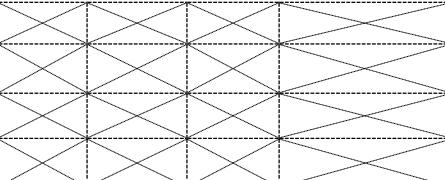


Niveau Post-Bac	A.R	Fiche Individuelle SAE 3.02	Documentation	1	1
Informatique	31/12/2025				

Fiche Individuelle SAE 3.02

Date: 31/12/2025	Durée conseillée: 30m	Durée du travail: 1h
Champs d'activité:	Définition des savoirs:	Niveaux Taxonomique:
CA-1.0: Gestion de l'Administration: <i>CA-1.0.1: Les documentations et procédures</i> <i>CA-1.0.3: L'analyse et la recherche</i>		DS-2.0 Gestion de l'Administration: <i>DS-2.0.1: Méthodologie d'apprentissage des procédures et documentation</i>
Compétences terminales CT-5.0: Gestion de l'Administration: CT-5.0.1: Capacité à créer, organiser et maintenir des documents administratifs selon les normes établies. CT-5.0.3: Maîtrise des méthodologies de recherche et d'analyse pour la collecte d'informations pertinentes.		Critère d'évaluation : CE-4.0: Gestion de l'Administration: CE-4.0.1: Qualité et rigueur des documents administratifs créés et maintenus. CE-4.0.3: Autonomie, rigueur et pertinence dans la recherche et l'analyse de l'information.
Légende: Rouge = Mot définis dans le glossaire ou directement. Bleu clair = Lien, cliquez dessus pour plus d'information Gras = Important Gras Sous-ligné = Très important		

1. INTRODUCTION:

Cette SAE, consacrée à la conception et au déploiement d'une architecture de routage sécurisé, a représenté un projet d'envergure tant sur le plan technique que personnel, elle est arrivée en plein moment de mon développement personnel en développement sur le langage Python. Ce travail m'a permis de transformer des apprentissages théoriques en compétences opérationnelles, en abordant des problématiques concrètes de cryptographie, de développement logiciel et d'administration système. À travers la réalisation d'une application distribuée implémentant un protocole de routage en couches avec chiffrement RSA (Variante PKCS#1 v1.5) développé « from scratch », j'ai pu consolider mes connaissances en programmation réseau, gestion de bases de données et interfaces graphiques, tout en m'appropriant les bonnes pratiques de documentation et de gestion de projet. Cette fiche individuelle retrace mon parcours d'apprentissage, analyse l'alignement du projet avec les apprentissages critiques et composantes essentielles visés, et dresse un bilan honnête des réussites comme des difficultés rencontrées, notamment dans le cadre d'un travail en autonomie.

2. ANALYSE DES COMPETENCES (AC/CE):

La réalisation de cette architecture de routage m'a permis d'apprendre et de pratiquer énormément de savoir. Voici l'analyse détaillée de ma montée en compétence à travers ce projet.

2.1 Mobilisation des Apprentissages Critiques (AC)

AC23.01 | Automatiser l'administration système avec des scripts

Bien que l'infrastructure ne soit pas administrée par des scripts Shell classiques, l'automatisation a été intégrée au cœur du code Python.

- **Réalisation:** Le fichier « `router.py` » a été conçu pour être lancé en instances multiples via des arguments en ligne de commande (`sys.argv`). Le Master gère automatiquement le nettoyage de la base de données au démarrage (`init_db`) pour garantir un état système sain sans intervention humaine.

Niveau Post-Bac	A.R	Fiche Individuelle SAE 3.02	Documentation	1	2
Informatique	31/12/2025				

AC23.02 | Développer une application à partir d'un cahier des charges donné

Le projet respecte les contraintes strictes imposées par le sujet (pas de librairie de cryptographie externe, utilisation de PyQt, architecture distribuée, liste de fonctionnalité à avoir).

- **Réalisation:** J'ai développé une interface graphique complète pour le Client et le Master en respectant la maquette fonctionnelle attendue. L'application gère les cas d'erreurs spécifiés (champs vides, IP invalides) et implémente la logique métier (choix du nombre de sauts) demandée.

AC23.03 | Utiliser un protocole réseau pour programmer une application client/serveur

C'est le cœur technique du projet. J'ai dû concevoir un protocole applicatif personnalisé au-dessus de la pile TCP/IP.

- **Réalisation:** Utilisation du module socket en mode SOCK_STREAM. J'ai défini un protocole textuel basé sur des séparateurs (|) pour structurer les échanges (ENREGISTREMENT, ROUTEURS, MESSAGE). J'ai géré la concurrence d'accès via le module threading, permettant au Master et aux Routeurs de traiter plusieurs connexions simultanées. (Voir l'analyse protocolaire de la Documentation Technique pour plus d'information sur ces points)

AC23.04 | Installer, administrer un système de gestion de données

Le projet nécessite la persistance de la topologie et des traces d'audit.

- **Réalisation:** Installation et configuration d'un serveur **MariaDB**. Création d'un schéma relationnel adapté (routeurs, logs).

AC23.05 | Accéder à un ensemble de données depuis une application

L'application Python interagit dynamiquement avec la base de données et les fichiers de configuration ainsi que de logs.

- **Réalisation:** Utilisation du connecteur mysql-connector-python. Le Master effectue des opérations CRUD (Plus particulièrement create, read, delete): insertion des nouveaux routeurs, suppression lors de la déconnexion, et lecture lors de la distribution de la topologie aux clients. La configuration de la connexion vers la base de données est fait en intéragissant avec un fichier de configuration par l'intermédiaire de la fonction chargement_conf_bdd().

2.2 Validation des Composantes Essentielles (CE)

- **CE3.01 (Écoute des besoins):** J'ai respecté la contrainte majeure d'implémenter RSA "from scratch" (Algorithme_de_chifffrage.py) plutôt que d'utiliser une solution de facilité comme cryptography ou une semi-implémentation.
- **CE3.02 (Documentation):** Le code est commenté (Docstrings), et une documentation technique structurée (installation, architecture, protocole) a été produite pour faciliter la reprise du projet si besoin.
- **CE3.03 & CE3.04 (Outils numériques et de développement):** J'ai choisi un environnement de développement adapté (VS Code, Git) et des outils de gestion de projet informatique (Trello, Google Sheets, Docs, GitHub).
- **CE3.05 (Sécurité):** La sécurité est le sujet même du projet. J'ai intégré des notions d'anonymisation (Routage en couche), de cryptographie asymétrique et de validation des entrées utilisateurs (prévention des injections SQL basiques et validation des IP dans l'interface).

Niveau Post-Bac	A.R	Fiche Individuelle SAE 3.02	Documentation	1	3
Informatique	31/12/2025				

3. CONCLUSION DE LA SAÉ:

3.1 Bilan du projet:

Ce projet d'architecture multi-distribuée fut un défi technique très stimulant, dépassant le cadre d'une simple application client-serveur pour toucher aux fondamentaux de la cryptographie et du routage.

3.1.1 Les points forts (Les "Plus"):

- **Maîtrise complète de la chaîne:** De la génération des nombres premiers pour RSA jusqu'à l'affichage pixel-perfect de l'interface graphique, chaque composant a été fait à la main et franchement je suis très fière du résultat final.
- **Architecture modulaire:** La séparation claire entre le moteur cryptographique, la logique réseau et l'interface graphique rend le code maintenable et évolutif.
- **Robustesse:** La gestion des signaux (SIGINT) et des threads permet une fermeture propre des applications, évitant les sockets orphelins, un problème classique en programmation réseau.
- **Aspect pédagogique:** L'implémentation manuelle de RSA a permis de démythifier le fonctionnement mathématique du chiffrement asymétrique (J'ai décidé de choisir une implémentation un peu plus complexe de RSA car l'algorithme de base est limitant sur la taille des messages).

3.1.2 Les points faibles (Les "Moins"):

- **Performance de RSA:** L'implémentation en Python pur (bien que fonctionnelle) est nettement plus lente qu'une implémentation native en C (OpenSSL), ce qui se ressent lors de la génération de clés de grande taille (2048 bits+).
- **Gestion de projet:** En toute honnêteté, mon approche à la gestion de projet était originellement bonne, mais j'avais initialement été très passionné par la SAE sur la Cybersécurité « Pentest ». J'ai passé énormément de temps dessus, ce qui m'a fait prendre un énorme retard sur ma gestion de projet. Malgré cela, je pense que j'ai pu réussir l'entièreté du projet en matière technique.

3.2 Le choix de travailler seul:

Le choix de réaliser ce projet en autonomie a été une décision importante pour moi comportant des avantages significatifs, malgré la charge de travail accrue.

- **Cohérence technique:** Travailler seul a garanti une homogénéité parfaite de mon code. Il n'y a pas eu de problèmes d'intégration ("Merge conflicts") ou de disparité de style de codage entre les modules Master, Routeur et Client.
- **Montée en compétence globale:** Cela m'a forcé à intervenir sur **toutes** les couches du projet (Base de données, Réseau, Cryptographie, Interface Graphique). Je ne me suis pas cantonné à une seule spécialité, ce qui m'a permis d'acquérir une vision "Full Stack" de l'application.
- **Gestion du temps:** J'ai pu avancer à mon propre rythme, en itérant rapidement sur les bugs sans dépendre de la disponibilité d'un binôme.

Si c'était à refaire, je pense que je resterais dans mon choix de travaillé seul.

En conclusion, cette SAÉ, m'a permis de consolider mes acquis en programmation réseau tout en explorant des concepts avancés de sécurité et d'architecture logicielle comme la connaissance des patrons de design, validant ainsi ma capacité à concevoir des outils informatiques.