same,it can be Hastad’s broadcast attack.Maybe we won’t have topic aboout RSA,but I wish you can explore it Non-stop.hctf{Hastad’s_broadcast_attack_is_interesting}