Cryptographically Enforced Access Control

Consider a Personal Health Record (PHR) system that is used by the following types of parties: patient, doctor, insurance, health club, employer, and hospital.

Design and implement a demonstrator that supports the following functionality:

1. A patient can insert personal health data into his *own* record.
2. A patient can provide his doctor, his insurance, and his employer with read access to (parts of) his health record.
3. A hospital can insert patient health data for any patient that has been treated by *that* hospital.
4. A health club can insert training-related health data to any patient record that is a member of that club.

Your report on the design of the demonstrator should include:

1. A definition of the data model.
2. The description of the access control model of your choice. The access control model should be selected based on how well-suited it is for distributed environments, and especially the environment described by the aforementioned system requirements.
3. The definition of the access policies for each type of parties.

All design choices should be motivated.

The implementation should support the four above-mentioned system requirements, while ensuring that nothing beyond this functionality can be provided to any type of party.

***Note**: You need to use the (cryptographic) techniques taught in the lectures to enforce access control and not to implement classical access control. This is because in the scenario under consideration, the data held in the PHR is private patient information, which should remain confidential. Thus, it is a prerequisite that the data is encrypted in the first place.*

Search in Encrypted Data

Consider a financial consultant that uses a cloud storage service to store the financial data of his clients. The cloud storage server is **honest-but-curious**. To prevent data leakage, the consultant stores all data on the cloud server in encrypted form.

Design and implement a demonstrator that supports the following functionality:

1. The consultant can insert financial data for all of his clients in the storage server.
2. The consultant can search for specific information for any specific client of his in the encrypted data on the server.
3. The client can insert data in his own encrypted record on the storage server.
4. The client can search for specific data in his own record on the storage server.

To achieve the aforementioned functionality, apart from a suitable encryption scheme, you need to deploy a key distribution method, during the system setup.

Your report on the design of the demonstrator should include:

1. A definition of the data model.
2. The description of an encryption scheme and a key distribution method of your choice. The combination of the encryption scheme with the key distribution method should allow the consultant to search in any client's record and the clients to search **only** in their *own* records.
3. The system should be designed in such a way that clients can NOT search in other clients' data, while the consultant can search in all data. This should be demonstrated in your report, by a usage scenario of the implemented system.

All design choices should be motivated.

***Note**: We have assumed that the cloud storage service provider is honest-but-curious. Hence, although we trust it to follow the protocol honestly, we assume that it wishes to learn as much information as possible. Your demonstrator should guarantee that the cloud storage service provider is not able to learn about the actual encrypted data it holds.*

Decentralized private ML

Consider a cybersecurity company providing ML-based Intrusion Detection Systems (IDS) ; i.e., security solutions to detect attacks in large networks.

Currently, each IDS is trained locally in the client's data center using the client's data. To improve the IDS performance, the company would like to train a common model based on all the clients' data. However, due to confidentiality reasons, **the clients do not want to share their traffic data** with anyone (including the IDS provider). Thus, the company would like to use decentralized private ML to train this common model

Design and implement a demonstrator that supports the following functionality:

1. The IDS provider defines the model architecture.
2. The clients can query the common model locally (= have a plaintext access).
3. The clients have access to all the successive versions of the common model.
4. The IDS provider can ask for performance metrics about the common model (e.g., accuracy on each client's data).

Your report on the design of the demonstrator should include:

1. A definition of the data model.
2. The description of the decentralized ML protocol of your choice. The protocol should be selected based on the suitability for ML models used in IDS, and the number of parties involved in the training process..
3. The system should be designed in such a way that clients leak as little information as possible about their data, while obtaining the best model performance possible.. This **privacy-accuracy trade-off** should be thoroughly discussed and motivated.

All design choices should be motivated.

***Note**: We have assumed that the IDS provider can deploy a server to assist and/or coordinate the training. Moreover, we have assumed that this server is honest-but-curious. The company has about 50 IDS deployed everywhere in Europe.*