Document de réponse

Projet SAE 3.02 – Routage en oignon

Nom du groupe : πThon
Membres du groupe :
- HENRY Aurélien
- HALTER Mathis

Table des matières

Présentation du projet 1
Eléments implémentés et non implémentés 2
Structure du code, modules, protocole, API 2
Bibliothèques utilisées : 2
Description de l'algorithme de chiffrage (forces et faiblesses) 3
Rapport de projet avec gestion du projet 3
Organisation du travail 3
Suivi et outils utilisés 4
Conclusion 4

Présentation du projet

Le projet SAE 3.02 « Développer des applications communicantes » a pour objectif la conception et l'implémentation d'un système de communication client–serveur anonyme, inspiré du principe de routage en oignon (Onion Routing).

L'architecture repose sur plusieurs entités :

- un Master (serveur maître),
- plusieurs routeurs virtuels,
- des clients capables d'échanger des messages de manière anonyme.

Chaque message est chiffré en plusieurs couches successives et transite par une chaîne de routeurs. Chaque routeur ne connaît que son voisin direct, garantissant ainsi l'anonymat de la communication.

Eléments implémentés et non implémentés

Nous avons implémenté les éléments suivants :

- Routage multi-sauts avec sockets Python
- Gestion multithread des connexions
- Chiffrement asymétrique (clé publique/privée simplifiée)
- Anonymisation par couches (routage en oignon)
- Base de données MariaDB (clés, tables de routage)
- Interface Qt (visualisation des connexions, statistiques, client)

Nous n'avons pas implémenté les éléments suivants :

- Faire en sorte qu'un client soit aussi routeur.

Structure du code, modules, protocole, API

Bibliothèques utilisées :

Les bibliothèques Python utilisées dans le projet sont :

- random : génération aléatoire (choix des routes),
- math : opérations mathématiques liées au chiffrement,
- socket : communication réseau TCP,
- threading : gestion des connexions simultanées,
- time : temporisation et gestion des délais,
- sys : récupération des arguments de la ligne de commande,
- PyQt5 : création des interfaces graphiques du client et du master,
- Mysql.connector : Assure la connexion et les échanges avec la base de données.

Aucune bibliothèque externe de cryptographie n'a été utilisée, conformément aux contraintes du cahier des charges.

Description de l'algorithme de chiffrage (forces et faiblesses)

Le projet utilise un chiffrement asymétrique simplifié, inspiré de l'algorithme RSA. Chaque routeur génère une paire de clés (publique / privée). La clé publique est transmise aux clients via le Master, tandis que la clé privée reste uniquement connue du routeur.

Les clients chiffrent les messages en plusieurs couches successives (routage en oignon). Chaque couche est chiffrée avec la clé publique d'un routeur du chemin. Lors du transit, chaque routeur ne déchiffre qu'une seule couche, ce qui lui permet uniquement de connaître le prochain saut, sans accéder au message complet.

Le chiffrement est effectué sur les données du message à l'aide d'opérations mathématiques RSA, puis les données sont transmises sous forme de chaînes de caractères via les sockets.

Forces :

- Respect du principe du chiffrement asymétrique
- Anonymisation efficace grâce aux couches de chiffrement
- Implémentation conforme aux contraintes du cahier des charges (sans bibliothèque externe)
- Approche pédagogique claire

Faiblesses :

- Tailles de clés réduites, peu sécurisées en conditions réelles
- Chiffrement simplifié, sans mécanismes avancés (padding, signatures)
- Solution non adaptée à un usage en production

Rapport de projet avec gestion du projet

Organisation du travail

Le projet a été réalisé en groupe, avec une répartition claire des tâches :

- HENRY Aurelien

Rôle: chef de projet, dev front, dev back, design, test

Tâches principales:

- Routage multi-sauts avec sockets Python
- Gestion multithread des connexions

- Base de données MariaDB (clés, tables de routage)
- Interface Qt (visualisation des connexions, statistiques, client)

- HALTER Mathis

  Rôle: dev front, dev back, design, test

Tâches principales:

- Chiffrement asymétrique (clé publique/privée simplifiée)
- Anonymisation par couches (routage en oignon)
- Base de données MariaDB (clés, tables de routage)
- Interface Qt (visualisation des connexions, statistiques, client)

Suivi et outils utilisés

- Utilisation de Git/GitHub pour le visionnement du code
- Respect du planning de développement (diagramme de GANTT dans le rendu préparatoire du projet)

Conclusion

Ce projet nous a permis de mettre en pratique :

- la programmation réseau,
- la gestion asynchrone avec les threads,
- les principes fondamentaux du chiffrement et de l'anonymisation.

L'implémentation respecte les objectifs pédagogiques et techniques définis dans le cadre de la SAE 3.02.