Mathematical Cryptology, 2019

Dept. of EIT
Lund University

WEEK 1,

Reading instructions (Smart's book "Cryptography Made Simple"):

- Appendix A (repetition)
- Chapter 1 (partly repetition)
- Chapter 2 (partly repetition)
- Chapter 3 (mostly new)

There will be 4 hours of lectures covering the above.

Hand-in exercises:

1) Review exercises: 1.1-1.9, 2.1-2.6, 3.1-3.6

2) Standard exercises: 1.10-1.13, 2.7-2.10, 3.7

3) Programming exercise: 1.14 (maximum 5 hours)

Hand in no later than **February 1, 2019.**

Ch 1: Review exercises:
1.1 What are the axioms of a group, ring and field?
1.2 Why are the nonzero integers not a group under multiplication?
1.3 Why do we say nonzero real numbers when we look at the reals as a group under multiplication?
1.4 Why are the integers not a field?
1.5 Why is $\mathbb{F}_p[x]/(f(x))$ always a ring?
1.6 What is the value of the Euler ϕ function at $N = p$ and $N = pq$, where p and q are primes.
1.7 Define the Legendre and explain how this is used to test for squares modulo a prime p.
1.8 If an integer a modulo a composite number N has Jacobi symbol 1 with respect to N, is it a square modulo N?
1.9 If there are 60 different coloured balls in a bag, how many do I have to take out on average (with replacement) before I obtain a repeated colour.

Ch 1: Standard exercises:
1.10 Show that if n is odd and factors in k distinct prime factors then the number of solutions of $x^2 = 1 \pmod{n}$ is equal to 2^k.
1.11 Show that g is a generator of $\mathbb{F}_p^*$ if and only if $g^{p-1} = 1 \pmod{p}$ and $g^{(p-1)/q} \neq 1 \pmod{p}$ for all prime divisors q of $p - 1$.
1.12 Given a composite integer N and the three integers $a, b, a^{1/3}b^{1/5} \pmod{N}$, show how you can compute $a^{1/3} \pmod{N}$ and $b^{1/5} \pmod{N}$ (with low complexity).
1.13 By hand or calculator, compute the Jacobi symbols

$$\left(\frac{311}{653}\right) \text{ and } \left(\frac{666}{777}\right).$$

Ch 1: Programming exercise:
1.14 Implement an algorithm to compute Legendre symbols which uses the law of quadratic reciprocity.

Ch 2: Review exercises:
2.1 What is the Fermat primality test and what is the main problem associated with it?
2.2 What is the main difference between a primality testing algorithm like Miller-Rabin, and a primality proving algorithm like ECPP?
2.3 What is meant by a smooth number and how can one use smooth numbers in factoring algorithms. 2.4 Describe the $P - 1$ factoring algorithm.
2.5 Why does finding two numbers x and y such that $x^2 = y^2 \pmod{N}$ allow us to factor N with probability at least 1/2?
2.6 What is meant by a sieve and why are they used in modern factoring algorithms?

Ch 2: Standard exercises:
2.7 Show that if a composite number n passes the Fermat test to base a, but fails the Miller-Rabin test for the same base, then we can factor n.
The next three questions discuss Pollard's Rho method to factoring. Define the sequence

$$x_0 = 2, x_{i+1} = x_i^2 + 1 \pmod{N}.$$

2.8 Show that if we find two values of the sequence with $x_i = x_j \pmod{p}$ then we can factor N with high probability by computing $gcd(x_i - x_j, N)$.
2.9 Argue that we must eventually find two indices $i \neq j$ such that $x_i = x_j \pmod{p}$.
2.10 Show how to find two such indices using a small amount of storage.

Ch 3: Review exercises:
3.1 What cryptographic conclusion do we draw from the Pohlig-Hellman algorithm?
3.2 How does the Baby-Step/Giant-Step method work?
3.3 What are the practical problems associated with the Baby-Step/Giant-Step method and how are these overcome when using Pollard's Rho method?
3.4 What cryptographic conclusions can we draw from the existence of the Pollard's Rho method?
3.5 What cryptographic conclusions can we draw from the existence of index-calculus algorithms in finite fields?
3.6 Discuss the statement that elliptic curves offer higher strength per bit than finite fields.

Ch 3: Standard exercises:
3.7 Use a calculator to compute x such that

$$3^x = 5 \pmod{p},$$

where $p - 1 = 2 \cdot 3 \cdot 101 \cdot 103 \cdot 107^2$.