PUBLIC KEY CRYPTOGRAPHY

Lab 2 (Weeks 3-4)

All programs will be written in versions of C or Python with commented code.

Topics: ciphers, congruences, (pseudo)primality.

- Implement one of the following algorithms, which will be assigned to you during the labs:

1. Belaso cipher.

2. Hill cipher (for $m = 2$).

3. Permutation cipher.

4. Algorithm for solving systems of congruences.

5. Algorithm for computing the value of Euler's function for natural numbers. For a given value v and a given bound b, list all natural numbers less than b which have v as the value of Euler's function.

6. The sieve of Eratosthenes algorithm for generating all prime numbers less than a given bound.

7. Algorithm for determining all Carmichael numbers less than a given bound.

8. Algorithm for determining all bases b with respect to which a composite odd number n is pseudoprime. Use the repeated squaring modular exponentiation method.

9. Algorithm for determining all bases b with respect to which a composite odd number n is strong pseudoprime. Use the repeated squaring modular exponentiation method.

10. Algorithm for determining all generators of the cyclic group $(\mathbb{Z}_n, +)$, where $n \geq 2$ is a natural number. A *generator* of $(\mathbb{Z}_n, +)$ is an element $\hat{g} \in \mathbb{Z}_n$ such that for every $\hat{x} \in \mathbb{Z}_n$ there exists $k \in \{0, 1, \ldots, n-1\}$ such that $\hat{x} = k\hat{g}$.

Points

- **1 point** if handed in by Week 5 (odd week groups) or Week 6 (even week groups).
- **0.5 points** if handed in by Week 7 (odd week groups) or Week 8 (even week groups).

Note: *Each student will keep her/his semigroup for the lab throughout the semester! Taking and presenting labs in weeks with a changed parity may only be done in exceptional cases, if the teaching assistant agrees with it and if time allows.*