The art of cryptography: Lattices and cryptography, summer 2013

PROF. DR. JOACHIM VON ZUR GATHEN, DR. DANIEL LOEBENBERGER

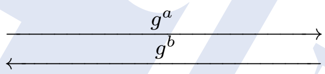
7. Exercise sheet
Hand in solutions until Sunday, 09 June 2013, 23:59h.

Exercise 7.1 (Key exchange). (10+5 points)

As a preliminary step for the Diffie-Hellman key exchange protocol, Alice and Bob have to agree on a cyclic group G and a generator g.

Protocol 7.2. Diffie-Hellman key exchange.

1. Alice chooses $a \in \mathbb{N}_{<\#G}$ and computes g^a.
2. Bob chooses $b \in \mathbb{N}_{<\#G}$ and computes g^b.
3. Alice computes $(g^b)^a = g^{ab}$.
4. Bob computes $(g^a)^b = g^{ab}$.



There are three central topics to be dealt with: correctness, efficiency, and security. The first one is evident from the definition of the protocol. The latter two depend on the choice of the group.

(i) First a note on the efficiency: For the protocol Alice needs to compute g^a. Sketch an efficient algorithm that computes g^a that runs with at most $2\log(a)$ group operations. 3

(ii) Can you do better? Justify. +5

(iii) Name a group G and a generator g for which security may not be ensured. Hint: Extended Euclidean algorithm. 3

(iv) Consider $G = \mathbb{Z}_p^\times$ with p and $\frac{1}{2}(p-1)$ prime, $n := \lfloor \log_2 p \rfloor + 1$. The most efficient known algorithms for computing discrete logarithms in these groups have a running time of $c \cdot \exp((1 + o(1))\sqrt{n \log n})$. During an experiment with prime numbers p as above in the range of 2^{45} the running time for the computation of a discrete logarithm was about 3 seconds. 4

How big should n be so that the key exchange is secure for 100, 1 000 or 10 000 years, respectively? [You are to assume $o(1) = 0$.]

Exercise 7.3 (The security of leading Diffie-Hellman bits). (16 points)

In the lecture we discussed a reduction from computing a solution to the computational Diffie-Hellman problem over $\mathbb{Z}_p^\times$ to the problem of computing μ highest order bits of the solution to the problem.

(i) Compute bounds on μ when the prime p has 512, 1024, 2048 or 4096 bits. 2

(ii) What is in each cases a lower bound on the probability that your reduction worked? Use here the bound $1/2$ on the success probability of the hidden number algorithm given uniformly selected inputs. 2

(iii) Give better bounds. 2

(iv) On the website you find a Diffie-Hellman challenge. It contains several parameter choices as well as an instance of the computational Diffie-Hellman problem. Additionally there is a routine (which should serve as a black box) which implements the leading bit algorithm employed in the reduction from the lecture. Solve the challenge. 10