Casper Kristiansson, 2022-12-02

RSA Challenge

Answer: *“comp3632{I_guess_you_learned_some_number-theory}\n”*

I started off by understanding what exact information that we were provided and how the encryption could be re-engineered.
Because we were given the value of CRY we can use a bit of math to find both p and q. Below is a explanation how i found both the p and q value.

To make it easier (Protected variables): Cry=a, N=b
Because we both have the value Cry and N we can derive these values together to solve both p and q

In[ ]:=
```
a = (p + 520) * (q + 520)
```
Out[ ]=

$(520 + p)\ (520 + q)$

In[ ]:=
```
Expand[a]
```
Out[ ]=

$270400 + 520\,p + 520\,q + p\,q$

In[ ]:=
```
b = p * q
```
Out[ ]=

$p\,q$

In[ ]:=
```
a = 270400 + 520 p + 520 q + b
```
Out[ ]=

$270400 + 520\,p + 520\,q + p\,q$

We can then solve for q

```
q = (a - 520 p - b - 270400) / 520
```

Because we know that p*q=N

In[ ]:=
```
b = p ((a - 520 p - b - 270400) / 520)
```

We can then solve for p. Didn’t for some reason get it to work in mathematica so used symbolab. Direct link to calculation (https://www.symbolab.com/solver/solve-for-equation-calculator/-solve%20for%20p%2C%20b%20%3D%20p%5Cleft(%5Cleft(a-520p-b-270400%5Cright)%2F520%5Cright)?or=input)

In[ ]:=

$$p = -\left(\frac{-a + b + 270400 + \sqrt{(-540800\,a + (a - b)^2 - 540800\,b + 73116160000)}}{1040}\right)$$

Out[ ]=

$$\text{Hold}\left[p = -\frac{-a + b + 270400 + \sqrt{-540800\,a + (a - b)^2 - 540800\,b + 73116160000}}{1040}\right]$$

We can then extract the real values of N and Cry

In[ ]:=

```
a =
 26 609 708 421 376 677 628 454 402 900 087 009 846 291 167 287 676 911 113 310 671 001 067 916 215
  975 654 619 357 943 078 675 057 781 284 419 971 876 364 188 201 285 756 254 849 493 795 101 184 689
  472 972 451 252 559 267 516 902 582 277 554 505 702 670 110 528 791 300 961 267 369 272 080 284 734
  306 320 521 513 748 467 464 633 545 459 859 474 195 548 892 296 577 923 424 451 509 458 569 436 363
  709 731 572 253 846 238 252 647 161 985 685 432 295 738 082 766 877 396 752 019 943 012 580 636 589
  164 644 125 010 073 946 413 108 951 305 564 059 881 537 794 476 457 602 047 138 719 485 228 161 010
  739 405 064 157 783 241 778 448 944 470 473 298 163 156 034 126 054 406 807 297 456 937 129 548 816
  176 179 704 045 207 131 224 909 988 357 244 665 869 859 061 263 890 702 529 905 040 557 579 134 990
  132 844 969 289 396 259
b =
 26 609 708 421 376 677 628 454 402 900 087 009 846 291 167 287 676 911 113 310 671 001 067 916 215
  975 654 619 357 943 078 675 057 781 284 419 971 876 364 188 201 285 756 254 849 493 795 101 184 689
  472 972 451 252 559 267 516 902 582 277 554 505 702 670 110 528 791 300 961 267 369 272 080 284 734
  306 320 521 513 748 467 464 633 545 459 859 474 195 548 892 296 577 923 424 451 509 458 569 436 363
  709 731 402 197 392 186 162 426 572 460 924 170 144 815 459 280 292 038 798 573 517 240 473 723 212
  917 475 994 555 278 140 089 160 884 080 770 934 882 248 855 992 019 482 512 867 322 735 936 930 918
  031 567 624 003 424 284 507 526 700 957 286 437 082 738 893 899 468 444 943 650 565 398 213 516 262
  653 534 101 927 337 725 614 414 267 105 976 588 592 783 298 584 640 344 155 571 836 662 897 588 729
  868 409 203 459 117 059
```

In[ ]:=

$$\text{solve}\left(p = -\left(\frac{-a + b + 270\,400 + \sqrt{\left(-540\,800\,a + (a-b)^2 - 540\,800\,b + 73\,116\,160\,000\right)}}{1040}\right)\right)$$

Out[ ]=

```
solve[
 152 214 699 019 836 494 903 547 377 802 069 891 835 160 125 675 054 664 673 130 799 748 364 468 558
  321 404 960 301 825 513 048 599 430 160 487 214 741 427 637 605 602 436 739 682 636 007 962 183 727
  685 419 830 128 035 018 386 108 980 306 504 647 311 849 508 836 638 970 390 702 126 767 981 969 659
  368 705 567 782 992 886 242 214 861 025 214 489 794 145 558 725 639 804 851 710 370 913 203 383 073
  294 650 749]
```

From that we found a solution for p

In[ ]:=

```
ps =
 152 214 699 019 836 494 903 547 377 802 069 891 835 160 125 675 054 664 673 130 799 748 364 468 558
  321 404 960 301 825 513 048 599 430 160 487 214 741 427 637 605 602 436 739 682 636 007 962 183 727
  685 419 830 128 035 018 386 108 980 306 504 647 311 849 508 836 638 970 390 702 126 767 981 969 659
  368 705 567 782 992 886 242 214 861 025 214 489 794 145 558 725 639 804 851 710 370 913 203 383 073
  294 650 749
```

Out[ ]=

```
152 214 699 019 836 494 903 547 377 802 069 891 835 160 125 675 054 664 673 130 799 748 364 468 558
 321 404 960 301 825 513 048 599 430 160 487 214 741 427 637 605 602 436 739 682 636 007 962 183 727
 685 419 830 128 035 018 386 108 980 306 504 647 311 849 508 836 638 970 390 702 126 767 981 969 659
 368 705 567 782 992 886 242 214 861 025 214 489 794 145 558 725 639 804 851 710 370 913 203 383 073
 294 650 749
```

We can then just use the expression N=p*q to find q

In[ ]:=
```
solve[q = b / ps]
```
Out[ ]=
```
solve[
 174 816 943 388 029 313 922 461 778 471 297 267 056 160 230 782 548 116 914 892 482 777 359 083 688 
  083 315 800 864 182 079 388 371 325 849 126 802 447 965 512 624 271 162 100 219 847 126 831 485 190 
  468 723 167 866 716 755 159 108 686 734 034 616 419 355 464 166 944 965 939 180 404 063 851 727 736 
  019 982 642 612 411 913 221 729 885 277 241 425 134 083 768 886 618 528 232 702 478 214 220 202 380 
  101 176 191]
```

In[ ]:=
```
qs =
 174 816 943 388 029 313 922 461 778 471 297 267 056 160 230 782 548 116 914 892 482 777 359 083 688 
  083 315 800 864 182 079 388 371 325 849 126 802 447 965 512 624 271 162 100 219 847 126 831 485 190 
  468 723 167 866 716 755 159 108 686 734 034 616 419 355 464 166 944 965 939 180 404 063 851 727 736 
  019 982 642 612 411 913 221 729 885 277 241 425 134 083 768 886 618 528 232 702 478 214 220 202 380 
  101 176 191
```
Out[ ]=
```
174 816 943 388 029 313 922 461 778 471 297 267 056 160 230 782 548 116 914 892 482 777 359 083 688 
 083 315 800 864 182 079 388 371 325 849 126 802 447 965 512 624 271 162 100 219 847 126 831 485 190 
 468 723 167 866 716 755 159 108 686 734 034 616 419 355 464 166 944 965 939 180 404 063 851 727 736 
 019 982 642 612 411 913 221 729 885 277 241 425 134 083 768 886 618 528 232 702 478 214 220 202 380 
 101 176 191
```

We can then check if the p and q values that we find is correct, meaning that they fulfils both the N and Cry value

In[ ]:=
```
qs * ps === b
```
Out[ ]=
```
True
```

In[ ]:=
```
a === (520 + qs) (520 + ps)
```
Out[ ]=
```
True
```

Next step is then just to decrypt the message. This can be done using the libnum library and performing inverse mod operation to find the d value and then perform power mod. As you can see we calculate two different d values. This is because we encrypt using CRY. It can be calculated by performing the inverse mod of E and (p+520-1)(q+520-1). Because the encryption method uses a random variable to encrypt the message we just perform inverse root on all numbers from 1 to 30. This actually gives us two different solutions.

One:

“\x0f\x00qhK\xad\xd9P\xc5k6u\x1e\x1du:\xae\xccC\x1c}]\x9e\x08\x10\xafX\xa0\xf8\xc2\xb0/z\xe8\x02\x8a\xc6\xed\xde\xcf7A\xbdT\xbfv0Oz{\x89\xfd\xbdMB\xaf|\x9a\xdd\xfc(\xfb(\xaf*\x0eg\xb1\xa8\x94v\xac\x9f\xc9\x1eG\x99l\xa5\xc2#\xdf\xdft*\xe6\x08\x01\x0b\xf0\xdd\x91\xf1W\xf0\xa9$\x02\x95!\xd1)J\xa7\xcfC4\xf7\xae\xa9\x99n\xc1\x05\xa3’\x1d\xe6\xfa4\x95N{7\x80\x17O\xde\xfd_g&\x87|N\x87\xb4\x9d\xb0\x0f9\x0e\xa0\x9el\x7f\xe8”

Two:

“comp3632{I_guess_you_learned_some_number-theory}\n”

In our case we can clearly see that the second option is the correct decrypted message.

```
def example_decryption(c):
    p, q = calculate_pq()
    d = libnum.invmod(E, (p-1)*(q-1))
    d1 = libnum.invmod(E, (p+520-1)*(q+520-1))


    c = gmpy2.powmod(c, d1, CRY)
    c = gmpy2.powmod(c, d, N)

    for i in range(1, 30):
        m, exists = gmpy2.iroot(c, i)
        if exists:
            print('Decrypted Message:', libnum.n2s(int(m)))
```