The Common Modulus Attack on RSA

November 8, 2023

1 Introduction

In this challenge, we deploy Bezout's identity to read a message sent to *two* different people ($m_1 = m_2 = m$), i.e., encrypted under two different RSA keys that share the same modulus ($n_1 = n_2$).

Thm. (Bezout): Let a, b integers with greatest common divisor d. Then there exist integers x, y s.t. $ax + by = d$.

2 The Solution

Firstly, we deploy the Extended Euclidean Algorithm (EEA, see `crt.py`) to find d as well as the integers x, y from Bezout's identity. a, b in this case are the two RSA keys' public exponents e_1, e_2.

Next, observe the obvious fact that $c_i = m^{e_i}$, for $i = 1, 2$. Taken together, we can compute (assuming $s < 0$ and $t > 0$):

$$\begin{aligned}(c_1^{-1})^{|s|} \cdot c_2^t &\equiv ((m^{e_1})^{-1})^{|s|}(m^{e_2})^t \\ &\equiv m^{se_1+te_2} \\ &\equiv m^d \, mod \, n\end{aligned}$$

Now, a tiny twist comes in. In the most basic form of a common modulus attack, the two RSA keys' public exponents e_1, e_2 are co-prime, which means that $m^d = m^1 = m$. In our case however, $d = 17$. Thus, we aren't done yet! Luckily though, m^{17} is apparently smaller than n, since taking the 17-th root of $m^{17} \, mod \, n$ gives us the flag.