

Système de Communication Anonyme

Introduction

1 - 1 Présentation du projet

Le projet Routeur oignon consiste à développer un système de communication anonyme utilisant un mécanisme de routage en oignon, où les messages passent par plusieurs routeurs pour garantir l'anonymat des utilisateurs. Chaque routeur ne connaît que son voisin immédiat, ce qui protège l'origine et la destination des communications. Ce système vise à offrir une solution de confidentialité pour les échanges d'informations sur des réseaux publics.

1 - 2 Objectif principal du projet

Le principal objectif de ce projet est de résoudre le problème de la sécurité et de l'anonymat des communications sur des réseaux non sécurisés. En utilisant un routage multi-sauts et un chiffrement asymétrique, ce système permet de masquer l'identité des utilisateurs et de garantir la confidentialité des messages envoyés entre eux.

1 - 3 Public cible

Le projet s'adresse aux utilisateurs qui veulent protéger leur vie privée, surtout ceux qui souhaitent échanger des informations sensibles sans craindre d'être surveillés ou que leurs messages soient interceptés sur Internet.

Répartition du travail

Analyse et planification :

- Comprendre les exigences du projet.
- Définir les objectifs et les fonctionnalités principales.
- Organiser le planning de développement.

Développement du système de routage :

- Implémenter le mécanisme de routage en oignon (routage multi-sauts).
- Développer la logique de communication entre les clients et les routeurs.

Chiffrement des messages :

- Concevoir et implémenter un système de chiffrement asymétrique (clé publique/privée).
- Assurer l'anonymisation des messages en ajoutant plusieurs couches de chiffrement.

Développement de l'interface utilisateur (Qt) :

- Créer une interface graphique pour le client et le serveur maître.
- Permettre aux utilisateurs de visualiser les connexions et l'état du réseau.

Base de données (MariaDB) :

Configurer la base de données pour stocker les clés de chiffrement et les tables de routage.
Intégrer la base de données avec le système pour gérer les informations de routage.

Gestion du code et tests :

Suivre l'évolution du projet via GitHub pour versionner le code.
Tester chaque partie du système (connexion, chiffrement, routage, etc.).

Documentation :

Rédiger la documentation technique (explication du code, des algorithmes et des choix techniques).
Créer un guide d'installation et d'utilisation.
Préparer un rapport de gestion du projet .

Vidéo de démonstration :

Préparer une vidéo montrant le fonctionnement du système : lancement du serveur, envoi de messages anonymes, visualisation des connexions.

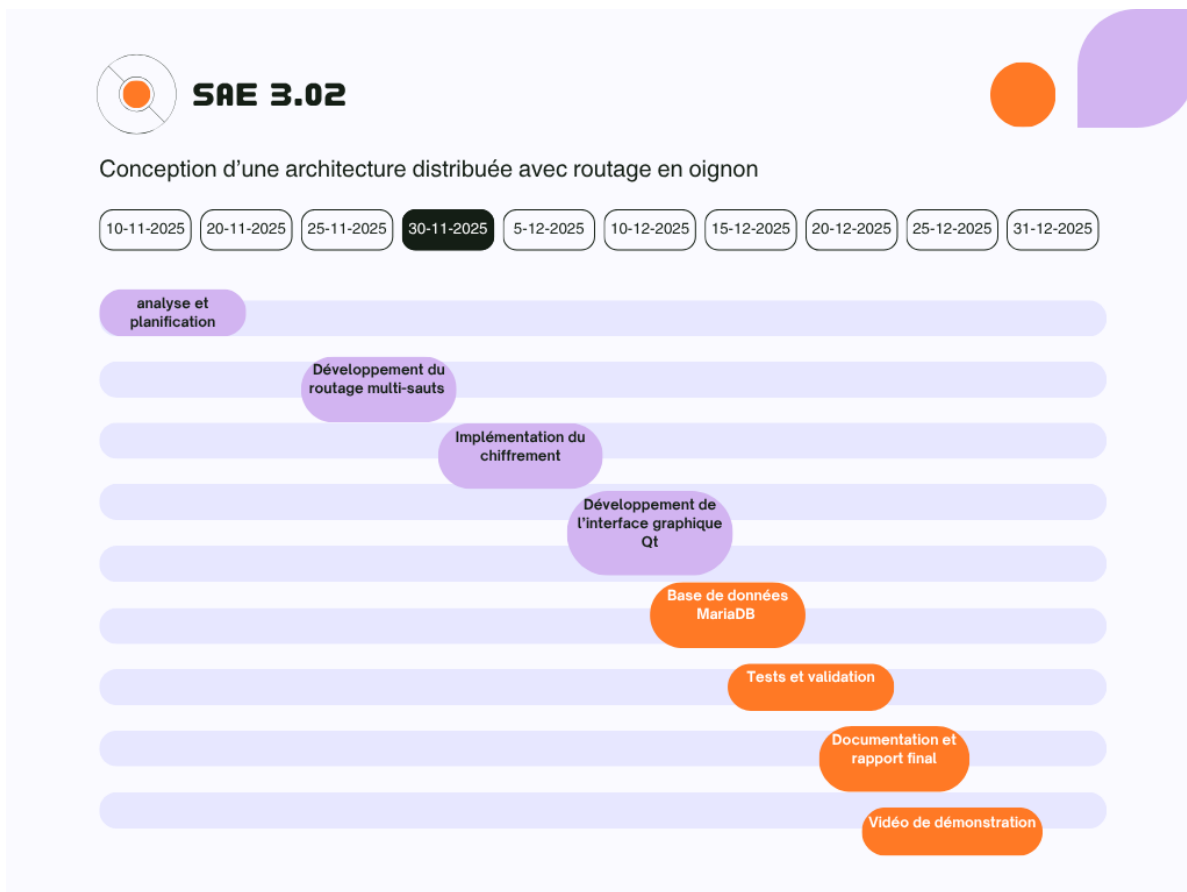
Fonctionnalités prévues

Fonctionnalité	Description courte	Priorité	Dépendances éventuelles
Routage multi-sauts	Permet aux messages de traverser plusieurs routeurs pour garantir l'anonymat.	Essentielle	Mise en place des routeurs et communication entre eux.
Chiffrement asymétrique (clé publique/privée)	Chiffrement des messages avec des clés publiques/privées pour sécuriser les échanges.	Essentielle	Mise en place des routeurs, base de données pour stocker les clés.
Interface graphique Qt pour le client	Création d'une interface graphique pour envoyer et recevoir des messages de manière anonyme.	Secondaire	Fonctionnalités de routage et chiffrement.
Base de données MariaDB	Stockage des clés de chiffrement et des tables de routage dans une base de données.	Essentielle	Développement des routeurs et chiffrement des messages.
Gestion de la communication simultanée (multi-threads)	Gérer plusieurs connexions clients en même temps pour assurer un système fluide.	Essentielle	Implémentation du serveur maître et des routeurs.

SAE 3.02 Conception d'une architecture distribuée avec routage en oignon

Journalisation des échanges	Enregistrer des logs des échanges pour surveiller les messages envoyés et reçus.	Secondaire	Mise en place de la communication entre clients et routeurs.
Tests et validation	Vérification du bon fonctionnement du système (routage, chiffrement, base de données).	Essentielle	Toutes les fonctionnalités développées.
Documentation et rapport final	Rédiger un rapport complet sur le projet, expliquer le fonctionnement et les choix techniques.	Essentielle	Implémentation des fonctionnalités principales.
Vidéo de démonstration	Créer une vidéo montrant le système en fonctionnement, avec un focus sur l'anonymisation des messages.	Secondaire	Mise en place complète du système.

Planification :



Organisation et outils

GitHub sera utilisé pour la gestion du code source. Je vais effectuer des commits réguliers pour suivre les modifications et garantir la traçabilité du développement.

Communication :

Comme je travaille seul, il n'y a pas besoin d'un outil de communication d'équipe. Cependant, en cas de besoin, j'utiliserai Discord pour échanger avec d'autres camarades sur le projet.

Gestion de projet :

Trello : J'utilise Trello pour organiser les tâches et les jalons du projet. Je crée des cartes pour chaque étape, avec des dates d'échéance et des descriptions détaillées, ce qui me permet de suivre l'avancement et de respecter les délais.

Suivi personnel :

Je n'ai pas de réunions régulières avec une équipe, mais je vais suivre l'avancement du projet de manière quotidienne en prenant des notes manuelles. Ces notes me permettront de suivre les tâches accomplies, les problèmes rencontrés, et d'ajuster mon planning si nécessaire.

Planification et révisions :

Toutes les semaines, je ferai une révision personnelle pour évaluer l'avancement, identifier les éventuels blocages, et ajuster mes priorités.

Risques identifiés

Complexité technique:

La mise en place du **routage en oignon** et du chiffrement asymétrique peut être complexe, surtout pour s'assurer de la sécurité et de l'efficacité du système.

Erreurs dans les tests:

Il est possible que certaines erreurs passent inaperçues durant les **tests**. Cela pourrait affecter la qualité du système final, surtout en ce qui concerne la gestion du chiffrement et l'anonymisation des communications.

Moyens envisagés pour les réduire :

Complexité technique :

Pour réduire ce risque, je vais bien noter chaque étape du développement et faire des tests fréquents dès que je commence à coder, pour vérifier que mes choix techniques sont bons.

Erreurs dans les tests:

Pour éviter les erreurs non détectées, je vais faire des tests réguliers tout au long du projet, plutôt que d'attendre la fin pour tout tester d'un coup.