# Minutes
21/01/2003 2:00 - 3:00pm

- **Attendance: All present**

**Topics discussed:**

a) Encapsulation and abstract classes: very important!

b) Pollard Rho and Quadratic Sieve qualify as *probabilistic* methods, in the sense that a successful outcome is probabilistic (not guaranteed)

c) Decided on the basic design of our implementation and encapsulation methods:

   - Primality checker (Just use the BigInt package but be sure to show that it is reliable and efficient). This is performed by the server.

   - Perform Trial Division on server (by table) for numbers up to 8 digits, then perform primality check again on the factors, if they are primes, then success, we can exit. The prime table will be needed for Pollard Rho as well.
     If there are large composite factors left over, we perform Pollard Rho on these, up to 16 digits. We need to specify a max cycle size (see later). If there are still large composite factors left over or nothing is found when we have recursed up to the max cycle, we terminate and pass numbers onto Quadratic Sieve. There?s no need to distribute Pollard Rho, it is fast enough running on the server.

   - Quadratic Sieve deals with large numbers. This method is distributed.

d) Ways of distributing:
   i. Use different polynomials on different computers
   ii. Can we have access to multi-processor machine? This way we could run program on multiple clients simultaneously
   iii. Discussed the possibility of a java-based screensaver which could be run in the lab and in colleges. What about a

graphical implementation? This is not easily done in Java, but perhaps we could get away with a very tiny and simple C program?

iv. Client seeks server rather than the other way round, since it?s difficult for the server to find out who is available and when.

e) Suggested to use a ?queues of factors? where we have queue 1 for prime factors and queue 2 for factors waiting to be primality checked. Recursively check the head of queue 2, if it?s prime, append to queue 1, if not, factor it. (similar to an ML implementation)

f) Suggested cycle size for Pollard Rho ? we store all primes from 2 up to p where p is less or equal to 16 digits long. The max cycle size is the number of primes in this range.

g) RMI vs UDP: Decided to use RMI. Although it is difficult to get initial contact, once this is done, the transactions between server and client are transparent. Error warnings are also given.

**TO DO before Thursday Meeting:**

Design Requirements formally laid down, to be discussed and finalised during Thursday meeting.
The requirements should be comprehensive, clearly defined and well explained.

**Agenda for Thursday:**

- Finalise design requirements
- Go on to specification