Distributed Systems **Spring-Summer 2016**

Assignment **Simon Razniewski**

Assignment 4: Distributed MD5 Cracker

This assignment counts 10% towards the final grade. MD5 (= Message Digest algorithm 5) is an algorithm that computes for a given string (e.g., a file, a part of a file, or a password) a 128-bit hash value. The goal in this assignment is to write a distributed MD5 cracker. Given a MD5 hash value the software should be able to find an (numeric) String with that hash value by using a simple brute-force approach. The software should be able to take advantage of multiple, networked computers to distribute the compute load.

To obtain the MD5 hashes to be cracked, your system should contact a server (in the evaluation this will be the lab teachers machine) using the following RMI interface:

```java
package server;

import java.rmi.Remote;

public interface ServerCommInterface extends Remote {

        public void register(String teamName, ClientCommInterface cc) throws Exception;

        public void submitSolution(String name, String sol) throws Exception;

}
```

With the method *register*, your system would register to the server, providing your teamname and an object reference that the server can use to give you tasks. With the method *submitSolution*, you can submit solutions to the server.
The ClientCommInterface must implement a single method with which the server can give you tasks:

```java
package server;

import java.rmi.Remote;

public interface ClientCommInterface extends Remote {

        void publishProblem(byte[] hash, int problemsize) throws Exception;

}
```

For simplicity, passwords will be strings of numbers, e.g. “53071” or “1234567”. The parameter problemsize will tell you the maximal integer that the password may represent. Only one machine of your team can connect to the server.

Task

Your task is to implement a distributed MD5 cracker that can run using 2-4 RPIs. Only one of your machines may connect to the server, and you have to take care of distributing the task between the machines. You may choose Java RMI, TCP or UDP sockets for internal communication.
Your project submission must include:

- The source code of the software;
- A short documentation.

Evaluation

The evaluation will consist of two parts, your documentation and the performance of your system in a live competition to be held on 10.5. (Group B) and 11.5. (Group A).
In the competition, the server will pose a number of hashes to all participating teams, and each time the team that cracks the hash first will get a point. For each wrong submission, a point is deducted. Teams achieving the most points in this competition will get the highest grades.