

### Problem 1: RSA with a low-entropy prime generator

*Solution:* Here, the main idea is that if the prime generator has a low-entropy, it is very likely that two of the sampled  $N$  will have a common factor. Since the gcd of two numbers can be calculated efficiently, we can easily factorize the number and thus break RSA.

```
1 def attack():
2     L=[] # A list for storing N
3     for i in range(200):
4         (N, e), ct = restart_system()
5         for i in L:
6             p=gcd(i,N)
7             if p!=1 and p!=N and p!=N:
8                 q=N//p
9                 phiN=(p-1)*(q-1)
10                d = inverse(e, phiN)
11                return dec(ct, N, d)
12     L.append(N)
13
```

Listing 1: RSA with a low-entropy prime generator

### Problem 2: Another Attack on RSA Signatures

*Solution:*

### Problem 3: Attack on RSA PKCS Padding: Bleichenbacher's attack

*Solution:* Here we implemented Bleichenbacher's attack as mentioned in [this](#) paper. We also referred to [this](#) video for understanding better.