week2

crypto

LikiPrime

1.题目如下

N =
19858023172592350347056092934289108867332559820014412422904873391017368390491204441405019301757692163358568720772222990956725956851185748993194247509341282944599494869052187136538318228030158498819690453401127435361901336085426078504784681943364531711065174975983716420495028746384606845785372939301089730005926566463747103432059044671872242955497981523513934676966435965071546243806229757535166637230488729730728485385658213056705925975131014188636902766933432089015682333389267431091292884487115372205039490382803937401421955411982947059980331709717554525578778134753034391245521690419747757535275447930878225939266156940283052519288176494757285984852790145769896988098538568229269554739365579557338929030174465367637802427627628041108890438305307944788900351181440725939031809614137323649939205464659496179449690542821116994466286941853061704449032316936940978636790571298138464629674731309967874023341286875808653891416653395583192716151770642345403903682388754568911009454996351355739925397589533036571720163865356105479057579120798919721895825183557226594989393083395316606681078156623090006543260085763484712778521846926037690024181529089469615353825182375937199057383908496512104767426855118452562909945699059975582317479377046459220417679539920947555666471967730972034108527168646157106169169591007731130524128501044920760911243861775289031493068199408329039530243835636783220388605757251354908363912525368074079043467996590429183134900581082770912108435816093457842117274119056399170358154605791872508725859025206171689073921537631984373303686386023481548970304079134276396722669360891835696817131202284900958041416285996284089571628212958506559053863321 7
e = 65537
c =
46560154884809765083707934824951737492910520961847248642247576535415933606600354943507345213418873971535153678758379092046576390914157141767599741378856888002322824869461673005884913142257151106875746732838112867181526939380130676381000287219835865853313401143369743979686206309354010039446738250361360436274876780221682796540482633488477890372362891180704238733618760822798850487276433758726387380715274302497925507090799650380370669925517766943172539218126894034368535974224452292395161347948106629595049811210486351660852406502000462937766137201804824824970577433643912520584635476029848964799102383058326398381207468892239097516353981419440417417811587181781525422807486955527864632609897203586877283762657651238195100457021725745284934898009102000732615995587955568528970327723009020228122460110436865183321860469134882124768113376932612488397995351843621864773720771234236761810676128914013731767402199176029141859192721974809311080637300155097379517174585228639401602786943829502320041639846647104014709626675162122570433483276407269675261359920093490834460727475905628672909166993858163688904886776016254475729977560810076674061462255325050794836642329677984981574835731942521831975493673580734682146167413947001619504217865219182733031415214875026481826505779535495194794700583548292037330993457150815216034031048768712937426163515976

579293793907835538126685201143898444073290402247882920531199789972398829749115009864217148871228807870509404433779144971473999778494139950609523201414929394940002112382066618660911536712614461380941584965721399494124569331460756831869322717020469587610914049647179395693234452929902635227089504498101133030249746753610798 7

2.查阅RSA定义

RSA算法的具体描述如下： [5]

（1）任意选取两个不同的大素数p和q计算乘积$n=pq,\varphi(n)=(p-1)(q-1)$ [5]；

（2）任意选取一个大整数e，满足$gcd(e,\varphi(n))=1$，整数e用做加密钥（注意：e的选取是很容易的，例如，所有大于p和q的素数都可用） [5]；

（3）确定的解密钥d，满足$(de)mod\varphi(n)=1$，即$de=k\varphi(n)+1,k\geq 1$是一个任意的整数；所以，若知道e和$\varphi(n)$，则很容易计算出d [5]；

（4）公开整数n和e，秘密保存d [5]；

（5）将明文m（m<n是一个整数）加密成密文c，加密算法为 [5]

$$c=E(m)=m^e modn$$

（6）将密文c解密为明文m，解密算法为 [5]

$$m=D(c)=c^d modn$$

然而只根据n和e（注意：不是p和q）要计算出d是不可能的。因此，任何人都可对明文进行加密，但只有授权用户（知道d）才可对密文解密 [5]。

3.显而易见的解法是因式分解出p，q。通过http://www.factordb.com/得出p，q

4.查阅前人的代码

5.运用python，代码如下

”

```
import libnum
from Crypto.Util.number import long_to_bytes
```

```
c =
4656015488480976508370793482495173749291052096184724864224757653541593360660035494350734521341887397153515367875837909204657639091415714176759974137885688800232282486946167300588491314225715110687574673283811286718152693938013067638100028721983586585331340114336974397968620630935401003944673825036136043627487678022168279654048263348847789037236289118070423873361876082279885048727643375872638738071527430249792550709079965038037066992551776694317253921812689403436853597422445229239516134794810662959504981121048635166085240650200046293776613720180482482497057743364391252058463547602984896479910238305832639838120746889223909751635398141944041741781158718178152542280748695552786463260989720358687728376265765123819510045702172574528493489800910200073261599558795556852897032772300902022812246011043686518332186046913488212476811337693261248839799535184362186477372077123423676181067612891401373176740219917602914185919272197480931108063730015509737951717458522863940160278694382950232004163984664710401470962667516212257043348327640726967526135992009349083446072747590562867290916699385816368890488677601625447572997756081007667406146225532505079483664232967798498157483573194252183197549367358073468214616741394700161950421786521918273303141521487502648182650577953549519479470058354829203733099345715081521603403104876871293742616351597657929379390783553812668520114389844407329040224788292053119978997239882974911500986421714887122880787050940443377914497147399977849413995060952320141492939494000211238206661866091153671261446138094158496572139949412456933146075683186932271702046958761091404964717939569323445292990263522708950449810113303024974675361079874
n =
1985802317259235034705609293428910886733255982001441242290487339101736839049120444140501930175769216335856872077222299095672595685118574899319424750934128294459949486905218713653831822803015849881969045340112743536190133608542607850478468194336453171106517497598371642049502874638460684578537293930108973000592656646374710343205904467187224295549798152351393467696643596507154624380622975753516663723048872973072848538565821305670592597513101418863690276693343208901568233338926743109129288448711537220503949038280393740142195541198294705998033170971755455255787813475303439124552169041974775753527544793087822593926615694028305251928817649475728598485279014576989698809853856822926955473936557955733892903017446536763378024276276280411088904383053079447889003511814407259390318096141373236499392054646594961794496905428211169944662869418530617044490323169369409786367905712981384646296747313099678740233412868758086538914166533955831927161517706423454039036823887545689110094549963513557399253975895330365717201638653561054790575791207989197218958251835572265949893930833953166066810781566230900065432600857634847127785218469260376900241815290894696153538251823759371990573839084965121047674268551184525629099456990599755823174793770464592204176795399209475556664719677309720341085271686461571061691695910077311305241285010449207609112438617752890314930681994083290395302438356367832203886057572513549083639125253680740790434679965904291831349005810827709121084358160934578421172741190563991703581546057918725087258590252061716890739215376319843733036863860234815489703040791342763967226693608918356968171312022849009580414162859962840895716282129585065590538633217
# n = int("",16)
e = 65537
# e = int("",16)
```

```
q
=190797007524439073807468042969529173669356994749940177394741882673528979787005053706368049835514900244303495954950709725762186311224148828811920216904542206960744666169364221195289538436845390250168663932838805192055137154390912666527533007309292687539092257043362517857366624699975402375462954490293259233303137330643531556539739921926201438606439020075174723029056838272505051571967594608350063404495977660656269020823960825567012344189908927956646011998057988548630107637380993519826582389781888135705408653045219655801758081251164080554609057468028203308718724654081055323215860189611391296030471108443146745671967766308925858547271507311563765171008318248647110097614890313562856541784154881743146033909602737947385055355960331855614540900081456378659068370317267696980001187750995491090350108417050917991562167972281070161305972518044872048331306383715094854938415738549894606070722584737978176686422134354526989443028353644037187375385397838259511833166416134323695660367676897722287918773420968982326089026150031515424165462111337527431154890666327374921446276833564519776797633875503548665093914556482031482248883127023777039667707976559857333357013727342079099064400455741830654320379350833236245819348824064783585692924881021978332974949906122664421376034687815350484991
p
=104079321946643990819252403273640855386152622472667048053191123504036080596733602980122394417323241848424216139542810077913835662483234649081399066056773207629241295093892203457731833496615835504729594205476898112116936771475484788669625013844382602917323488853111608285384165850282556046662248318909188018470682222031405210266984354887329580288780508697361869007147207105557031687290 87

d = libnum.invmod(e, (p - 1) * (q - 1))
m = pow(c, d, n)  # m 的十进制形式
string = long_to_bytes(m)  # m明文
print(string)  # 结果为 b‘ m ’ 的形式
```

(代码转自https://blog.csdn.net/vhkjhwbs/article/details/101160822)