SecureWin

Plan de revue finale

Introduction : Présentation des membres de l'équipe

I. Présentation des applications

Développée en méthodologie agile SCRUM du 20 septembre 2023 au 11 janvier 2024, puis de repris du **4 mars au 5 avril.**

Commandée par Fabien Laguillaumie, précédemment accompagné par Antoine Chollet, et aujourd'hui accompagné de Francis Garcia.

3 applications pour un système d'enchères en 3 rôles :

SecureWin_Bidder dédiée aux enchérisseurs,

SecureWin_Seller dédiée aux vendeurs,

SecureWin_Manager dédiée aux employés de l'entreprise puisqu'il faut que ça soit une personne de confiance.

II. Fonctionnalités

1. Principale : Enchère à pli fermé. (créer son enchères, participer, déterminer le gagnant, recevoir les résultats). Anonymat respecté.
2. **Chiffrement des prix, avec cette fois ci du chiffrement homomorphe.**
3. Signatures et authentifications.
4. Persistance des signatures (protégé par mot de passe uniquement accessible par code source)
5. Echanges sécurisés et protocoles limitant toute attaque.
6. Alerte en cas de chiffré absent (Enchérisseurs & Autorité de gestion)
7. **Interface graphique et en ligne de commande, au choix avec une option de lancement de l'application.**
8. **Conteneurisation, permettant de déployer les applications.**

Protocole @Katia

Nous n'avons pas pu terminer cette partie où l'autorité de gestion utilise le produit des chiffrés.

Système de Damgård–Jurik @Paul

Docker @Rémi

III. Développement @Loan

Nous avons utilisé la méthode agile scrum, en continuité avec les sprints précédents.

Durant tout le projet nous avons assuré le développement avec la méthode feature branch workflow.

Enfin, pour assurer la qualité du code nous avons effectué des tests et utiliser les principes SOLID et les Design Pattern.

Amélioration de la structure du code pour **l'interface graphique et en ligne de commande** et les **objets signés envoyés sur le réseau.**

IV. Choix techniques

1. Supporté uniquement par un OS Linux. (à la demande du client)
2. Java version 17. (Contient l'outil keytool qui permet de gérer la génération de certificat de signature en .jks Java KeyStore et implémente javafx)
3. Utilisation de Javafx.
4. Fichiers en .jks et .cer
5. Signature générés par RSA.
6. Fichiers relatifs à l'application contenu dans les dossiers de configurations de l'utilisateur. (permet une meilleure maintenance en cas de dysfonctionnement de

l'application d'un utilisateur, par exemple, l'utilisateur pourra supprimer les fichiers en relatif à la signature si elle a été usurpée)

V. Pistes d'amélioration

Au niveau des besoins non réalisés :

- **Utiliser le produit des chiffrés pour résoudre l'enchère comme demandé par le client.**

Niveau entreprise :

1. Étendre l'application à plusieurs OS.

Niveau sécurité :

2. Permettre aux autorités de gestion d'avoir dans leur dossier Documents un dossier SecureWin contenant l'historique des enchères (avec informations) qu'il a supervisé, et qui contiendrait aussi les alertes de enchérisseurs avec leurs coordonnées pour assurer un suivi en cas d'attaque cyber.
3. Permettre aux utilisateurs d'attribuer eux même le mot de passe de leur signature sauvegarder.

Niveau UX :

4. Développer le design de l'interface.
5. Paramétrage de l'application : Thème sombre, langue, …

VI. Démonstration @everyone