### Skynet Is (Almost) Taking Over

Skynet is using a very small list of primes for RSA style encryption purposes. In fact their list is only the size of the smallest odd prime. One of the robots sent a message to three other robots. These are futuristic robots with the ability to use quantum computing and so they don't mind prime factoring huge numbers.You can't do that though. Find out what message the robot sent to his friends. Flag is in flag{} format. [https://mega.nz/#!7WZg2I5I!UiyBukv8\_IjartojnY86nhN5jsQFKE4tPCEF1lPqsQ8](https://mega.nz/" \l "!7WZg2I5I!UiyBukv8_IjartojnY86nhN5jsQFKE4tPCEF1lPqsQ8)

After downloading text file from this link it displays this :

e: 65537

c1: 5024836662627906750454817701922271080214720765897113783786369197810770999608528443597447448508876214100063962982376037712548944474807897847869334582773452689962992522987755069402952836848501053684233233850594080254869

n1: 10603199174122839808738169357706062732533966731323858892743816728206914395320609331466257631096646511986506501272036007668358071304364156150345138983648630874220488837685118753574424686204595981514561343227316297317899

c2: 130884437483098301339042672379318680582507704056215246672305503902799253294397268030727540524911640778691710963573363763216872030631281953772411963153320471648783848323158455504315739311667392161460121273259241311534

n2: 5613358668671613665566510382994441407219432062998832523305840186970780370368271618683122274081615792349154210168307159475914213081021759597948038689876676892007399580995868266543309872185843728429426430822156211839073

c3: 40136988332296795741662524458025734893351353026652568277369126873536130787573840288544348201399567767278683800132245661707440297299339161485942455489387697524794283615358478900857853907316854396647838513117062760230880

n3: 43197226819995414250880489055413585390503681019180594772781599842207471693041753129885439403306011423063922105541557658194092177558145184151460920732675652134876335722840331008185551706229533179802997366680787866083523

After this I factorize n1 through <http://factordb.com/> and got p1 and q1. Then I wrote a python script to decrypt RSA. The script is :

from Crypto.Util.number import inverse

e=65537

n=10603199174122839808738169357706062732533966731323858892743816728206914395320609331466257631096646511986506501272036007668358071304364156150345138983648630874220488837685118753574424686204595981514561343227316297317899

c=5024836662627906750454817701922271080214720765897113783786369197810770999608528443597447448508876214100063962982376037712548944474807897847869334582773452689962992522987755069402952836848501053684233233850594080254869

p=1173821128899717744763168991586024137475923012574062580049287532012184965219319828285650431646942194944437493

q=9033062119150775356115605417902072538098631081058159551678022048966520848600866260935959311606867286026034943

phi=(p-1)\*(q-1)

d=inverse(e, phi)

m=pow(c, d, n)

print(hex(m)[2:])

After running this script a hex string was obtained : 666c61677b77696c6c5f68655f62655f6261636b7d

After decoding it to ascii the flag was retrieved : flag{will\_he\_be\_back}