Peer-to-peer networks – (due till March 11, 2009)

Exercise 2.1: Hash-values

1) Why are hash-values used in Napster?

2) You are offered a requested file from an anonymous participant in a content distribution network. The name of the offered file matches your request but the MD5 hash differs. Would you accept? Or would you accept a matching MD5 hash but a different filename? Explain why. (The hash can be verified by you after download and is thus considered to be credible.)

3) Excursus: One ability of hashes is to hide clear texts. Many (operating) systems store passwords as hash values rather than as clear-text. A password entered for login is hashed and compared to previously hashed passwords in a password file. If the hashes are equal, access is granted.

   Advantage: Even if an attacker can get hold of the password file she can not access a system from outside without knowing which password results in a certain hash. Entering the hash-value itself will not be successful because it would be hashed into a completely different value. But recently, trivial methods have become well-known, how the clear-text behind in particular simple password can be found.
   Find a trivial way to get the original password which results in the following MD5 hash: **380E537ACDAEDD487CA1ADB49D020F7E**

Peer-to-peer networks

Exercise 2.2: Fountain Codes (according to example in exercise on March 5)

A sender wants to transmit the following 32 bits

```
10110100 01011011 01010101 10110110
```

in four chunk-packets, each of which contains 8 bits. Both, sender and receiver use the same random number generation which produces the following bit-stream:

```
1110 0101 1001 0110
```

For data transmission, the **Random Linear Fountain Code** from the lecture is used.

Proceeding:

Sender side
- Divide the message into chunks. Combine the chunks bit-wise according to the bit-merging vector which is taken from the output of the random number generator.
- Stop sending further chunks as soon as a sufficient number of XORed chunks with linear independent merging vectors have been sent.

Receiver side
- Collect the incoming chunks until a sufficient number is received. Sufficient means, that their merging vectors are linearly independent. The merging vectors are taken from the output of the random number generator as was done on the sender side.
- After having gathered enough data, calculate the modulo 2 inverse matrix of the matrix formed by the merging vectors.
- XOR the received chunks according to the inverse matrix.