CPE5021 Semester 1 – 2007

Individual Assignments on Cryptography

(Due date: Friday, 5:00PM – week 7)

Assignment 1: Design and Implementation of RSA

The main requirement of this assignment is "a simple design and implementation of RSA and ECC packages". The assessment is based on your understanding of the algorithms more than the efficiency, correctness and security of the packages. Design and implement a simple package based on the RSA algorithm to provide encrypting/decrypting and digital signature signing and verifying.

1. Your work must include a function (or method) to generate two prime numbers (or believed to be prime numbers), p and q. (see lecture note for the algorithm).

   (There is free code implementing Miller-Rabin test which can be used to find possible prime numbers and you can borrow it provided you understand it).

2. Your work must also include basic functions (or methods) to generate random numbers and to implement Euclid's algorithm and extended Euclid's algorithm (to find **e** and **d** such that **e*d mod [(p-1)(q-1)] = 1 – see** the lecture notes about RSA and **find** Euclid's and extended Euclid's algorithms from books or papers**)**.
   a. The rest of the work is of your choice.
   b. When you have completed the simple implementation, then you can improve your design and implementation as you wish.

There are many free sources on the Web and you can study them. However, you have to write your own code. You can use the math functions in the library such

as div(), but you have to understand all the code including any code you borrow from the library.

Assignment 2: Design and Implementation of ECC

You are required to design and implement a simple ECC package based on Diffie-Hellman **protocol and the underlying field F_p. The package is to provide encrypting/decrypting and digital signature signing and verifying.**

The Diffie-Hellman **protocol is loosely described as follows (refer to the lecture for** The Diffie-Hellman and ECC)**:**

Let's assume that Alice wants to encrypt and send a message to Bob and Bob wants to decrypt the encrypted message.

(Assume that we are working with (Z_p, modulo) and p is a large prime number.)

c. Alice chooses a random integer **x** and sends Bob **$a = g^x \bmod p$**
d. Similarly Bob chooses a random integer **y** and sends Alice **$b = g^y \bmod p$**
e. Alice then computes **k** from **b** that Bob sent: **$k = b^x \bmod p$**
f. Similarly Bob then computes **$k' = a^y \bmod p$**

Both **k** and **k'** are equal to **$g^{xy} \bmod p$**. Alice or Bob can now use either **k** or **k'** as their shared secret key.

To simplify the implementation, you can work on the underlying field Z_p field, where p is either 11, 23, or 37, and $E(Z_p)$ is defined as described in the lecture. You can also choose any hash function which is available as free source.

To represent a message on an EC, you can use free source code or write your own if you wish to do so. If you use any free source code or library function, you have to understand it.

Submission

You have to submit a hard copy of your work with the **standard assignment format** and a soft copy via email or a floppy disk (Please check this with your tutor).

i) You have to list and explain the steps you follow to complete your work that includes the design and implementation considerations; any suggestions for code improvement; any security and efficiency improvements, etc.

ii) You must provide the **`logic in pseudo code`** of your packages
- Guide to the **`detailed pseudo code`** of the solution to the problem:

> The **`pseudo code`** should be detailed enough so that the mapping between the **`pseudo code`** and the C or Java program is trivial.

iii) Your code has to be well documented.

If you want to include an additional structure chart, it is O.K. However, it does not replace the "**`logic in pseudo code`**"

iv) Report limitations of your design and implementation.

Interview

Your tutors will carry out an interview with each of you to asset your work. If you cannot explain your code or cannot rewrite some part of your own code when you are asked to do so, you may get the lowest mark. This is to make sure that the work you submit is your own work.