Basic number theory for cryptography

Jean-Sébastien Coron

Université du Luxembourg

The following exercises can be implemented in C, in Python, or using the Sage library, available at http://www.sagemath.org/. Please provide a single file. Each function should be properly tested in the file.

1 Euclid's Algorithm

Implement a function taking as input 2 integers and outputting their gcd, using Euclid's algorithm.

```
>>> gcd(12,15)
3
```

2 Multiplicative inverse

Write a function taking as input deux integers a and n, and outputting the multiplicative inverse of a modulo n if it exists, using Euclid's extended algorithm.

```
>>> modinverse(5,7)
3
```

3 Chinese Remainder

Write a function taking as input a_1, n_1, a_2, n_2 with $\gcd(n_1, n_2) = 1$, and returning z such that $z \equiv a_1 \pmod{n_1}$ and $z \equiv a_2 \pmod{n_2}$.

```
>>> crt(4,5,3,7)
24
```

Find a formula to generalize the CRT to more than two moduli. Write a function taking as input two lists $[a_1, \ldots, a_k]$ and $[n_1, \ldots, n_k]$ and returning z such that $z \equiv a_i \pmod{n_i}$ for all $1 \leq i \leq k$.

```
>>> crtlist([1,2,3],[5,7,11])
366
```

4 Jacobi symbol

Write a function computing the Jacobi symbol:

```
>>> jacobi(37,47)
1
```

5 Square roots and quadratic equations

Write a function computing square roots modulo a prime $p \equiv 3 \pmod 4$.

```
>>> sqroot(7,19)
[8,11]
```

Write a function finding the roots of a quadratic equation $ax^2 + bx + c = 0 \pmod p$ for $p \equiv 3 \pmod 4$.

```
>>> solvequad(2,4,8,19)
[3,14]
```