Dividing Secrets.

The code running on the server provides you with a prime p of length 512 bits, g, a random value within 0... p exclusive.

The flag secret is encoded as x, a ~~secr~~ value we presume must be $< p$.

Given: p, g, $enc = g^x \bmod p$.

Now, ~~given~~ we can query the server by asking a value h, and getting back:

$$g^{x//h} \bmod p, \text{ where}$$

$$x//h = \cancel{x} \left\lfloor \frac{x}{h} \right\rfloor.$$

First notice that if we send a small h, we get back $g^{x//h}$. We can calculate $(g^{x//h})^h = g^{(x//h)h} = g^{x-(x \bmod h)}$

By iteratively multiplying $g^{x-(x \bmod h)}$ ~~with~~ with g, till we get g^x, we can work out what $x \bmod h$ is.

This process can be repeated 64 times, with 64 primes starting from 499

this is large enough as we will see later

We get a system of modul~~ar~~ ~~co~~relations:

$$x \equiv h_1 \pmod{p_1}$$
$$x \equiv h_2 \pmod{p_2}$$
$$\vdots$$
$$x \equiv h_{64} \pmod{p_{64}}$$

for $p_1 \cdots p_{64}$ ~~primes~~, consecutive primes starting at $p_1 = 499$.

The Chinese Remainder Theorem gives us a unique value for $x \bmod N$, where $N = \prod_{i=1}^{64} p_i$

~~T~~The choice of p_1 gives that N is more the 512 bits long, hence $x \bmod N = x \in \mathbb{Z}$ (no wrap-around).

This gives us x, the flag.