Institut for Matematik og Datalogi
Syddansk Universitet

March 25, 2003
JFB

Cryptology – F03 – Note 8

Lecture, March 18

We finished chapter 5 (skipping sections 5.7.3 and 5.9) and covered sections 6.1 and 6.2 (though the latter only briefly) chapter 6. We also covered Diffie-Hellman key exchange from some notes (copied from the earlier edition of the textbook).

Lecture, March 25

We will cover the first four sections of chapter 6.

No lecture, April 1

Lecture, April 8

We will introduce digital signature schemes from chapter 7 and then begin on hash functions from chapter 4.

Assignment due Tuesday, April 15, 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. If it is late, it will not be accepted.

Write a program which chooses a public-private key pair for ElGamal Public-Key Cryptosystem in $\mathbb{Z}_p^*$, encrypts with the chosen key pair and decrypts with the chosen key pair. Your program should choose the keys, encrypt a message using the public key, and decrypt it using the private key. (It is not necessary that your messages come from character input; they can just be numbers if you find that easier.) These should be three separate functions in

your program, so that it would be easy to separate them into three separate programs.

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 at

`http://www.imada.sdu.dk/Technical/Manpages/jdk1.3/docs/api`

You may use most of the standard methods provided there, including the routines for generating random primes, but write your own modular eponentiation method. (You may test your own by comparing your results to the ones given by the method in the package.)

Please turn in your program and some output, including the public and private keys, plus the input and output for the encryption and decryption. You should write a brief report, explaining how your program should be used, and discussing how long your program took to run. What confidence level did you choose for primality checking? (Choose primes which are about 512 bits long.)

Your report should also mention any major functions you used. You should also send me your program and any extra files via e-mail (they can just be attachments in `pine`).

If you do not choose to do this program in Java, please come talk with me by Thursday, March 27.

Problems for Thursday, April 10

1. 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.

2. 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 (a few years ago), 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.

5. Do problems 6.12 and 6.22 in the textbook.