Fermat Factorization

November 14, 2023

1 Introduction

This challenge is pretty bare-bones: we are given n, e, and a ciphertext c. Since the challenge title is *Factorisez-Moi*, our task seems to consist in factoring n, thus breaking RSA and decrypting the ciphertext to obtain the flag.

2 The Solution

Closer inspection of the script used for generating the challenge (`factorisez-moi.py`) reveals that p and q, the prime factors of n, are pretty close - the construction actually ensures that they will nearly be the same in the upper half of bits (p is 1024 bits, and q is obtained by adding a random number of about half that bit size to p). This means that one of the simplest strategies for factoring will work, and that is *Fermat Factorization*!

$m \leftarrow \lceil \sqrt{n} \rceil$
for $i \in N$ **do**
    $\Delta_i = \sqrt{(m+i)^2 - n}$
    if $\Delta_i \in N$ **then**
        return $p \leftarrow m + i - \Delta_i$
    end if
end for

This works because (odd) $n = pq$ can be expressed as a difference of squares: $n = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$. When the algorithm exits, $\Delta_i = \frac{p-q}{2}$, which is precisely the case when $m + i = \frac{p+q}{2}$. The number of steps required for the algorithm to finish is the smaller, the closer p and q are. I used an implementation found on the internet, see code and source in `fermat.py`.