Advanced Cryptography: Algorithmic Cryptanalysis
DANIEL LOEBENBERGER, KONSTANTIN ZIEGLER

10. Exercise sheet
Hand in solutions until Saturday, 25 June 2011, 23:59h.

To estimate the average effort you put into solving the following exercises, please add after each exercise the amount of time you spent for it.

Exercise 10.1 (DL in \(\mathbb{Z}_d, +\)). (7 points)

We consider the DL for the additive group \(\mathbb{Z}_d, +\).

(i) State input and output of the problem in additive notation.
(ii) Write down an efficient algorithm to compute the DL in \(\mathbb{Z}_d\).
(iii) Estimate the run-time of the algorithm for a \(d\) of bit-length \(n\).

Exercise 10.2 (baby-step giant-step for DL). (10 points)

(i) Consider the cyclic group \(G = \mathbb{Z}_{25}^\times\) with generator \(g = 2\) and compute the discrete logarithm of \(x = 17\) using the baby-step giant-step algorithm from the lecture. Document your steps and set up a table with the values computed for \(xg^k\) and \(g^{km}\).

(ii) To compute the runtime of the algorithm in the general case, \(G\) be a cyclic group with generator \(g\) and of size \(d\). Let \(a = \text{dlog}_g x\) and \(a = im - j\) be the division with remainder of \(a\) by \(m\), where \(0 \leq j < m\). How many baby steps and how many giant steps does the algorithm take exactly and at most?

(iii) Consider the following randomized variation of the algorithm. In every round, a number \(i \in \mathbb{Z}_d = \{0, \ldots, d - 1\}\) and a bit \(b \in \mathbb{Z}_2 = \{0, 1\}\) are independently randomly chosen. If \(b = 0\), we compute \(xg^i\) and store \((i, xg^i)\) in a table \(X\). If \(b = 1\), we compute \(g^i\) and store \((i, g^i)\) in a table \(Y\). We stop, as soon as a value occurs in both tables. How can you compute the discrete logarithm of \(x\) from such a “collision”? How long do you expect this process to take?