Cryptography, winter 2014/2015

PRIV.-DOZ. DR. ADRIAN SPALKA, DR. DANIEL LOEBENBERGER

9. Exercise sheet
Hand in solutions until
Wednesday, 14 January 2015, 23:59:59

Exercise 9.1 (Hardware random number generators). (6 points)

Find on the internet an example of a hardware-based random number generators. Describe detailed how it works and what it is capable of. How is the quality of the generator ensured? Which techniques are used? 6

Exercise 9.2 (Generating random bits in software). (3 points)

Suppose that on some machine, clock time is measured in nanoseconds $1\,\text{ns} = 10^{-9}$ seconds, and that we take the current time, modulo 24 hours, to be a random value. How many random bits would this provide? How many, if we take the time modulo one hour? Modulo one minute? 3

Exercise 9.3 (Playing fair). (4 points)

Suppose you are given a binary symmetric source for which the probability of getting 1, say p, is unknown. If $p \neq 1/2$ we call the source *biased*. How can you use this source to generate an unbiased ($\text{prob}(1) = \text{prob}(0) = 1/2$) binary symmetric source? Give a scheme for which the expected number of bits of the biased source for extraction one unbiased bit is no more than $\frac{1}{p(1-p)}$. Hint: Consider two consecutive bits from the biased source. 4

Exercise 9.4 (Selecting primes uniformly at random). (8 points)

In this exercise we will explore how to select primes uniformly at random from a finite set. For example one can fix a bound $B \in \mathbb{N}$ and select a prime from the interval $[B+1, 2B]$ uniformly at random. There is a famous theorem, called Bertrand's postulate, which says that one can be sure that for any B the number ℓ of primes in $[B+1, 2B]$ is non-zero. Consider now the following algorithm:

Algorithm.
Input: A positive integer $B \in \mathbb{N}$.
Output: A prime p from $[B+1, 2B]$.

1. Repeat
2. $\quad m \leftarrow_R [B+1, 2B]$
3. Until m prime
4. $p \leftarrow m$
5. Return p

Of course the question of deciding whether m is prime or not needs some further attention. For the beginning, let us assume that we are able to decide that problem in a non-probabilistic fashion efficiently. We will come back to that problem later.

3 (i) Analyze the runtime of the above algorithm, i.e. give an estimate on the expected number of loops of the procedure. Hint: You may use here that $\ell \geq \frac{B}{2 \ln B}$ whenever $B \geq 6$.

5 (ii) Show that the above algorithm produces every prime from $[B+1, 2B]$ with the same probability. Hint: Consider the event that the algorithm returns a specific prime after k rounds and sum over all k.