Cryptographic RBAC Compiler

Third Sprint

12/11/18 - 25/11/18

Layout of the slides by Coman Catalin Andrei

Agenda

- Recap
- Sprint Backlog
- User Stories
- Design
- Implementation
- Testing
- Next Sprint

- Refactoring of the system architecture in **two subsystems**: client and server (class diagram, user stories, test-cases, UML-sequence diagrams, packages, modifiers of variables and methods, ...)
- Design and implementation of Java classes for **communication** between these two parts (through **sockets**)
- ~~Creation of a **Keystore** (server side for the admin, client side for the users) for managing the keys~~
- ~~Add **symmetric keys** to the environment (generation, encrypt, decrypt...)~~

Sprint Backlog

- Creation of a **storage solution** (server side for the admin, client side for the users) for managing the keys -> **<suspended>**
- Implementation of **symmetric keys** in the system (generation, encryption and decryption of keys and files, ...)
- Creation of ER diagram and consequent tables for the database
- Implementation of adding and retrieval of **users, roles and files** (for now file key = its path) through a **database manager** class that exposes such APIs

User Stories

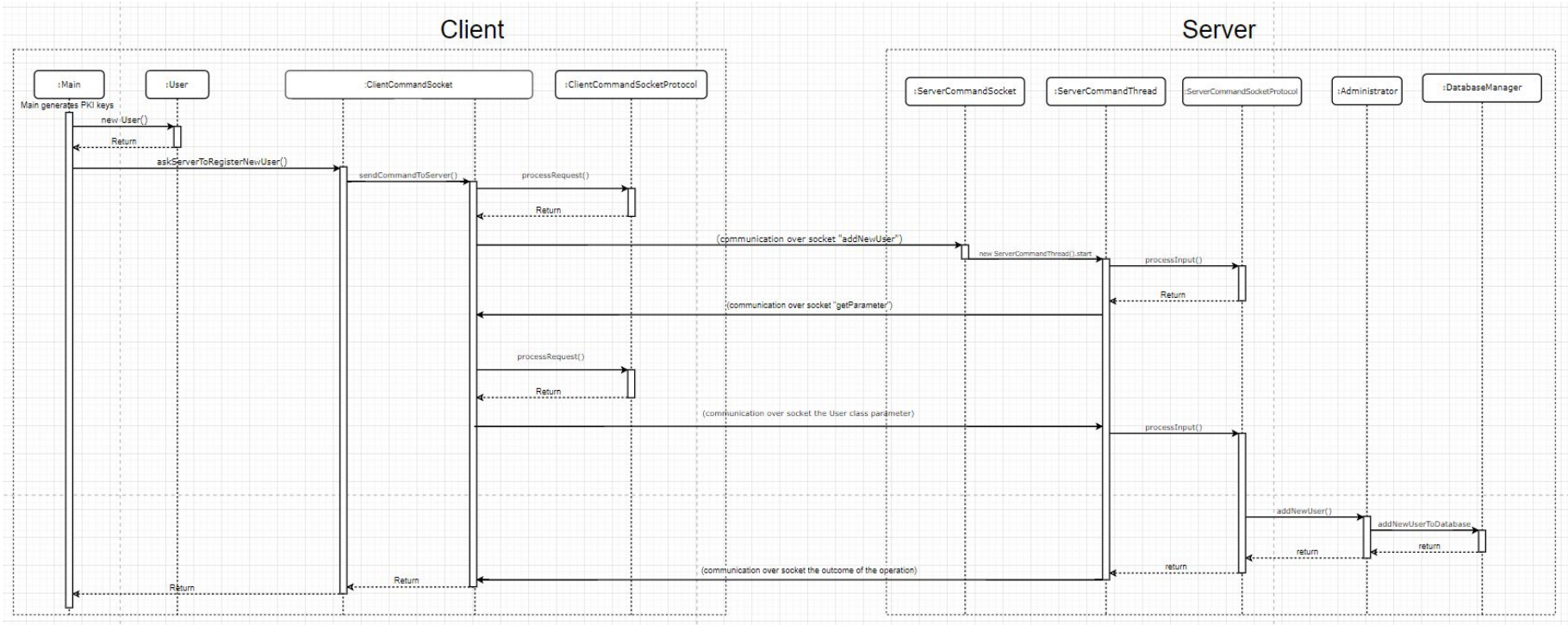
Third Sprint

1. (Refactoring) As a new **User**, I want to generate my PKI keys in order to send them to the server to register me as a new user.

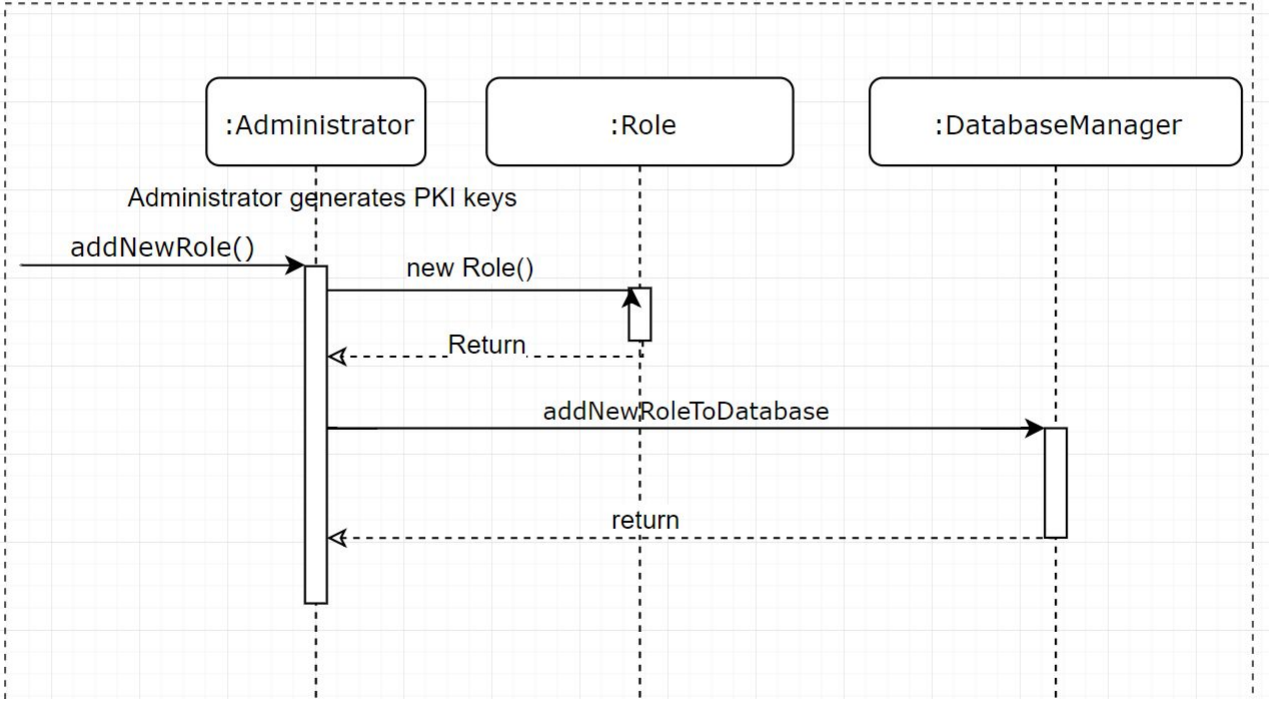
2. (Refactoring) As the **Administrator**, I want to instantiate a new Role in order to create and store its keys.

7. As the **Administrator**, I want to encrypt a stored File with a symmetric key to later decrypt it

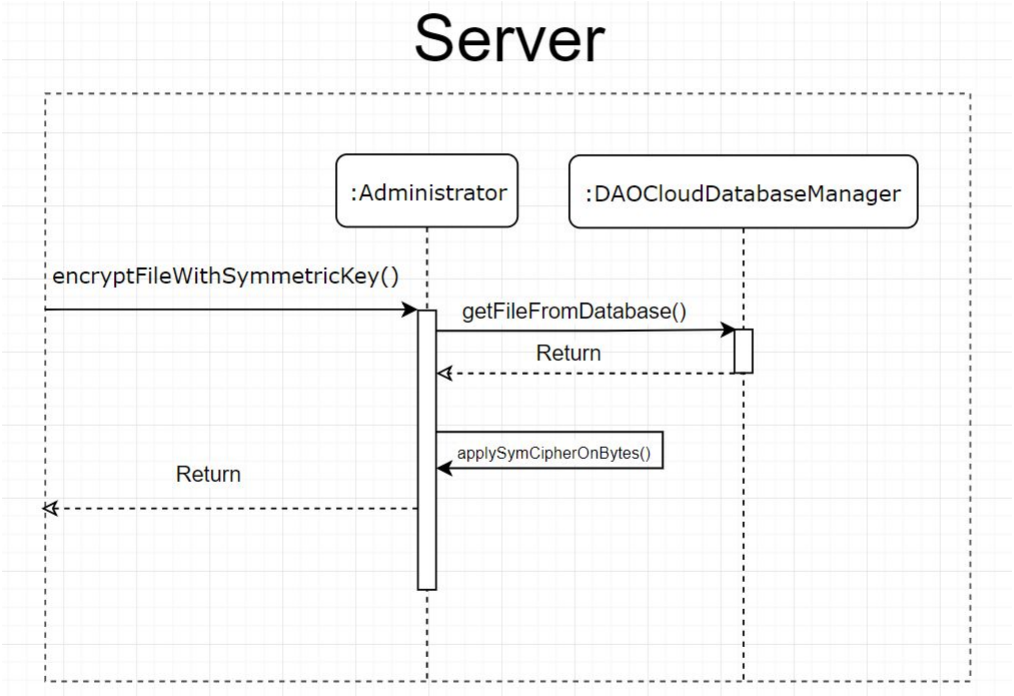
As a new User, I want to generate my PKI keys in order to send them to the server to register me as a new user.

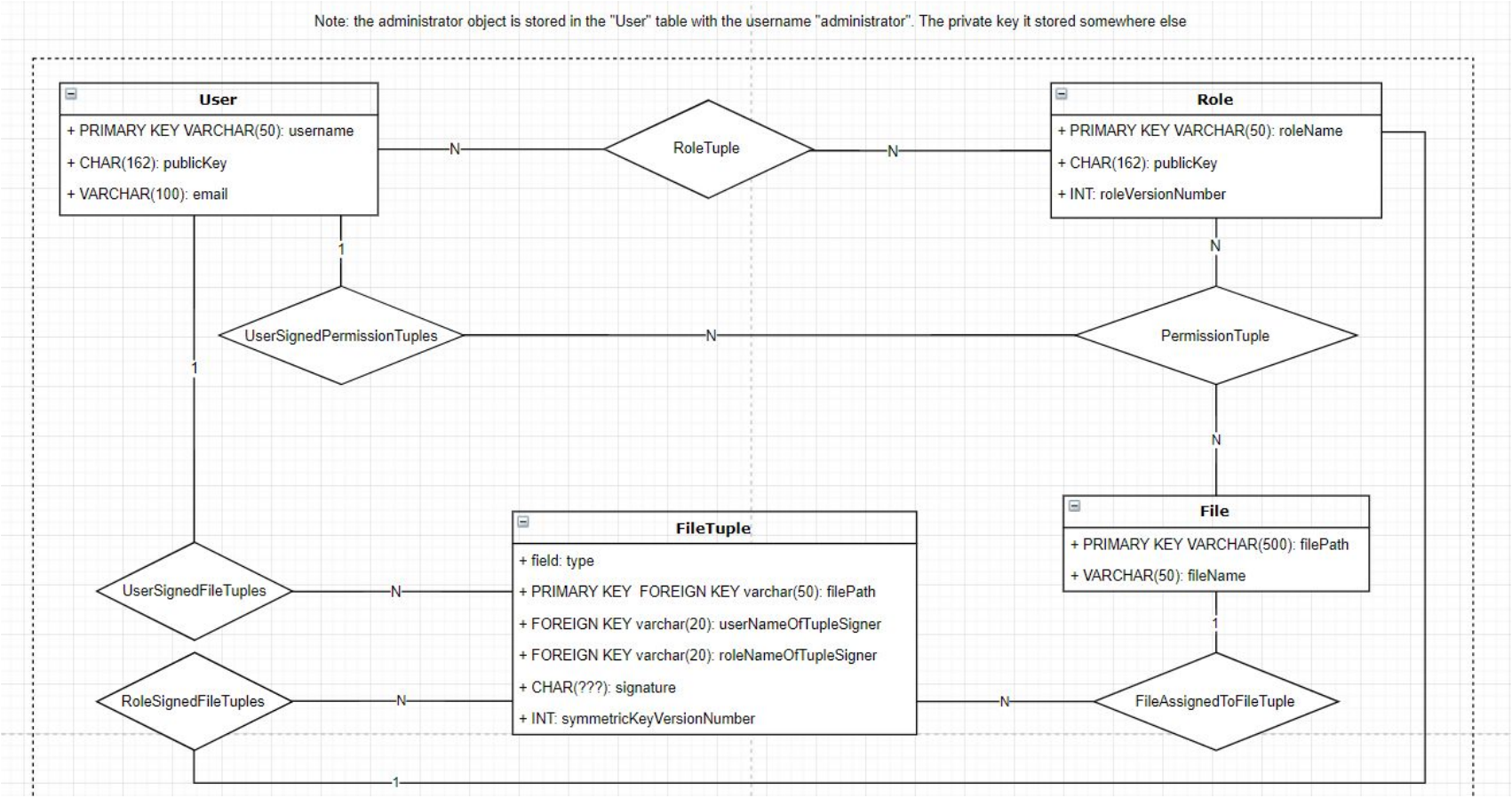


As an **Administrator**, I want to instantiate a new Role in order to create and store its keys.



As the **Administrator**, I want to encrypt a File stored in the cloud with a symmetric key to later decrypt it




Note: the administrator object is stored in the "User" table with the username "administrator". The private key it stored somewhere else
User
+ PRIMARY KEY VARCHAR(50): username
+ CHAR(162): publicKey
+ VARCHAR(100): email
RoleTuple
Role
+ PRIMARY KEY VARCHAR(50): roleName
+ CHAR(162): publicKey
+ INT: roleVersionNumber
UserSignedPermissionTuples
PermissionTuple
FileTuple
+ field: type
+ PRIMARY KEY FOREIGN KEY varchar(50): filePath
+ FOREIGN KEY varchar(20): userNameOfTupleSigner
+ FOREIGN KEY varchar(20): roleNameOfTupleSigner
+ CHAR(???): signature
+ INT: symmetricKeyVersionNumber
File
+ PRIMARY KEY VARCHAR(500): filePath
+ VARCHAR(50): fileName
UserSignedFileTuples
RoleSignedFileTuples
FileAssignedToFileTuple
N
1

Note: ALL tuples in RoleTuple are signed by the Admin

User

+ PRIMARY KEY VARCHAR(50): username
+ CHAR(162): publicKey
+ VARCHAR(100): email

Role

+ PRIMARY KEY VARCHAR(50): roleName
+ CHAR(162): publicKey
+ INT: roleVersionNumber

File

+ PRIMARY KEY VARCHAR(500): filePath
+ VARCHAR(50): fileName

RoleTuple

+ PRIMARY KEY FOREIGN KEY varchar(50): username
+ PRIMARY KEY FOREIGN KEY varchar(50): roleName
+ CHAR(???): encryptedRoleKeys
+ CHAR(???): signature

FileTuple

+ field: type
+ PRIMARY KEY FOREIGN KEY varchar(50): filePath
+ FOREIGN KEY varchar(20): userNameOfTupleSigner
+ FOREIGN KEY varchar(20): roleNameOfTupleSigner
+ CHAR(???): signature
+ INT: symmetricKeyVersionNumber

PermissionTuple

+ PRIMARY KEY FOREIGN KEY varchar(50): roleName
+ PRIMARY KEY FOREIGN KEY varchar(500): filePath
+ FOREIGN KEY varchar(50): usernameOfTupleSigner
+ CHAR(???): signature
+ VARCHAR (2): permission
+ INT: versionNumber
+ CHAR(???): encryptedFileKey

How were functionalities implemented?

- Symmetric Key through native implementation (`SecretKey`)
- Local and cloud storage with DAO pattern

(see code for details)

Testing

Wrote tests for:

- Symmetric key encryption and decryption
- Encryption and decryption with symmetric key (files)
- AddUser and AddRole functionalities

(see code for details)

Next Sprint

- Create the project WIKI (Code on gitlab, graphs, documentation, ...) and share it with Adam
- Definition and implementation of scenario as sequence of operations (AddUser, AddRole, AssignUserToRole, AddFileFromUser, ...)
- Refactoring, Test implementation and TODO resolution

Thanks

Layout of the slides by Coman Catalin Andrei