The art of cryptography: security, reductions, and group cryptography

Prof. Dr. Joachim von zur Gathen, Konstantin Ziegler

8. Assignment

(Due: Monday, 11 June 2012, 13^{00})

Exercise 8.1. Let N be a Blum integer, $S_N = \{-(N-1)/2, \ldots, (N-1)/2\}$, $S_N^\times = \{x \in S_N : x \text{ is a unit in } \mathbb{Z}_N\}$, and $U^+ = \{x \in S_N^\times : x \geq 1\} \subset S_N^\times$.

(i) (3 points) Show that $S_N^\times$ is a group and that U^+ is not.

(ii) (3 points) Describe an operation on U^+ that makes it into a group. Determine its order.

Exercise 8.2 (Bonus: Extracting randomness from real-world processes)**.** This exercise investigates three processes to extract randomness from real-world processes.

(i) (+3 points) Describe a process to use the flips of a fair coin to generate a 512-bit number. How long does it take? Assume that it falls on one of its two sides, each with probability 1/2. (Thus neglecting the possibilites that it might land on its edge, fall off the table, does not fall down due to some quantum quirk.) Make a reasonable assumption for the time to flip a coin.

(ii) (+2 points) How long do you expect it to take to obtain two random 512-bit *primes*.

(iii) (+3 points) In a popular German lottery 6 numbers are drawn out of 49 possible numbers (without replacement, ignoring the order). How many bits are necessary to enumerate all possible drawings?

(iv) (+1 points) With weekly drawings, how long does it take to generate two 512-bit random primes?

(v) (+2 points) Suppose that on some machine, clock time is measured in nanoseconds (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?

It is a difficult task to measure the "randomness" of a given generator. The two most popular sets of tests are the *Diehard tests* by George Marsaglia (1995) and the *Statistical Test Suite* by NIST (2000).

GPL-licensed implementations of those tests (and others) are available in the *Dieharder* test suite by Robert G. Brown et.al. It is available from the repositories of most popular Linux distributions and also for download from `http://www.phy.duke.edu/~rgb/General/dieharder.php`.

The tests take as input either a single file of bits or the output of an executable generator.

(vi) (+15 points) Write a program that turns lottery drawings into n-bit strings, where n is the number you computed in (iii). Apply your program to a database with actual lottery drawings and test the resulting files of bits for "randomness". How many tests do they pass?

(vii) (+15 points) Write a program that returns a string based on the current CPU time and test the resulting generator for "randomness". How many tests does it pass?