

Ok, we got 4 files. An image, a PEM key and 2 txt files.

I entered the image on aperi solve and... steghide found something: "THISwasTOOiziECB"

Ok, so ECB is a reference to AES ECB, and it has exactly 16 characters so it might be a key.

The screenshot shows a web-based AES decryption tool. The top section is titled "From Base64" and includes a dropdown menu for "Alphabet" with the value "A-Za-z0-9+/" selected. Below this are two checkboxes: "Remove non-alphabet chars" (checked) and "Strict mode" (unchecked). The bottom section is titled "AES Decrypt" and includes a "Key" input field with the value "THISwasTOO...", a "LATIN1" dropdown, an "IV" input field with the value "IV", and a "HEX" dropdown. At the bottom, there are three buttons: "Mode" (set to "ECB"), "Input" (set to "Raw"), and "Output" (set to "Raw").

In the conversation we find the sum of p and q.

If we open the PEM certificate in a pem decoder, we also find the value of n and e, where n is  $p \cdot q$ .

Now we can find p and q!

```
from sympy import symbols, Eq, solve
S =
247221161699924658810702711996901693624554581933321801618713706784599300902502877
49736220134651803789983780083709278592071250797494670685808152851319806502
P =
152189662152498056403705414170568508936266727797833989074846193193998251501633923
591654577149021291177531465603011130507338728477914317649256149828628303152425417
194350609637963385945455709522253307371596194326796191907374721561001464612958007
488588079796562020731848146769110923462880010550279833363167428057
x = symbols('x')
solutions = solve(Eq(x**2 - S*x + P, 0), x)
p, q = solutions
print("p:", p)
```

```
print("q:", q)
```

Now, before we retrieve the flag we need to turn the secret into a digit by doing base64 -> hex and then hex -> decimal (using cyberchef then rapidtables for example)

Then, all we need to do is plug our values into dcode's RSA calculator and the flag is ours!

**Made with love by: AndreiCat**