Institut for Matematik og Datalogi
Syddansk Universitet

October 23, 2003
JFB

Introduction to Computer Science
E03 – Lecture 8

Lecture, October 20

Lecture was cancelled due to illness. You were asked to read chapter 8.

Lecture, October 27

We will cover security from section 3.7 in the textbook and begin on cryptology (not following the textbook; there are some notes on cryptography, from PGP, using a link on the course's homepage).

Lecture, November 3

We will cover the first five sections of chapter 11.

Discussion section: week 44 – Terminal Room

Discuss the following problems in groups of two or three. You will be using the program `gpg` to try encryption. Usage information can be obtained by typing `gpg -h | more` (hitting the space bar will get the rest of it; the vertical line says pipe the output through the next program, and `more` shows a page at a time).

1. Create a public and private key using `gpg --gen-key`. You should choose DSA and El Gamal, and size 1024. Go to the directory `.gnupg` using `cd .gnupg`. List what is in the directory using `ls -al`. Try the commands `gpg --list-keys` and `gpg --fingerprint` to list the keys you have, with the fingerprints, which make it easier for you to check that you have the correct key from someone. How would you use fingerprints?

2. You can save your public key in file in a form that can seen on a screen using `gpg --export -a Your Name >filename`. You are "exporting" your key and specifying where the output should go. Then look at it using `more filename`; the `-a` made it possible to see it reasonably on your screen, since it changes it to ASCII.

3. Mail this file to someone else. (Either another group or within your own group.)

4. Try to figure out how to use `gpg` to "import" the public key you got from someone else. Check the fingerprint.

5. Create a little file and encrypt it. You can use `gpg -sea filename`. What does this do?

6. Mail your file to whoever has your public key. Read their file using the command `gpg -d inputfile >outputfile`. Then look at the output file you created.

7. You can also encrypt a file for your own use using a symmetric key system protected by a pass phrase. Try using `gpg --force-mdc -ca filename`. Then try decrypting as with the file you decrypted previously. Why might you want to do this?

8. Although I did not cover the knapsack cryptosystem, the modular arithmetic described in section 11.6 is used in other systems. Find the multiplicative inverse of 5 modulo 77. You could try using `xmaple` and finding out about the function for computing the Extended Euclidean Algorithm by typing `?igcdex`. Does it help?

9. Discuss the questions 2, 6, and 7 on pages 493–494.

Assignment due 8:15, November 4

Late assignments will not be accepted. Working together is not allowed. (You may write this either in English or Danish, but write clearly if you do it by hand.)

1. Find the multiplicative inverse of 18 modulo 35.

2. Find four different square roots of 4 modulo 35 (numbers which multiplied by themselves modulo 35 give 4).

3. Add two of these different square roots which are not negatives of each other modulo 35 (two where adding them together does not give 35). Find the greatest common divisor of this result and 35. Subtract these same two different square roots and find the greatest common divisor of this result and 35. Think about why you get these results.