攻防世界——OldDriver

原创

lrving- 于 2021-01-23 12:50:09 发布 876 收藏 1

分类专栏： 题解

本文链接：https://blog.csdn.net/weixin_51867782/article/details/113036245

题解 专栏收录该内容

8 篇文章 0 订阅

订阅专栏

攻防世界——OldDriver

原理：rsa低加密指数广播攻击

拿到文件后，发现一份明文用不同的n，相同的e,进行了多次加密，产生了多份密文。并且e=10，不是很大，这是典型的低加密广播攻击。由c=E(m)=(m^e)modn 可以通过中国剩余定理，计算m^e。

```
    x, y = 0, 1
    lastx, lasty = 1, 0
    while b:
        a, (q, b) = b, divmod(a, b)
        x, lastx = lastx - q * x, x
        y, lasty = lasty - q * y, y
    return (lastx, lasty, a)


def chinese_remainder_theorem(items):
    N = 1
    for a, n in items:
        N *= n
    result = 0
    for a, n in items:
```

```
        m = N // n
        r, s, d = extended_gcd(n, m)
        if d != 1:
            N = N / n
            continue
        # raise "Input not pairwise co-prime"
        result += a * s * m
    return result % N, N


sessions = [{"c": 73660675747411714617220651332429160804955059136632503300827474653836768939704114765507483948414374181053123539710950034243226796169403711230289821895020 42, "e": 10, "n": 25162507052339714421839688873734596177751124036723831003300959761137811490715205742941738406548150240861779301784133652165908227917415483137585388986274803},
{"c": 21962825323300469151795920289886886562790942771546858500842179806566435767103803978885148772139305484319688249368999503784441507383476095946258011317951461, "e": 10, "n": 23976859589904419798320812097681858652325473791891232710431997202897819580634937070900625213218095330766877190212418023297341732808839488308551126409983193},
{"c": 65696894202740669578359833905835852865700876190481101411877005841937926952354050778115443551692903823571493741070764060861541033518978907935989976870539 83, "e": 10, "n": 18503782836858540043974558035601654610948915505645219820150251062305120148745545906567548650191832090823482852604346478335353784501076761922605361848703623},
{"c": 45082461680445135184524938827135363906367415415518058217903389737976159712718672485843798131141254781952846926959286689465536254831796332660571229675470 52, "e": 10, "n": 23383087478545512218713157932934746110721706819077423418060220083657713428503582801909807142802647367994289775015595100541168367083097506193809451365010723},
{"c": 22966105670291282335588843018244161552764486373117942865966904076191122337435542553276743938817686729554714315494818922753880198945897222422137268427611672, "e": 10, "n": 31775649089861428671057909076144152870796722528112580479442073365053916012507273433028451755436987054722496057749731758475958301164082755003195632005308493},
{"c": 17963313063405045742968136916219838352135561785389534381262979264585397896844470879023686508540355160998533122970239261072020689217153126649390825646712087, "e": 10, "n": 22246342022943432820696190444155665289928378653841172632283227888174495402248633061010615572642126584591103750338919213945646074833823905521643025879053949},
{"c": 16524175347090294503805706539737053209861176795975638730226831408005074825604829483101315404948227797045505390333146191586749269249548168247316404074014639, "e": 10, "n": 25395461142670631268156106136028325744393358436617528677967249347353524924655001151849544022201772500033280822372661344352607434738696051779095736547813043},
{"c": 15585771734488351039456631394040497759568679429510619219766191780807675361741859290490732451111264877664812677975936842820519468472151649702629098178623935 2, "e": 10, "n": 32056508892744184901289413287728039891303832311548608141088227876326753674154124775132776928481935378184756756785107540781632570295330486738268173167809047},
{"c": 89651234216376940500442168445233791633474780291248150328328132250507325585242396606487462848841407467888236818860105773422548410145945700674679056823597 97, "e": 10, "n": 52849766269541827474228189428820648574162539595985395992261649809907435742263020551050064268890333392877173572811691599841253150460219986817964461970736553},
{"c": 13560945756543023008529388108446940847137853038437095244573035888531288577370829065666320069397898394848484847030321018915638381833935580958342719988978247, "e": 30415984800307578932946399987559088968355638354344823359397204419191241802721772499486615661699080998502439901585573950889047918537906687840725005496238621}]

data = []
for session in sessions:
    e = session['e']
    n = session['n']
    msg = session['c']
    data = data + [(msg, n)]
x, n = chinese_remainder_theorem(data)
e = session['e']
x=int(x)
print(x)
```

输出的x为m^e,然后对x开10次方，可以用http://factordb.com/ 分解x,得到m后转为字符串，得到flag{wo0_th3_tr4in_i5_leav1ng_g3t_on_it}