Notebook

October 21, 2024

1 Problem 1: RSA signature

1.1 Problem

We want to sign the message M using the RSA-signature. As usually, let $N = p \cdot q$ be the RSA-modulus, where p and q are two big primes. Let e be the RSA-public exponent and d be the RSA-secret exponent satisfying that $e \cdot d = 1 \mod (p-1)(q-1)$. The desired signature is given by

$$S \equiv M^d \mod N.$$

Suppose that the attacker knows the value

$$M_p \equiv M^{d_p} \mod p,$$

but he doesn't know the value

$$M_q \equiv M^{d_q} \mod q,$$

where

$$d_p \equiv d \mod (p-1), d_q \equiv d \mod (q-1).$$

If the attacker knows the modulus N (but not p and q), the public exponent e (but not d), and the original message M, what secret signature parameters can he calculate? Justify the answer.

1.2 Solution

We have

$$d_p \equiv d \mod (p-1) \Rightarrow e.dp \equiv 1 \mod (p-1).$$

So that

$$M_p^e = M^{d_p * e} \equiv M \mod p \Rightarrow (M^{d_p * e} - M)\%p = 0$$
$$\Rightarrow GCD(M^{d_p * e} - M, n) = p$$

from p we can calculate $q = n/p$ and all secret signature parameters.

1.3 Example

[1]:
```
!pip install pycryptodome
```

```
Requirement already satisfied: pycryptodome in
/home/aothuatgiadp/miniforge3/envs/sage/lib/python3.9/site-packages (3.20.0)
```

[1]:
```
from Crypto.Util.number import bytes_to_long, GCD, getPrime, inverse

p, q = getPrime(1024), getPrime(1024)
n = p * q
e = 0x10001
d = inverse(e, (p - 1) * (q - 1))
dp = d % (p - 1)


M = bytes_to_long(b'nsucrypto')

Mp = pow(M, dp, p)

assert GCD(Mp ** e - M, n) == p
```