CSci 4554 Assignment 10

Due Wednesday, April 23 in class

Problem 1 (5 points). You are given $N = 33$ and $x_0 = 16$.

1. Show that N is a Blum integer and that $x_0 \in QR_N$ (hint: you can easily guess the square root of x_0).
2. Generate a sequence of 5 pseudo-random bits using Blum-Blum-Shub generator with N, x_0.

Problem 2 (8 points). Consider the discrete logarithm problem for a group generated by g, i.e. a problem of finding x s.t. $h = g^x$ given g and h. Suppose you are given a choice of a parity oracle (PO) or a half-order oracle (HO). The order of a group generated by g in a group of operations $\mod N$ is defined as the number of distinct elements g^i.

- Which of the two oracles would you use for each of the problems below and why?
- Show your computations for the steps **before** the first call to an oracle.
- Show the data sent to the first call of the oracle and explain what would you do, depending on the result.

The discrete logarithm problems are as follows:

1. $g = 2, h = 7, N = 19$, i.e. as an intermediate step you need to find the order of a group generated by $g = 2$ in the group with operations $\mod 19$.
2. $g = 2, h = 8, N = 14$.

Problem 3 (6 points), a programming problem: can be done in collaboration. Test Java hash functions MD5, SHA-1, and SHA-256 on 10000 non-random bit strings (say, representing a counter) to make sure that the results are uniformly distributed: divide the range into 16 equal regions and measure the number of hits in each region. The link to the Java security package is posted on the course page. Note that the output of the three functions has different size.

Problem 4 (10 points), a programming problem (done in two groups) Use SHA-1 and the approach described in section 10.3.2 to create 10 valid pairs of a message and a message digest HMAC with the shared key 01100111. Add two fake pairs. Send the resulting 12 pairs (in any order) to the other group. Then check the twelve pairs sent by the other group to find two non-matching pairs.

Would you consider this message digest to be sufficiently secure to use in real-life applications? Please explain.

Problem 5 (10 points). Compute a digital signature for a message $m = 77$ according to the digital signature algorithm (with the difference that the hash function is defined on numbers, not on bit strings - see below). Then verify the signature. Show all your compuations. The parameters for the DSS algorithm are: $p = 83, q = 41, g = 2, x = 7, H = (x + 3) \mod 41$. Note that x is the private key; you need to compute y - the public key. Choose l at random (different people may choose different values of l).

Problem 6 (10 points) , a programming problem (done in two groups). Implement OAEP with $N = 61*53 = 3233$. The two groups should pick different e and d satisfying the RSA conditions. Use 4 bits for the message, 4 bits for k_0, and 4 bits for k_1. Use the Haval's function as a hash function (see resources page). As in problem 4, one group computes the encryption and the other group decrypts it and verifies it.