Molecon Teaser: Moo

Remark 1 The two primes are generated in the following way:

$$p = 2a + 1$$

$$q = 2b + 1$$

with a,b values obtained by the multiplication of 4 primes:

$$a = k1^{x_1} \times k2^{x_2} \times k3^{x_3} \times k4^{x_4}$$

$$b = l1^{x_5} \times l2^{x_6} \times l3^{x_7} \times l4^{x_8}$$

and k_i, l_i primes for every i.

Remark 2 The tool answers with the LCM between the exponents e_1 and e_2 such that, chosen an input g:

$$g^{e_1} = 1 \mod (p)$$

$$g^{e_2} = 1 \mod (q)$$

Moreover, the exponents are chosen from a list qs which contains the multiplication between the values $2, k1, k2, k3, k4$ in one case and $2, l1, l2, l3, l4$ in the other one.

Remark 3 Given a prime p we know that: $g^{\phi(p)} = 1 \mod (p)$ which means $g^{p-1} = 1 \mod (p)$

Remark 4 The server choses $e_1 = 2a$ and $e_2 = 2b$, or $e_1 = a$ and $e_2 = b$. This means that when we compute the LCM, if a and b has not common primes in their factorization, the server will return either $2ab$ or ab.

Solution 1

In the first solution we will not find p and q, meaning we will not perform the factorization. What is $\phi(N) = (p-1)(q-1)$ which means $\phi(N) = (2a)(2b) = 4ab$. Well, the server returns either $2ab$ or ab. Then we can easily take the bigger value we obtained from the server, double it and ... we have $\phi(N)$. From that point we know $e = 65537$ and $\phi(N)$, then we can compute $d = e^{-1} \mod (\phi(N))$ and decrypt the ciphertext. Use long_to_bytes to obtain the flag.

Solution 2

In the second solution we will try to factorize N in order to obtain the two primes p and q. We know $N = pq = (2a+1)(2b+1) = 4ab + 2a + 2b + 1$. As in the previous solution, we can find $\phi(N) = 4ab$. Then we have that:

$$N = 4ab + 2a + 2b + 1$$

$$2a + 2b = N - 4ab - 1$$

$$a + b = \frac{N - 4ab - 1}{2}$$

. We know ab which is $\phi(N)/4$. Put the above formulas ($ab = v$ and $a + b = v_2$) with v and v_2 known from before computations.

$$b = a - v_2$$

which means by substitution that

$$a(a - v_2) = v$$

$$a^2 - av_2 - v = 0$$

. Use **sage** to solve the equation:

```
a = var('a')
assume(a, 'integer')
eq = (a*a - a*v2 - v == 0)
sol = solve(eq)
```

The code will answer with two values for a. One is a, the other is b. Check if $2a + 1$ is prime and $2b + 1$ is prime. If **true** you have successfully compute the two primes p and q. Then, you can compute the flag as in the previous case.