Cryptography

Priv.-Doz. Dr. Adrian Spalka, Konstantin Ziegler

Assignment 7: (P)RNGs and a hardcore bit
Due: Monday, 20 December 2010, 10^{00}

Exercise 7.1. (6 points) Find on the internet hardware-based RNGs. Describe how they work and what they are capable of.

Exercise 7.2. (8 points) Implement the Blum-Blum-PRNG. Under which conditions does it satisfy K3? The NIST provides a statistical test suite. Pick one to analyze the quality of the Blum-Blum-PRNG. Compare it to some randomly chosen digits of π and publicly available statistical data.

Pick two further PRNGs discussed in the lecture and examine their design and statistic quality.

Show how to construct a PRNG from

(i) a symmetric cryptosystem,

(ii) an asymmetric cryptosystem, and

(iii) a hash function.

Which levels of security (K1-K4) can you meet?

Exercise 7.3 (Hardcore predicate for the discrete logarithm)**.** (6 points) Let G be a cyclic group of even order d with a generator g, and let $\omega = g^{d/2}$. Furthermore suppose that an algorithm for computing square roots in G is known. Let Bit_0 be a probabilistic algorithm that, given g^i, computes the least significant bit of i, i.e. $Bit_0(i)$, in expected polynomial time. ($Bit_0(i) = i \mathrm{rem} 2$.)

The square root algorithm takes as input g^{2i} with $0 \leq i < d/2$ and computes either the square root g^i or the square root ωg^i. Let $Oracle$ be a probabilistic expected polynomial time algorithm that decides, which of the two square roots is g^i.

Formulate an algorithm for the discrete logarithm that uses at most polynomially many calls to $Oracle$ and otherwise uses expected polynomial time.

(*Recall:* The algorithm gets as input g^i and should compute the discrete logarithm $dlog_g(g^i) = i$ with $0 \leq i < d$.)

Note: This means that it is already hard to compute the second least significant bit of the discrete logarithm. This is why this bit is called a *hardcore bit*.