Elementary Attacks on RSA

Jean-Sébastien Coron

University of Luxembourg

1 Introduction

Use SageMath and submit a notebook (`.ipynb`) containing your code, checks and short answers. You may reuse keyGen, encrypt and decrypt from TP1. Recall that keyGen(n) returns (N, p, q, e, d), and textbook RSA encryption is $c = m^e \bmod N$.

2 Distinguishing textbook RSA encryptions

The adversary receives (N, e) and chooses two distinct messages m_0, m_1 of the same bit length. The challenger picks a random bit b and returns $c = m_b^e \bmod N$. The adversary must guess b.

1. Implement the following function, using only the public key and the challenge ciphertext:

```
def distinguish(m0, m1, c, Nn, e):
    """Return a guess for the challenge bit b."""
    pass
```

2. Explain why the attack works without the private key. Why is randomized encryption necessary for IND-CPA security? Is randomness alone sufficient for IND-CCA2 security?

3 A chosen-ciphertext attack on textbook RSA

The adversary receives the public key (N, e), a challenge ciphertext c, and access to a decryption oracle. The oracle decrypts a ciphertext using textbook RSA, except that it rejects the exact challenge c. The adversary does not receive d.

1. Given r such that $\gcd(r, N) = 1$, construct $c' = c\,r^e \bmod N$. If $c' \neq c$, ask the oracle to decrypt c'. Show how to recover the plaintext of c from the oracle's answer. Implement:

```
def cca_recover(c, Nn, e, oracle):
    """Recover a suitable challenge using the oracle, not d."""
    pass
```

   Use inverse_mod to invert r modulo N. Choose r until it is invertible and $c' \neq c$.
2. Test your code. Keep d inside the oracle and make the oracle reject the exact challenge. Check that the recovered message equals m and encrypting it gives c.

4 OAEP and PSS

1. From the description of OAEP in the course, provide the pseudo-code of OAEP encryption and decryption.
2. From the description of PSS in the course, provide the pseudo-code of PSS signature and verification.