Exercise 2.1

Alice can read and write to the file x, read the file y and execute the file z. Bob can read x, read and write to y and has no access to z.

a) Write access control lists for this situation.

b) Write capability lists for this situation.

c) What is the difference between access control lists and capability lists in terms of revoking all access rights to a specific file and revoking all access rights for a specific person?

Exercise 2.2

Assuming that passwords have length 6 and all the (English) alphanumerical characters, upper and lower case, can be used in their construction. How long will a brute force attack take on average if

a) it takes one tenth of a second to check a password?

b) You use a GPU engine. Using a modern graphics card GPU engine one can implement a rather effective password search engine (see G Tech Case Study). Suppose your search engine can make 500 million passwords test per second.

c) Compare your answers to the case when the passwords have length 8.

Exercise 2.3

Construct the lattice of security labels for the security levels *public*, *con- fidential*, *strictly confidential*, and for the categories `ADMIN`, `LECTURERS`, and `STUDENTS`.

a) Which objects are visible to a subject with security label (confidential,STUDENTS) in a need-to-know policy?

b) How many labels can be constructed from n security levels and m categories? For illustration, consider the values n=16 and m=64.

Exercise 2.4

You are given a security policy that uses the lattice of compartments as security labels. Access is granted only when the subject's label is a subset of the objects label.

a) With the categories `ADMIN`, `LECTURERS` and `STUDENTS`, which objects can be accesses by a subject with label `STUDENTS`?

b) Why is a subject with label `ADMIN`, `STUDENTS` more constrained than a subject with label `STUDENTS`?

c) Interpret the roles of the labels $\emptyset$ and `ADMIN`,`LECTURERS`, `STUDENTS` in this policy.

Note the difference between this exercise and exercise 2.3.

Exercise 2.5

a) In Unix, why are shadow password files used and how do they differ from the normal password files?

b) What is a salt and why is it used when a password is encrypted?

Exercise 2.6

Consider a hash function h which is known to have good cryptographic properties and whose output word length (that is the size of the hash) is 64 bits.

Compute the length k of two (equally long) lists with random character strings/words of length 256 so that the probability that these two lists have at least a word in common whose hashes are equal is 0.5. Check by using the Birthday Paradox if the number of collisions inside the same list (that is words belonging to the same list that have identical hash values) can be ignored (that is the number of collisions in the list is small compared to k). State your answer as a power of 2.

Hint: Show first that for each value we compute in the second table, we have a probability $(N-k)/N$ that it does not coincide with any value in the first table. So, the probability $P(k)$ that NO value in the second table coincides with any value in the first table is $[(N-k)/N]^k$, where $N = 2^{64}$ is the number of possible different hash values. You will have use for the following approximation $\ln(1-k/N) \approx -k/N$ if $k << N$.

You may use the following notation when answering: K(ilo) words $= 2^{10}$ words, M(ega) words $= 2^{20}$ words, G(iga) words $= 2^{30}$ words.

Exercise 2.7

Explain why a random salt protects against the time-memory tradeoff (or rainbow) attack.

Exercise 2.8

Explain the access permissions given to the program `run` when it is exe- cuted. To what extent are the permissions useful?

```
-rwsr--r-- 1 root admins 504 2010-02-01 05:43 run
```

Exercise 2.9

In a Unix system what effect does the command `umask 027` has on newly created files and programs?

Exercise 2.10

Assume that the file `Bill.txt` is owned by user Alice and the group Stu- dents. The file permissions in Unix give the owner read access and the group write access (420). In Windows there is one ACE giving Alice read access and one ACE giving the group Students write access. How is this seemingly similar situation handled in Unix and Windows respectively.

Exercise 2.11

When a user starts a process, the permissions of the user have to be transferred to the process which will be acting on behalf of the user. How is this implemented in Unix/Linux and Windows respectively?
