Institut for Matematik og Datalogi March 31, 2005
Syddansk Universitet JFB

Cryptology – F05 – Lecture 8

Lecture, March 18

We finished chapter 5 (skipping sections 5.7.3 and 5.9.2).

Lecture, April 1

We will begin on chapter 6 and cover Diffie-Hellman key exchange from some notes (copied from the earlier edition of the textbook)

Lecture, April 8

We will continue with chapter 6, cover the McEliece Cryptosystem (copied from the earlier edition of the textbook), and introduce digital signatures from chapter 7.

Problem session April 6

1. Do problems 5.34, 6.12, 6.20 (work in the multiplicative group modulo 1103), and 6.22 in the textbook.
2. In class we have discussed the discrete logarithm problem modulo a prime, which means that we have discussed them over fields of prime order. There are also finite fields of prime power order, so for any prime p and any exponent $e \geq 1$, there is a field with $q = p^e$ elements, $GF(q)$. The elements of such a field can be represented by polynomials over $GF(p)$ of degree no more than $e-1$. The operations can be performed by working modulo an irreducible polynomial of degree e. For example, $y = x + x^5 + x^7$ is an element of the field $GF(2^{10})$, represented by

$GF(2)[x]/(x^{10}+x^3+1)$. One can calculate a representation for y^2, by squaring y and then computing the result modulo $x^{10}+x^3+1$, so one gets $x^2+2x^6+2x^8+x^{10}+2x^{12}+x^{14} \pmod{x^{10}+x^3+1} = 1+x^2+x^3+x^4+x^7$. In Maple, you can use the `powmod` function to do these calculations.

Try raising y to the powers $e \in \{33, 93, 341, 1023\}$ to see what result you get. What do you get? What does this prove about y?

3. On my computer using Mathematica (last time I tried), raising to the power 1023 directly failed due to lack of memory. What does this say about how Mathematica did the calculations? What can you do to get around this problem when you try these calculations? (Maple has no problems with these calculations.)

4. Why would there be a preference for working in $GF(2^k)$ for some large k, rather than modulo a prime for some very large prime? Hint: think about how arithmetic is performed.

Assignment due Friday, April 29, 10:15 AM

Note that this is part of your exam project, so it must be approved in order for you to take the exam in June, and you may not work with others not in your group. If it is late, it will not be accepted (though it could become the assignment you redo). You may work in groups of two or three.

Write a program which compares the Solovay-Strassen and Miller-Rabin algorithms for primality testing. Your program should contain implementations of both tests and should find random primes of a given length (the number of bits in the length should be input to the program, as should a security parameter telling how many tests to do before concluding that a number is prime). Try several tests with different lengths (including the length 512) to determine which algorithm is faster. Determine the running time of the algorithms on the actual primes found, in addition to the total running time for finding a random prime.

To deal with the long numbers necessary, you may use Java, there is a class in java.math called BigInteger which should be efficient and easy to use. There is documentation available on IMADA's system at `http://www.imada.sdu.dk/Technical/Manpages/jdk1.3/docs/api/`

This cannot be accessed outside of IMADA. You may use most of the standard methods provided there, though not the routines for generating random primes or for modular exponentiation. (You may test your own for efficiency and correctness by comparing your results to the ones given by the method in the package.)

Please turn in your program and some output. You should write a brief report, explaining how your program should be used, and the results of your timing tests. What accounts for the one algorithm being better than the other? What confidence level did you choose for primality checking?

You should send me your program and any extra files via e-mail (they can just be attachments in `pine`). But, in addition to the e-mail, I would like printed copies of everything.

If you prefer another language to Java, please come talk with me by Wednesday, April 6.