Rapport de Projet

Conception d'un VPN

Projet de 3e année de licence info 2022-2023

13 février 2023

Réalisé par Pauline DEHORS (22000055)

Table des matières

Réalisé par Pauline DEHORS (22000055)

I. Introduction.

Ce rapport de projet a pour but d'introduire et de décrire le projet de conception d'un VPN dans le cadre d'un enseignement universitaire. Ce document vise à présenter les différents aspects du projet, ainsi que les méthodes et techniques utilisées pour atteindre les objectifs fixés.

Le rapport est organisé de manière claire et concise pour permettre une lecture aisée. Il débute par une présentation du contexte et de l'objectif du projet. Ensuite, les différentes étapes du projet sont détaillées, incluant la méthodologie utilisée, les résultats obtenus et les conclusions qui en découlent.

Le document s'adresse à un public cible d'étudiants en informatique et de professionnels de l'informatique. Il vise à fournir une description complète et détaillée du projet, pour permettre aux lecteurs de comprendre les différents aspects du projet et les défis rencontrés.

En conclusion, ce rapport vise à fournir une vue d'ensemble complète du projet de conception d'un VPN. Il se veut être un outil pratique pour les personnes souhaitant en savoir plus sur les différentes étapes du projet et les techniques utilisées.

II. Analyse des écarts.

A. Objectifs techniques.

Dans le cahier des charges de ce projet, il a été clairement énoncé la mise en place de plusieurs modules pour remplir les objectifs fixés. Cette section du rapport a pour but de rappeler les fonctionnalités définies ainsi que de fournir une évaluation critique de la solution mise en place. Nous allons donc explorer les différentes parties du système et en tirer une analyse approfondie pour évaluer leur pertinence et leur efficacité dans le cadre du projet. Cette démarche permettra de mettre en avant les points forts et les points à améliorer pour atteindre les objectifs de manière optimale.

Vous trouverez ci contre une vue générale :

Fonctionnalités	**Statut**	**Améliorations**	**Implémentation des améliorations**
Connexion/ Déconnexion du VPN	✔️	Console qui retrace tous les échanges avec le serveurs	✔️
Configuration des sockets (clients et serveurs)	✔️	Multithreading des clients et mise en réseau	✔️
Méthode de Diffie-Hellman	✔️	Aucune	
Chiffrement/ Déchiffrement des données	✔️	Utiliser un algorithme de cryptographie plus complexe	✔️
		Ajouter une complexe d'intégrité	✔️
		Ajouter une signature	✔️
Speedtest	✔️	Rendre le compteurs dynamique	✔️
Graphique du trafic d'usage du réseau	✔️	Sauvegarde des données dans un SBD	✔️
Répertorier les	✔️	Ajouter un système de	✔️

documents échangés		messagerie	
Transfert de fichiers	✔️	Côté expéditeur : indiquer le destinataire via la liste de contact	✔️
		Côté destinataire : stockage du fichier dans le dossier courant	✔️
Interface	✔️	Ajout des différents modules ajouter ci-dessus	✔️

<u>La connexion et la déconnexion du VPN :</u>
Temps passé estimé sur l'implémentation : 5h
Temps passé sur l'amélioration : 8h
Un des objectifs était de permettre une activation et une désactivation facile du VPN. Pour ce faire, une interface utilisateur a été mise en place avec un bouton ON/OFF pour activer ou désactiver le VPN en un seul clic.

L'objectif d'avoir un bouton de contrôle facile a été atteint sans aucune difficulté, ce qui m'a encouragé à aller plus loin dans la mise en œuvre d'autres fonctionnalités pour améliorer l'expérience utilisateur. Pour suivre les échanges entre le client et le serveur, une console a été implémentée qui enregistre toutes les actions, telles que la connexion et la déconnexion du VPN.

Cette console offre une vue détaillée sur toutes les activités du VPN, ce qui permet aux utilisateurs de surveiller en temps réel la sécurité de leur connexion. De plus, cela facilite la résolution de tout problème éventuel qui pourrait survenir pendant l'utilisation du VPN.

En résumé, l'objectif initial de permettre l'activation et la désactivation facile du VPN a été atteint grâce à l'ajout d'un bouton ON/OFF dans l'interface utilisateur. Ce premier objectif a été approfondi en implémentant une console qui enregistre toutes les activités du VPN, offrant ainsi une meilleure expérience utilisateur et une sécurité accrue des données en ligne.

<u>La configuration des sockets clients et du sockets serveur :</u>
Temps passé estimé sur l'implémentation : 5h
Temps passé sur l'amélioration : 7h

Réalisé par Pauline DEHORS (22000055)

Dans le cadre de ce projet VPN, j'ai d'abord mis en place la synchronisation des sockets pour échanger des données entre les serveurs et les clients. J'ai accordé une importance particulière à la protection des différents types de données transmises. Cependant, pour optimiser le système, j'ai dû trouver une solution pour connecter plusieurs clients simultanément au serveur.

Au départ, j'ai songé à utiliser des threads pour ce faire, mais malheureusement, cette idée n'a pas abouti en raison du Global Interpreter Lock (GIL) de Python. Ce verrou unique limite l'exécution des threads, ce qui rend impossible la connexion simultanée de plusieurs clients.

Pour remédier à ce problème, j'ai cherché une autre solution pour gérer les données concurrentes entre les threads, mais je n'ai pas réussi à les déclarer dans la mémoire partagée. J'ai donc implémenté le système de multithreading avec des sous-processus. Il n'est donc pas en multithreading mais en multiprocessus.

Enfin, pour vérifier que le système était fonctionnel sur des réseaux distants, j'ai effectué des tests en configurant une redirection de port sur ma box internet et en modifiant les données de connexion du client et du serveur en conséquence. Les résultats de ces tests sont concluants et montrent la faisabilité d'une utilisation du VPN sur des réseaux éloignés.

En résumé, la configuration des sockets a été un succès relativement simple comparé au multithreading auquel j'ai dû trouver une alternative, s'ajoute à ça la mise en réseau distant tester avec succès.

L'échange des clés avec la méthode de Diffie-Hellman :

Temps passé estimé sur l'implémentation : 12h

L'algorithme de Diffie-Hellman est un système de cryptage de clé publique qui a été développé pour résoudre un problème fondamental en sécurité informatique : comment deux parties peuvent échanger des informations de manière sécurisée sans que la clé ne soit interceptée par une tierce partie.

La première solution qui s'offrait à moi était d'utiliser différentes librairies qui permettent de générer et d'échanger les clés à l'appel de fonction, toutefois je trouvais important d'implémenter moi même ces fonctions me permettant alors de vérifier ma compréhension de cette méthode.

Bien que la mise en place de cet algorithme soit relativement simple, le véritable défi réside dans la synchronisation des clés publiques pour garantir un échange de données

réussi. En effet, pour que ce système fonctionne correctement, il est nécessaire que les sockets soient parfaitement synchronisés pour permettre l'échange de clés publiques en toute sécurité.

En somme, la complexité de ce système ne réside pas dans la mise en place de l'algorithme en lui-même, mais dans la nécessité de garantir une bonne coordination des sockets pour garantir la sécurité des données transmises.

Le chiffrement et le déchiffrement des données :

Temps passé estimé sur l'implémentation : 15h
Temps passé sur l'amélioration : 27h

Le module principal du VPN est bien évidemment le chiffrement et le déchiffrement des données pour permettre une sécurisation optimale des données échangées entre les clients et le serveur.

Le premier algorithme de cryptographie utilisé a été l'algorithme de Caesar Cipher qui repose sur le principe de décalage des caractères selon la clé de chiffrement partagée. Cet algorithme a été implémenté avec succès sans utiliser de librairie python et sans grandes difficultés puisqu'il m'a simplement fallu reprendre un de mes projets déjà existants.

Cependant, dans une optique d'optimisation du projet, j'ai échangé sur la robustesse que nécessitait un logiciel VPN avec Monsieur Dequen. Ce dernier m'a alors proposé les idées suivantes pour suivre les garanties cryptographiques:

- Garantir la **confidentialité** avec une primitive symétrique c'est-à dire changer l'algorithme de chiffrement pour un algorithme plus robuste, en effet, l'algorithme de Caesar Cipher n'étant pas assez complexe, il a fallu trouver un autre algorithme plus robuste.

  Avec l'avis d'un professeur de l'université, mon choix s'est finalement porté sur l'utilisation de l'AES, et après plusieurs recherches, celles-ci m'ont affirmées que l'implémentation de cette algorithme avec les éléments de bases de Python n'est pas recommandée car elle pourrait présenter des failles de sécurité importantes. Pour affirmer c'est dire, j'ai demandé l'avis de Monsieur Dequen qui m'a affirmé les potentielles failles de sécurité que pourrait engendrer une implémentation de ma part. C'est pourquoi j'ai fait le choix d'utiliser la librairie Cipher.AES.

Cette optimisation m'a coûté cher en temps, en effet j'ai rencontré différent problème et j'ai dû revoir l'entière structure des échanges de données entre le serveur et les différents clients. En effet les nouvelles fonctions implémentées nécessitent d'autres valeurs (à savoir le nonce et le tag pour le chiffrement et le déchiffrement des données) or l'implémentation de base se contenter d'envoyer les données uniquement puisque l'algorithme de Cipher n'a besoin que de la clé partagé pour déchiffrer les données correctement.

- Garantir **l'intégrité** avec une primitive de hachage qui me permettra de vérifier si les données n'ont pas été modifiées pendant leur transfert par une tierce personne (notamment grâce à une attaque Man In The Middle).

  Toujours avec l'avis de Monsieur Dequen, j'ai opté pour le SHA256 pour hacher les données qui est la fonction de hachage connue pour sa robustesse.
  L'implémentation de ce test d'intégrité s'est faite assez simplement et relativement rapidement car il s'agit juste d'une vérification entre le hash envoyé et le hash retourné par la fonction de hachage sur les données reçues.

- Garantir **l'authenticité** avec une signature de la source qui permettra de vérifier si la personne qui a envoyé la signature est bien la bonne et inversement.

  Avec les conseils de Monsieur Dequen, j'ai pu comprendre la logique à priori simple mais relativement complexe tout de même qui se cachait derrière une simple signature.

  Le premier point important à comprendre à été de générer un nouveau couple de clés lié avec la librairie RSA afin de pouvoir déchiffrer avec la clé publique un message qui a été chiffré avec la clé privé du client à l'origine de la signature.

  Une fois cette logique comprise, l'implémentation de ce module a été réalisée sans rencontrer de problème, si ce n'est de comprendre l'utilisation des fonctions de la librairie utilisée.

<u>Mise en place des Speedtests :</u>
Temps passé estimé sur l'implémentation : 6h
Temps passé sur l'amélioration : 7h
La mise en place d'un test de rapidité de la vitesse a pour objectif de donner un ordre d'idée à l'utilisateur sur la vitesse de transfert de données en Mbit/s. La première

étape a été de comprendre le calcul à fournir pour pouvoir avoir une vitesse cohérente. J'ai d'ailleurs dû revoir à plusieurs reprises la méthode de mon calcul avant d'avoir un résultat qui me convenait.

Toutefois la partie la plus complexe mais aussi celle qui m'a demandé le plus de temps n'était pas le calcul mais plutôt l'interface. La première idée était de fournir un cadran qui afficherait la vitesse une fois le calcul établi. Après avoir réussi à implémenter une interface qui me convenait sans grandes difficultés j'ai voulu pousser l'expérience et implémenter une petite animation, celle-ci consiste à faire tourner l'aiguille comme un compteur de vitesse. Après y avoir passé de longues heures, j'ai finalement réussi à implémenter exactement l'idée que j'avais en tête.

Pour résumer le développement de ce module, on peut dire que son implémentation n'a pas été compliquée mais a demandé beaucoup de temps dû à l'implémentation de l'interface. Idem pour la programmation de l'amélioration imaginé qui a demandé encore plus de temps;

Graphique du trafic réseau :

Temps passé estimé sur l'implémentation : 17h

Dans un premier temps, rappelons l'idée principale et le résultat attendu de ce graphique. L'idée était de fournir un histogramme qui retrace la quantité de données échangées sur les 7 derniers jours de connexion de l'utilisateur.

Concernant l'implémentation de l'interface, celle-ci a pu se faire sans problème et a abouti au résultat imaginé dans la maquette fournie dans le cahier des charges. Toutefois il a fallu rendre ce module redimensionnable et c'est dans cette tâche que j'ai passé beaucoup de temps.

Ensuite il a fallu implémenter le stockage des données, sachant que je voulais retrouver les données stockées même après avoir fermer le programme deux solutions s'offraient à moi :

- ➔ Ecriture dans un fichier
- ➔ Stockage dans une base de donnée

J'ai donc ici pris la décision de mettre en place une base de données en utilisant la librairie SQLight qui permet de simplifier les interactions et la création de la base de données. Une fois cette dernière mise en place, il a juste fallu stocker et utiliser les données correctement dans le programme.

Pour conclure sur ce module on peut dire que la plus grande difficulté rencontrée a été de le rendre responsive tout en continuant d'avoir un graphique qui utilise les bonnes données. La deuxième difficulté qu'aurait pu être la base de données ne l'a finalement pas été grâce à la librairie qui a permis de simplifier toute son utilisation.

Répertorier les documents échangés :

Temps passé estimé sur l'implémentation : 6h
Temps passé sur l'amélioration : 13h

L'idée de ce module était de répertorier chaque fichier envoyé ou reçu dans une partie dédiée de l'interface. Le développement de cette idée avait été globalement réussi, toutefois étant donné qu'il s'agissait dans un premier temps de remplir le vide, ce module n'a été que temporaire pour laisser place à d'autres améliorations.

Sur la version fournie vous ne trouverez pas ce module car il a été remplacé par la possibilité de mise en place de nouvelles applications comme celle de la boite mail.

Dans l'idée que ce projet puisse réellement répondre au besoin des utilisateurs d'un VPN d'entreprise, j'ai trouvé important d'ajouter la capacité à pouvoir échanger des mails ou des messages de manière sécurisée au sein d'une entreprise. Encore une fois, le plus difficile a été l'implémentation de l'interface et les interactions interface/utilisateur et c'est pour cela que cette application ne propose que les fonctionnalités principale d'une boîte mail qui sont à mon sens les suivant :

- ➔ Envoi et réception de message
- ➔ Consultation des messages
- ➔ Répondre à un message
- ➔ Ajouter des contacts à la liste de contacts

Je n'ai pas rencontré de grands problèmes pendant le développement de ce module. Mise à part l'interface, le plus long a été de s'assurer de la sécurité des échanges grâce à différents tests.

Transfert de fichier:

Temps passé estimé sur l'implémentation : 14h

Le transfert de fichier est l'un des modules les plus importants dans l'utilisation d'un VPN et c'est pourquoi j'ai pris le temps de développer un module fonctionnel mais surtout sécurisé.

Dans un premier temps j'ai simplement mis en place un système d'envoi de fichier chiffré avec la clé partagé au serveur qui doit le déchiffrer de son côté. Cette partie ne

m’a pas posé de problème que ce soit tant la partie réseau que la partie chiffrement/déchiffrement (qui a ce moment-là se faisait encore avec l’algorithme de Caesar Cipher). La deuxième étape a été de transférer le fichier au bon utilisateur en passant par le serveur et c’est à cette étape que j’ai rencontré un problème.

Le principe est simple, le client envoie le fichier qu’il souhaite ainsi que le destinataire (celui à qui il souhaite envoyer le fichier) au serveur, le tout étant chiffré avec la clé partagé et l’algorithme de l’AES. Une fois que le serveur à reçu le fichier, il le déchiffre et le stocke dans la base de données avec les informations relatives à ce transfert (source, destinataire,nom de fichier…). Lorsqu’il reçoit une demande du client B pour recevoir les fichiers qui lui ont été envoyés, le serveur chiffre le fichier avec la clé partagée de ce client et le lui envoie.

Maintenant que le principe est expliqué, voici le problème que j’ai rencontré :
Lors du transfert de données via les socket, les données doivent être encodables et décodables en utf-8, or pour certain caractère, l’algorithme de l’AES peut renvoyé des symbole qui ne rendre pas dans ce cadre (UnicodeDecodeError : ‘utf8’ codec can’t decode byte 0xa9…). J’ai passé plusieurs heures à essayer de trouver une solution jusqu’à trouver l’idée suivante : il existe un argument de la fonction encode() qui permet de spécifier à celle-ci de ne pas prendre en compte les caractères qu'elle ne peut pas encoder. Celle-ci les laissant tel quel, l’envoi et la réception des données à pu fonctionner sans poser de problème de déchiffrement avec l'algorithme de l’AES.

B. Objectifs organisationnels.

Il a été très vite flagrant que le planning ne serait pas tenu et il y a, selon moi, plusieurs raison à ça :

- ➢ Une **surestimation de la charge de travail** pour chaque étape de développement. Vous verrez dans le diagramme ci-dessus qu’un délai de 9 semaines avait été estimé pour le développement des modules principaux (diffie-Hellman, chiffrement/déchiffrement avec Caesar Cipher, multi-processus, speedtest upload/download, stockage et calculs en lien avec le diagramme du trafic du réseau). A l’instar de l’estimation du backend, la partie frontend a été aussi très surestimée, voyez-ci dessus une estimation de 10 semaines pour seulement 2 réellement.

- ➢ Un **investissement en temps et en effort démesuré**. Bien que je n'aie pas enregistré mes heures, je suis consciente que j'ai travaillé plus longtemps que

ce qui était initialement prévu pour certaines tâches et cela parce que ce projet m'a vraiment intéressé ce qui m'a permis de m'y investir à fond. Ce projet devait être réalisable dans un minimum de 50 heures, je pense pouvoir vous affirmer que mon temps de travail sur ce projet avoisine les 20h par semaines soit un total de 200 heures à apprécier de travailler sur ce projet. De surcroît, cet investissement supplémentaire a permis de maintenir le projet sur la bonne voie tout en l'optimisant considérablement.

Ci-dessus vous trouverez un diagramme de Gantt qui retrace l'évolution du développement de ce projet en temps réel *(voir les bulles de couleurs)* mais aussi un rappel des temps estimés dans le diagramme de Gantt du cahier des charges *(voir les bulles hachurées*). Ainsi vous pourrez comparer ces deux données et constater les écarts considérables entre le temps estimé et le temps réellement passé sur les différentes étapes de ce projet.

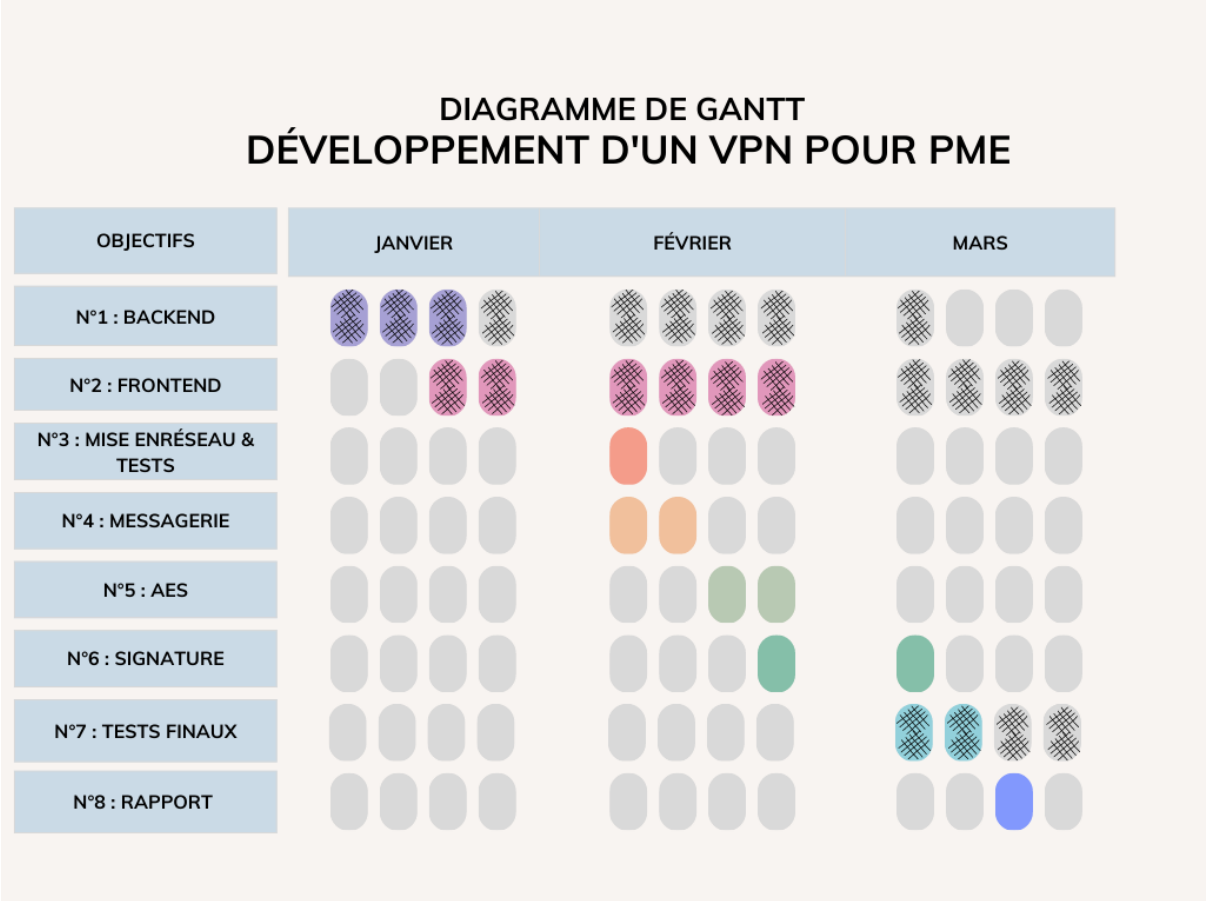

Diagramme de Gantt du projet VPN qui retrace le temps estimé et le temps utilisé.

Détails :

N°1 : Backend

- ➔ Diffie-Hellman
- ➔ Chiffrement/Déchiffrement avec Caesar Cipher
- ➔ Multi-processus
- ➔ Speedtest upload/download
- ➔ Stockage et calculs en lien avec le diagramme du trafic du réseau

N°2 : Frontend

- ➔ Interface
- ➔ Fusion du backend et de l'interface

N°3 : Mise en réseau & Tests

- ➔ Redirection de ports en passant par la box internet
- ➔ Première phase de test résolution des erreurs qui y sont survenues :
  - ◆ Problème interface, calculs et récidive des speedtests
  - ◆ Problème de réception du fichier
  - ◆ Problème responsive du diagramme du trafic réseau
  - ◆ Problème de reconnexion après déconnexion
  - ◆ Problème de stockage des données dans les BDD

N°4 : Messagerie :

- ➔ Ajout d'un module "Boite Mail" dans l'interface
- ➔ Veillez au bon chiffrement/déchiffrement des données
- ➔ Système de notification (pastille rouge) et de gestion de reception des nouveaux messages
- ➔ Stockage des informations dans la BDD

N°5 : AES :

- ➔ Implémentation d'une nouvelle fonction de chiffrement/déchiffrement en utilisant la librairie python pour l'AES
- ➔ Révision de la structure des échanges de données (problème de nonce et de tag)

N°6 : Signature :

- ➔ Mise en place de la signature à la connexion
  - ◆ Implémentation d'une fonction qui permette d'envoyer un message signé ainsi que la clé publique en utilisant une librairie python pour le RSA

N°7 : Tests finaux :

- ➔ Première phase de tests : Interface/Utilisateur
  - ◆ Révision de la méthode de calcul des speedtests
  - ◆ Histogramme non ergonomique visuellement
  - ◆ Mise à jour des messages de la console du client
  - ◆ Résolution d'un problème d'affichage de la fenêtre des mails
  - ◆ Résolution d'un problème de sauvegarde dans la BDD après modification
- ➔ Deuxième phase de tests : Partage de données en réseau
  - ◆ Mise à jour du menu déroulant des contacts
  - ◆ Résolution du problème d'encodage de certains caractères pour l'envoi de fichier
  - ◆ Résolution d'un problème dans le stockage de la taille des données échangées

N°8 : Rapport :

- ➔ Rédaction du rapport de projet
- ➔ Création des slides de présentation du projet
- ➔ Rédaction d'un ReadMe pour installer/configurer le VPN

C. Analyse.

Le projet à bien été une réussite pour plusieurs raisons. Tout d'abord, grâce à une charge de travail importante qui a permis d'avancer rapidement et efficacement. J'ai travaillé avec attention pour assurer que chaque étape du projet était développée avec soin. La charge de travail importante m'a permis de rester concentrée et de maintenir un niveau élevé de productivité tout au long du projet.

Ensuite, le fondement du projet était basé sur des recherches précises et importantes que j'ai menées moi-même. J'ai effectué des recherches approfondies sur les outils nécessaires pour le projet et cette recherche m'a permis de sélectionner les outils les plus appropriés pour le projet, de garantir que le projet serait réalisé avec les dernières technologies de sécurité.

Malgré la réussite du projet, j'ai dû faire face à des contraintes et des risques. La mise en place d'une base de données était une tâche cruciale pour le projet, mais elle n'avait pas été anticipée. Cela a entraîné un retard dans le projet, mais j'ai pu résoudre rapidement ce problème à la suite de nombreuses recherches qui m'ont permis de découvrir la librairie qui est utilisée dans le projet pour faciliter cette tâche.

Un autre défi rencontré était la nécessité d'adapter complètement la structure d'échange de données pour supporter les nouvelles fonctions avec l'AES. Cette tâche était complexe et nécessitait de nombreuses heures de réflexion. Heureusement, grâce à mes connaissances techniques et à ma détermination, j'ai pu surmonter ce défi avec succès.

Enfin, tout au long du projet, j'ai bénéficié de précieux conseils de la part de différentes personnes (professeurs, collègues, professionnels de la sécurité). Leurs conseils m'ont aidé à orienter le projet dans la bonne direction et à travailler plus efficacement. Je suis fière d'avoir mené ce projet à bien et d'avoir surmonté les défis rencontrés tout au long du processus.

III. Retour d'expérience.

- **Apprentissage des fondamentaux de la cryptographie des données** :
  Au cours de ce projet, j'ai eu la chance de me plonger dans les fondamentaux de la cryptographie des données, notamment l'AES, le RSA, les signatures et l'intégrité. Cette expérience a été très enrichissante pour moi, car j'ai pu approfondir mes connaissances théoriques dans ce domaine et les mettre en pratique dans un contexte concret. Grâce à cela, j'ai pu comprendre l'importance de la sécurité dans les systèmes informatiques, ainsi que les différents outils qui peuvent être utilisés pour la garantir.

- **Acquisition de connaissance et de pratique en réseau** :
  J'ai pu configurer moi-même un réseau local à partir de ma box internet, cette pratique a été très enrichissante pour mes compétences et mes connaissances personnelles et cela m'a permis de réaliser que j'apprécie finalement plus que ce que je ne pensais la complexité de ce domaine.

- **Acquisition de compétences en gestion de projet** :
  Ce projet m'a également permis d'acquérir des compétences en gestion de projet. J'ai appris à mettre en place un git pour le suivi de version de mon projet et à maintenir une veille constante pour garantir son avancement.

- **Apprentissage à partir des erreurs** :
  Bien que le projet se soit globalement bien déroulé, j'ai également rencontré quelques difficultés au cours de son exécution. Toutefois, j'ai appris à partir de mes erreurs et à en tirer des leçons utiles pour l'avenir. J'ai par exemple compris l'importance de commenter régulièrement mon code pour faciliter la compréhension et la maintenance. J'ai également appris à ne pas m'obstiner

quand je suis bloquée, et à chercher des solutions alternatives ou à demander de l'aide si nécessaire. Ces expériences m'ont permis de m'améliorer en tant que développeur et de mieux comprendre les pratiques de travail efficaces.

➢ **Confiance en moi** :
Enfin, ce projet m'a permis de renforcer ma confiance en moi et en mes capacités. C'est peut-être l'un des retours d'expérience auquel je ne m'attendais pas mais sans aucun doute le plus important. Au début du projet, je doutais de mes compétences et je pensais que je ne pourrais pas produire un projet aussi complet et complexe. Toutefois, grâce à l'expérience acquise au cours du projet, j'ai pu me prouver à moi-même que j'étais capable de mener à bien ce genre de tâches et d'atteindre mes objectifs. Cette expérience a été très positive pour moi, car elle m'a permis de réaliser mon potentiel et de me motiver à poursuivre mes efforts dans le domaine du développement informatique.

IV. Conclusion.

Pour conclure sur le rapport de ce projet, il est important de noter que ce dernier à atteint et même dépassé tous les objectifs qui lui avaient été désignés grâce à un effort régulier et important. Toutefois, ce projet peut encore être amené à évoluer.

➢ **Ajout de modules supplémentaires** :
Il serait intéressant d'ajouter d'autres modules pour offrir une expérience utilisateur plus complète et diversifiée. Par exemple, il serait possible d'ajouter un accès au web pour permettre aux utilisateurs d'accéder à certaines fonctionnalités à partir de leur navigateur tout en leur garantissant une sécurité optimale. Il serait également possible de mettre en place une solution de visioconférence sécurisée pour faciliter la communication entre les différents utilisateurs du VPN. Ces fonctionnalités supplémentaires pourraient améliorer la valeur ajoutée du projet et en faire un produit plus complet et attractif pour les entreprises.

➢ **Amélioration de l'interface** :
Une autre amélioration possible pour le projet serait d'améliorer l'interface utilisateur pour la rendre plus conviviale et facile à utiliser. Bien que l'interface actuelle soit fonctionnelle, elle pourrait être optimisée pour offrir une expérience utilisateur plus fluide et intuitive. Cela pourrait inclure des changements de design, une organisation plus claire des menus et des options, ainsi que des améliorations de l'ergonomie générale de l'interface. Ces

améliorations pourraient rendre le projet plus facile à utiliser pour les utilisateurs et améliorer leur expérience globale.

- **Optimisation des bases de données** :
  Enfin, une autre amélioration possible pour le projet serait d'optimiser les bases de données pour améliorer les performances et la stabilité du système. Cela pourrait inclure des optimisations de la structure des tables, des améliorations de la gestion des index et des relations entre les tables, ainsi que des améliorations des algorithmes de recherche et de tri des données. Ces optimisations pourraient améliorer les performances globales du projet et rendre son fonctionnement plus fluide et plus rapide pour les utilisateurs.