安全-RSA（i春秋）

原创

小狐狸FM 于 2021-08-24 09:29:51 发布 147 收藏

分类专栏： 安全 # CTF夺旗 文章标签： python 字符串 rsa 算法

本文链接：https://blog.csdn.net/smallfox233/article/details/119882177

安全 同时被 2 个专栏收录

91 篇文章 8 订阅

订阅专栏

CTF夺旗

38 篇文章 0 订阅

订阅专栏

文章目录

一、题目


```
This time John managed to use RSA " correctly "&ellipsis; I think he still made some mistakes though。

N=0x1564aade6f1b9f169dcc94c9787411984cd3878bcd6236c5ce00b4aad6ca7cb0ca8a0334d9fe0726f8b057c4412cfbff75967a91a370
a1c1bd185212d46b581676cf750c05bbd349d3586e78b33477a9254f6155576573911d2356931b98fe4fec387da3e9680053e95a47099342
89dc0bc5cdc2aa97ce62a6ca6ba25fca6ae38c0b9b55c16be0982b596ef929b7c71da3783c1f20557e4803de7d2a91b5a6e85df64249f48b
4cf32aec01c12d3e88e014579982ecd046042af370045f09678c9029f8fc38ebaea564c29115e19c7030f245ebb2130cbf9dc1c340e2cf17
a625376ca52ad8163cfb2e33b6ecaf55353bc1ff19f8f4dc7551dc5ba36235af9758b

e=0x10001

phi=0x1564aade6f1b9f169dcc94c9787411984cd3878bcd6236c5ce00b4aad6ca7cb0ca8a0334d9fe0726f8b057c4412cfbff75967a91a3
70a1c1bd185212d46b581676cf750c05bbd349d3586e78b33477a9254f6155576573911d2356931b98fe4fec387da3e9680053e95a470993
4289dc0bc5cdc2aa97ce62a6ca6ba25fca6ae366e86eed95d330ffad22705d24e20f9806ce501dda9768d860c8da465370fc70757227e729
b9171b9402ead8275bf55d42000d51e16133fec3ba7393b1ced5024ab3e86b79b95ad061828861ebb71d35309559a179c6be8697f8a4f314
c9e94c37cbbb46cef5879131958333897532fea4c4ecd24234d4260f54c4e37cb2db1a0

d=0x12314d6d6327261ee18a7c6ce8562c304c05069bc8c8e0b34e0023a3b48cf5849278d3493aa86004b02fa6336b098a3330180b9b9655
cdf927896b22402a18fae186828efac14368e0a5af2c4d992cb956d52e7c9899d9b16a0a07318aa28c8202ebf74c50ccf49a6733327dde11
1393611f915f1e1b82933a2ba164aff93ef4ab2ab64aacc2b0447d437032858f089bcc0ddeebc45c45f8dc357209a423cd49055752bfae27
8c93134777d6e181be22d4619ef226abb6bfcc4adec696cac131f5bd10c574fa3f543dd7f78aee1d0665992f28cdbcf55a48b32beb7a1c0f
a8a9fc38f0c5c271e21b83031653d96d25348f8237b28642ceb69f0b0374413308481

c=0x126c24e146ae36d203bef21fcd88fdeefff50375434f64052c5473ed2d5d2e7ac376707d76601840c6aa9af27df6845733b9e53982a8
f8119c455c9c3d5df1488721194a8392b8a97ce6e783e4ca3b715918041465bb2132a1d22f5ae29dd2526093aa505fcb689d8df5780fa174
8ea4d632caed82ca923758eb60c3947d2261c17f3a19d276c2054b6bf87dcd0c46acf79bff2947e1294a6131a7d8c786bed4a1c0b92a4dd4
57e54df577fb625ee394ea92b992a2c22e3603bf4568b53cceb451e5daca52c4e7bea7f20dd9075ccfd0af97f931c0703ba8d1a7e00bb010
437bb4397ae802750875ae19297a7d8e1a0a367a2d6d9dd03a47d404b36d7defe8469
```

二、WriteUp

> RSA公式如下

n = p*q

φ(n) = (p-1)*(q-1)

c = (m^e) mod n

d = 1 mod φ(n) / e

m = (c^d) mod n

> 目前已知的参数是 `n、e、p、d、c`，需要求的是明文 `m`
> `0x` 开头的数字是十六进制，需要先将其转换成十进制，使用python代码进行进制转换 `int(字符串, 进制类型)`

```
C:\Users\86138>python
Python 3.8.9 (tags/v3.8.9:a743f81, Apr  6 2021, 14:02:34) [MSC v.1928 64 bit
Type "help", "copyright", "credits" or "license" for more information.
>>> N="0x1564aade6f1b9f169dcc94c9787411984cd3878bcd6236c5ce00b4aad6ca7cb0ca8a
56931b98fe4fec387da3e9680053e95a4709934289dc0bc5cdc2aa97ce62a6ca6ba25fca6ae38
af370045f09678c9029f8fc38ebaea564c29115e19c7030f245ebb2130cbf9dc1c340e2cf17a6
>>> e="0x10001"
>>> phi="0x1564aade6f1b9f169dcc94c9787411984cd3878bcd6236c5ce00b4aad6ca7cb0ca
2356931b98fe4fec387da3e9680053e95a4709934289dc0bc5cdc2aa97ce62a6ca6ba25fca6ae
ec3ba7393b1ced5024ab3e86b79b95ad061828861ebb71d35309559a179c6be8697f8a4f314c9
>>> d="0x12314d6d6327261ee18a7c6ce8562c304c05069bc8c8e0b34e0023a3b48cf5849278
07318aa28c8202ebf74c50ccf49a6733327dde111393611f915f1e1b82933a2ba164aff93ef4a
c4adec696cac131f5bd10c574fa3f543dd7f78aee1d0665992f28cdbcf55a48b32beb7a1c0fa8
>>> c="0x126c24e146ae36d203bef21fcd88fdeefff50375434f64052c5473ed2d5d2e7ac376
2f5ae29dd2526093aa505fcb689d8df5780fa1748ea4d632caed82ca923758eb60c3947d2261c
3bf4568b53cceb451e5daca52c4e7bea7f20dd9075ccfd0af97f931c0703ba8d1a7e00bb01043
>>> N=int(N,16)
>>> e=int(e,16)
>>> phi=int(phi,16)
>>> d=int(d,16)
>>> c=int(c,16)
>>> N
432103260362901062510888102986679157020077445674678549881079372336228216717521
921826981422737611820470111192617799883774082088581095031774131945415245434471
833102258133221776576952569943150482709523150011005489495543239652404643005206
903070810100867296675525698283233540084413356742511
>>> e
65537
>>> phi
432103260362901062510888102986679157020077445674678549881079372336228216717521
921826981422737611820470111192617799883774082088581095031774131945415245434471
019519175138479519249363475746712127896942227280870285190901941312509993585181
894857318003820179805577103650982799493828316819521
>>> d
367456229405453438757599764341571978867555063587609822573085670370060743917191
802954992273860759295946415566580332073828480750733197862441611938017951059011
051922466222404757111334440549853675205645424886971676064808382947477103964011
423153150845917668811883716689451353912904767581451
>>> c
372098770261388243023016972923193281858240599125066447190412738959727479266981
158817619492976139678126839486220680203827712056099752204071190229666006754691
958022551212849115178822680927852469859816879133109656287931787034989616176281
796564954722997474017946357818479015881560994950175
>>>
```

```
N=432103260362901062510888102986679157020077445674678549881079372336228216717529305833785588083811970365429033965343373057234961071540503249535365309937442673860081206589102421439926567733607921826981422737611820470111192617799883774082088581095031774131945415245434478277778314099036647495205273755147486784621012614450399004089994299695107869604718871589576793371900915279133403833102258133221776576952569943150482709523150011005489495543239654046430052067327773613866235915444615337972636527805956392881588921222196267204814726164201180394366806472987147680717979412903070810100867296675525698283233540084413356742513
```

```
e=65537
```

```
phi=432103260362901062510888102986679157020077445674678549881079372336228216717529305833785588083811970365429033965343373057234961071540503249535365309937442673860081206589102421439926567733607921826981422737611820470111192617799883774082088581095031774131945415245434478277778314099036647495205273755147486784616840011796810258362432726415200460011083944116083154239671701237095697019519175138479519249363475746712127896942227280870285190901941312509993585181596918641691877238157196576885985768201041234565477069957940010208377566959234765822266226770877276994463234594894857318003820179805577103650982799493828316819521
```

```
d=367456229405453438757599764341571978867555063587609822573085670370060743917197053156667812000656251162031995986336220260704927757436186079666523939964196638533500859142527978877427826615948802954992273860759295946415566580332073828480750733197862441611938017951059010076401742547988318632265442680041499206253959252597156252484170784573171674585924445328250398280137681818603497051922466222404757111334440549853675205645424886971676064808382947477103964010265338201912991168911864742405202539533094741669639563344307988600840724503289317008812447173269029730616674404423153150845917668811883716689451353912904767581455
```

```
c=372098770261388243023016972923193281858240599125066447190412738959727479266985566121944553671723682772688837741027191131948506740129999713375505610616978264584683635744283786791797216725305158817619492976139678126839486220680203827712056099752204071190229666006754690447328912107892482844107394828769378577260264315932839601622987078921439705138048922483704274553184724020115360958022551212849115178822680927852469859816879133109656287931787034989616176286611132772490714905847747645711708913913550800337917456621191391708345293065486447160690251619891038806715800752796564954722997474017946357818479015881560994950171
```

> 因为使用下方的公式就能得到m的值，且 `c`、`d` 和 `n` 的值是已知的，所以可以直接求解
> 使用 `python` 计算出 `m` 的十进制后，再将其转换成十六进制

m = (c^d) mod n

```
C:\Users\86138>python
Python 3.8.9 (tags/v3.8.9:a743f81, Apr  6 2021, 14:02:34)
Type "help", "copyright", "credits" or "license" for more
>>> n=432103260362901062510888102986679157020077445674678
373057234961071540503249535365309937442673860081206589102
088581095031774131945415245434478277783140990366474952O5
576793371900915279133403833102258133221776576952569943150
444615337972636527805956392881588921222196267204814726164
2832335400844133567425l
>>> c=372098770261388243023016972923193281858240599125066
191131948506740129999713375505610616978264584683635744283
056099752204071190229660067546904473289121078924828441O7
704274553184724020115360958022551212849115178822680927852
847747645711708913913550800337917456621191391708345293065
8184790158815609949501?
>>> d=367456229405453438757599764341571978867555063587609
220260704927757436186079666523939964196638533500859142527
750733197862441611938017951059010076401742547988318632265
250398280137681818603497051922466222404757111334440549853
911864742405202539533094741669639563344307988600840724503
6894513539129047675814S
>>> pow(c,d,n)
384365526052440202360459651805033449148582243524328138349
889102850182140300335071766036185З
>>>
```



```
m=384365526052440202360459651805033449148582243524328138349913683453506738455616163926510705066867828115177854736411335061889102850182140300335071766036185З
```

> 转换成十六进制数

```
>>> m=384365526052440202360459651805033449148582З
335061889102850182140300335071766036185З
>>> hex(m)
'0x4963654354467b7273615f69735f617765736f6d655f77
e5f6e6f747d'
>>>
```

```
m=0x4963654354467b7273615f69735f617765736f6d655f7768656e5f757365645f636f72726563746c795f6275745f686f727269626c655f7768656e5f6e6f747d
```

> 使用在线工具将hex转换为str

IceCTF{rsa_is_awesome_when_used_correctly_but_horrible_when_not}