

**Université Euro Méditerranéenne Fès**

**Euro Mediterranean University of Fez**

**Ecole d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)**

**Projet de Fin de module**

**Filière :** 1ére année cycle d’Ingénieur Cybersécurité

**Semestre** : 6

**Module :** Distributed Applications

**Titre :**

**Gestionnaire de Mots de Passe – RPC & RMI**

Soutenu le 17/06/25.

**Réalisé par :**

- EZZAIMI SALMA

- HADI IMANE

**Encadré par :**

- AMAMOU AHMED

**SOMMAIRE**

1. Introduction
2. Buts éducatifs
3. Structure du projet
4. Fonctionnalités mises en œuvre
5. Éléments de sécurité
6. Interface de l'utilisateur
7. Présentation (extrait de la console et de l'IHM)
8. Problèmes rencontrés
9. Problèmes rencontrés
10. Possibilités d'améliorations
11. Conclusion

1. **INTRODUCTION :**

L'objectif de ce projet est de concevoir une application en Java qui assure la gestion sécurisée des mots de passe, en utilisant deux technologies spécifiques **: RPC (Remote Procedure Call)** **et RMI (Remote Method Invocation)**. Grâce à une interface graphique conviviale conçue avec Swing, l'utilisateur a la possibilité d'enregistrer, de chercher, de supprimer et de répertorier ses mots de passe.

**Définition des technologies employées :**

* **RPC (Remote Procedure Call) :** Le protocole RPC offre la possibilité à un programme d'invoquer une procédure (ou méthode) sur un autre système informatique, comme si elle était exécutée localement. Dans ce projet, l'implémentation du RPC se fait par le biais de sockets entre le client et le serveur.
* **RMI (Remote Method Invocation) :** RMI est une technologie Java qui autorise l'appel de méthodes sur des objets situés à distance. À l'inverse de RPC, RMI fait partie intégrante de Java et offre la possibilité de gérer directement des objets distants, tout en prenant en charge la sérialisation, la communication et les questions de sécurité.

**Description du travail accompli :**

La tâche consistait à développer deux variantes de l'application : l'une utilisant RPC et l'autre se basant sur RMI. Chaque version comprend :

* Une interface utilisateur a été conçue en utilisant Swing.
* Un système de sécurité pour l'authentification qui utilise le hachage SHA-256 pour protéger les mots de passe.
* Fonctionnalités complètes pour la gestion des mots de passe : ajout, suppression, recherche et visualisation en liste.
* Pour RMI, le suivi des sessions en cours se fait grâce à un système de traçage des utilisateurs connectés.
* Pour RPC, un protocole de communication élémentaire qui utilise des chaînes de caractères séparées par des points-virgules pour transmettre les actions et les paramètres. de créer une application Java assurant la gestion sécurisée des mots de passe, en utilisant deux technologies : RPC (Appel de Procédure à Distance) et RMI (Invocation de Méthode à Distance). Grâce à une interface graphique conviviale conçue avec Swing, l'utilisateur a la possibilité d'enregistrer, de chercher, de supprimer et de répertorier ses mots de passe.

1. **Buts éducatifs :**

* Comprendre les notions de communication distante : RPC et RMI
* Implémenter une architecture de type client/serveur en Java.
* Protéger les informations grâce à l'utilisation du **hachage SHA-256**.
* Concevoir une **interface graphique** conviviale.
* Vérifier les entrées de l'utilisateur et traiter les erreurs

**III .** **Structure du projet :**

Le projet comprend deux versions distinctes :

* **Version RPC** : se sert des sockets pour l'échange d'informations entre le client et le serveur.
* **Version RMI** : se sert de Java RMI pour l'appel de méthodes à distance.

Chaque version inclut les éléments suivants :

* **Client** : Swing interface graphique.
* **Serveur** : gestion des opérations et conservation en mémoire.
* **Modèle** : gestion des mots de passe avec logique de hachage (*PasswordStore* ou *PasswordManagerImp*l).

**IV.** **Fonctionnalités mises en œuvre :**

|  |  |
| --- | --- |
| **Fonctionnalité** | **Description** |
| Inscription | Création d’un nouveau compte utilisateur avec mot de passe haché |
| Connexion | Vérification des identifiants |
| Sauvegarde | Ajout d’un mot de passe lié à un service |
| Suppression | Suppression d’un mot de passe pour un service donné |
| Recherche | Recherche du mot de passe associé à un service |
| Liste | Affichage de tous les mots de passe enregistrés |

**V.** **Éléments de sécurité :**

* Avant d'être enregistrés, les mots de passe des utilisateurs sont transformés en **hachage SHA-256**.
* Les mots de passe stockés (associés aux services) ne sont pas encryptés (⚠ à améliorer).
* Validation des données sur le côté client.
* Gestion élémentaire des sessions via loggedInUsers (RMI).

**VI.** **Interface de l'utilisateur :**

* L'interface utilisateur a été réalisée en utilisant Java Swing. Elle inclut :
* Une interface de connexion/enregistrement.
* Une fenêtre principale comprenant :
* Zones d'entrée (service, mot de passe).
* Boutons : Save, Delete, Search, List, Logout.
* Un champ de texte pour présenter les résultats

**VII . Présentation (extrait de la console et de l'IHM) :**

* 🔐 whatsapp : 0000
* 🗑 Deleted
* ❌ Not found

#### IX . Problèmes rencontrés  :

* Gestion des identifiants préexistants («User already exists »).
* Des espaces invisibles provoquant des erreurs « Not found » lors de tentatives de suppression ou de recherche.
* Mise à jour des données sur le serveur (ajout, suppression) ;

**X.** **Possibilités d'améliorations :**

* Incorporation d'un système de cryptage pour les mots de passe stockés.
* Interface d'administration incluant un bouton de **Reset server**.
* Mots de passe : exportation et importation.
* HTTPS (protection du réseau).
* Incorporation de bases de données (MySQL, SQLite).

**XI.** **Conclusion :**

Ce travail a rendu possible l'apprentissage et la mise en application de deux paradigmes de communication à distance (RPC et RMI), tout en incorporant des concepts de sécurité, d'interface utilisateur et de gestion d'état. C'est un fondement robuste pour la création d'applications distribuées sécurisées avec Java.