Mind your keys WRITEUP

For this challenge we have a folder `keys` and a folder `msgs` with 20000 files, looking at the files in `keys` we can deduce that we are dealing with RSA cryptosystem, often this high number of public keys means one thing: Common Factor Attack, assuming that exists a positional relationship between key and message, we can now easily build a script to solve the challenge:

```
from pathlib import Path
from Crypto.PublicKey import RSA
from gmpy2 import gcd, invert
from base64 import b64decode
from Crypto.Cipher import PKCS1_OAEP

p = Path("keys")

names = [f.name for f in p.glob("*.pem") if f.name]

names.sort(key=lambda f: int(f[3:-4]))

pkeys = {}
bkeys = {}

for file in p.glob("*.pem"):
    with open(file, "rb") as f:
        key = RSA.import_key(f.read())

    pkeys[file.name] = (key.n, key.e)

while pkeys:
    filename, key = pkeys.popitem()
    n, e = key
    for keyfile, key_ in pkeys.items():
        n_, e_ = key_
        g = gcd(n_, n)
        if g != 1:
            p, q = g, n//g
            p_, q_ = g, n_//g
            phi = (p-1) * (q-1)
```

```
phi_ = (p_-1) * (q_-1)
key1 = RSA.construct((int(n), int(e), int(invert(e, phi)), int(p), i
key2 = RSA.construct((int(n_), int(e), int(invert(e_, phi_)), int(p_
c = open(f"msgs/msgs{filename[3:-4]}.enc", "r").read()
c = b64decode(c)
decryptor = PKCS1_OAEP.new(key1)
m = decryptor.decrypt(c).decode()
print(m) # {FLG:sh4r1ng_s3cr3ts_w34k3ns_th3m}
break
```