Wrocław University of Technology, WPPT 6.11.07

CRYPTOGRAPHY AND SECURITY, 2007 Assignments, list # 4

1. (a) Let p be a prime number. Design an iterative algorithm based on the school version of Euclidean Algorithm for computing $x^{-1} \bmod p$ for an input x. Estimate the runtime of this algorithm.

   Is the assumption of primality of p used anywhere?

   (b) Can you use for the same purpose the binary GCD algorithm? If no, then present a reason; if yes, then estimate its runtime.

2. Find a reasonable choice of the parameters for finding discrete logarithm with baby-step giant-step algorithms on a typical PC.

3. Recall the Floyd method applied for Pollard rho algorithm for finding discrete logarithms. Assume that we would like to save time and instead of computing x_i and x_{2i} for $i = 1, 2, \ldots$, we postpone a little bit and start from some j: we compute x_{j+i} and x_{j+2i} for $i = 1, 2 \ldots$.
   How to choose j?

4. One of the proposals to build a hash function is to take prime p such that $p = 2q + 1$, where q is again prime , find and element $g < p$ of order q and then compute hash of (a, b) for $a, b < q$ as the value:

   $$H(a, b) := g^a \beta^b \bmod p \ .$$

   In the last expression, β is an element less than p chosen at random and of order q.

   Why is this method secure?

/-/ Mirosław Kutyłowski