CAA 22-23

Mini-project

01.12.2022

- Submit **your code** and **your report** on Cyberlearn.
- The **quality** of the cryptographic implementation will be graded.
- The grade 4 is obtained by a correct (and clear) modelling of the cryptography in a report.
- The remaining 2 points are obtained from a good implementation.
- The programming language is free (preferably among Rust, C/C++, Java, Python/Sage). If you would like to use another language, please ask first.
- Do not hesitate to ask questions.

1 An Encrypted Vault

The goal of this laboratory is to implement an online vault storing very sensitive files (e.g. root certificates) for companies. Here are the requirements:

- We assume a **passive adversary**. We also assume that the communication channel between the client and the server is not confidential.
- First, the client needs to authenticate. Since the files are really sensitive, two people out of n need to gather to access a file. The process is the following:
  1. The company sends its `company name` to the server.
  2. Then, two members of the company (out of n) enter their credentials (`username + password`) to unlock the vault.
- The server is allowed to know the usernames. Beware that the same username might be used by two different companies.
- Once the client is authenticated, he receives an **encrypted** list of filenames. He can then select which file he wants.
- The selected file is sent by the server (still encrypted).
- The server should **never** see the documents in clear and should not be able to recover them (assuming "good" passwords).
- The client should not have to enter more than **one password** per member.
- Clients should be able to connect to the vault from any computer and change device as they want.
- If a document's encryption key leaks, one should not be able to decrypt other documents.
- Finally, you need to be able to **revoke** a user. This should **not require** the re-encryption of all files.

2 Deliverable

You have to deliver the following:

- A report describing your cryptographic architecture and explaining your choices (3/5). In particular, provide a scheme describing how the keys are managed.
- Your code (2/5). Note that we do **not** ask you to implement the networking part if you do not want to. You also do not have to use a real database. You can simulate everything with a local file if you prefer.