Cryptography, winter 2006

PROF. DR. WERNER SCHINDLER, DIPL.-INF. DANIEL LOEBENBERGER

9. Exercise sheet (24.01.2007)
Hand in solutions to the homework exercises on Wednesday, February 7th, in the tutorial/the lecture.

Exercise 9.1 (Pollard's $p-1$ method). *We consider Pollard's $p-1$ method from the lecture.*

- *How can one find all primes $\leq B$, where $B \in \mathbb{N}$?*
- *Assume we want to factor $n = 12827$ using Pollard's $p-1$ method. The bound B is choosen as $B = 6, 13, 27$ respectively. Discuss the impact of these choices on the outcome of the algorithm.*
- *Discuss in general the cases where the algorithm fails to find a factor of n.*

Exercise 9.2. *The public key of an RSA cryptosystem is given by $(n, e) = (247, 17)$.*

- *Encrypt the message $m = 101$ using the public key.*
- *Compute the private key.*
- *Decrypt the message $m = 42$ using the private key you computed.*
- *Assume that the exponentiation with the secret exponent d are performed with the CRT. Calculate $d \pmod{p-1}$ and $d \pmod{q-1}$, N_p, N_q (notation as in C.35).*
- *Decrypt the message $m = 42$ using the CRT.*

Exercise 9.3 (Homework: Multiplicativity of RSA). (6 points)

Let m_1,m_2,m_3 be three messages with known signatures $m_1^d \pmod{n}$, $m_2^d \pmod{n}$ and $m_3^d \pmod{n}$. Let $m := m_1 \cdot m_2^2 \cdot m_3 \pmod{n}$. Compute $m^d \pmod{n}$. 6

Remark: Hash functions prevent the aimed construction of meaningful messages m to exploit the multiplicity of the RSA algorithm. Also padding has positive influence since finding such relations is more difficult for larger integers.

Exercise 9.4 (Homework: RSA). (6 points)

6 Let $n = pq \in \mathbb{N}$ with p, q prime be an RSA modulus, $e \in \mathbb{Z}_n^\times$ and $d = e^{-1} \pmod{\varphi(n)}$. Prove:

$$(x^e)^d = x = (x^d)^e \pmod{n}$$

Hint: CRT!

Exercise 9.5 (Homework: Pollard $p - 1$). (8 points)

8 Here you have the choice which task you want to solve:

- Either: Implement Pollard's $p - 1$ algorithm (presented in class) and factor the number $n = 504380101$. Hand in the (commented) source code, the search bound B you used as well as the factors.
- Or: Factor $n = 28959395670380785503 7$ with a computer algebra system of your choice. Use this knowledge to propose an appropriate smoothness bound B for Pollard's $p-1$ algorithm that yields a successful attack. Justify your proposal. Estimate the number of modular squarings and multiplications that are necessary for one run of Pollard's algorithm.