**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

Université Abderrahmane Mira, Bejaïa

Faculté des Sciences Exactes

Département Informatique

En vue de l'obtention du diplôme de master en informatique

Option : Administration et Sécurité des Réseaux


*Mémoire de fin de cycle*

***Thème***

# *Implémentation d'un protocole d'authentification et de partage de clés dans un système distribué. Cas d'étude : Mini-Chat.*

*Réalisé par* :

Mr DJAFRI Driss

Mr DJAFRI Farouk

*Devant le jury composé de* :

**Président:** Mr MEHAOUED
**Examinatrice:** Mme R. BELALTA
**Encadreur:** Mr N. DJEBARI

***Promotion: 2015/2016***

# ***Remerciements***

*Nos vifs remerciements vont d'emblée à Dieu tout puissant qui nous a doté d'une grande volonté et d'un savoir adéquat pour mener à bien ce modeste travail.*

*Nous tenons tout d'abord à remercier Mr DJEBARI Nabil pour l'honneur qu'il nous a fait en acceptant de nous encadrer. Ses conseils précieux nous ont permis une bonne orientation dans la réalisation de ce modeste travail.*

*Nos remerciements sont adressés également à nos chers parents pour tous les sacrifices consentis à notre égard et leur énorme soutien.*

*A tous nos proches amis (e).*

# ***Dédicaces***

*Ce modeste travail est dédié :*

*A nos chers parents qui nous ont soutenus et encouragés durant toute notre scolarité,*

*A nos frères et sœurs,*

*A nos proches,*

*A nos amis(e),*

*A toutes les personnes qui nous ont apportés de l'aide.*

# ***Table de matières***

**Chapitre 2:** ***Protocoles d'authentifications et partage de clés***

**Chapitre 3: *Analyse et conception***

**Chapitre 4:*Implémentation***

# ***Liste des figures***

# ***Liste des tableaux***

# ***Liste des abréviations***

AAA: Authentication, Authorization, Accounting.

AES: Advanced Encryption Standard.

ECC: Elliptic Curve Cryptography.

FAI : Fournisseur d'Accès Internet.

JDBC: JAVA DataBase Connectivity.

JDK: JAVA Devlopement Kit.

KDC: Key Distributing Center.

MySQL: My Structured Query Language.

NAS: Network Access Server.

RADIUS: Remote Authentication Dial-In User Service.

RSA: Rivest, Shamir, Adleman.

TCP/IP: Transmission Control Protocol/Internet Protocol.

TGS: Ticket Garanting Service.

TGT: Ticket Garanting Ticket.

UDP: User Datagramme Protocol.

UML: Unified Modeling Language.

VSA: Vendor Spesific Attrebutes.

2TUP: 2Track Unified Process.

# Introduction générale

L'utilisation croissante des réseaux informatiques et en particulier internet vient du fait que ce dernier offre une diversité de service, mais aussi grâce à sa facilité d'utilisation qui vise à le rendre accessible par toute les catégories de la population.

Ces dernières communiquent entre elles sur ce réseau par échange de données qui circulent en permanence et qui risque d'être exposées à plusieurs types d'attaques, à savoir la lecture clandestine, la modification des données, l'usurpation d'identités, etc…

Pour cela, il est nécessaire de mettre en place un système de sécurité qui assure certains services tels que la confidentialité, l'intégrité des données et l'authentification. L'un des systèmes les plus répondus pour assurer ces services est le système Kerberos. Mais aussi ce système doit être capable de chiffrer les données échangées sur le réseau pour éviter la circulation en claire de données confidentielles telles que les mots de passe. L'un des meilleurs systèmes les plus répondus pour régler ce problème est le système de la cryptographie hybride pour le partage de clés et le chiffrement.

Kerberos est un système qui permet l'authentification des utilisateurs et des services sur un réseau supposé non sûr, Kerberos utilise une tierce partie de confiance envers qui toutes les entités de réseau font confiance. Le système Kerberos est composé de deux serveurs qui communiquent à travers un canal sécurisé et qui partagent une clé symétrique : le serveur **KDC** (**K**ey **D**istribution **C**enter) et le serveur **TGS** (**T**icket **G**ranting **S**ervice) qui assure respectivement le service d'authentification et le contrôle d'accès.

La cryptographie hybride combine les deux systèmes afin de bénéficier de la rapidité de la cryptographie symétrique pour le contenu du message, et utilisation de la cryptographie asymétrique uniquement pour la clé.

L'objectif de ce travail consiste à concevoir et implémenter un prototype du système d'authentification Kerberos et de protocole hybride de partage de clé et de chiffrement (RSA, AES) sous JAVA.

Ce mémoire est constitué de quatre chapitres. Le premier chapitre est consacré aux concepts et terminologies de base de la sécurité et la cryptographie. Le deuxième chapitre se focalise sur quelques  protocoles d'authentifications et de partage de clés existant, le déroulement de ces protocoles, leurs avantages et inconvénients. Dans le troisième chapitre, nous expliquons la solution proposée, et nous décrivons en détail le système choisi et sa conception en utilisant UML. Dans le quatrième chapitre nous décrivons les outils de développement utilisés et l'implémentation de notre application, et enfin nous clôturons le mémoire par une conclusion générale.

# Chapitre 1

## *Généralités sur la sécurité informatique*

### 1.1 Introduction

La sécurité des systèmes informatiques consiste à protéger l'accès et la manipulation des données et des ressources d'un système par des mécanismes d'authentification, d'autorisation, de contrôle d'accès, etc. Cependant, avec l'ouverture des entreprises et des personnes à Internet, l'assurance de la sécurité des systèmes devient très difficile, du fait que, les attaques et les intrusions augmentent de plus en plus et deviennent de plus en plus complexes et difficiles à éviter.

Ce chapitre introductif présente une vue générale sur les systèmes distribués et des notions de base de la sécurité informatique ainsi que des critiques sur quelques systèmes de messagerie instantanée les plus connus.

## 1.2 Système distribué

### 1.2.1 Définition

Un système informatique distribue (reparti)= ensemble d'ordinateurs connectés par un réseau de communication [48].

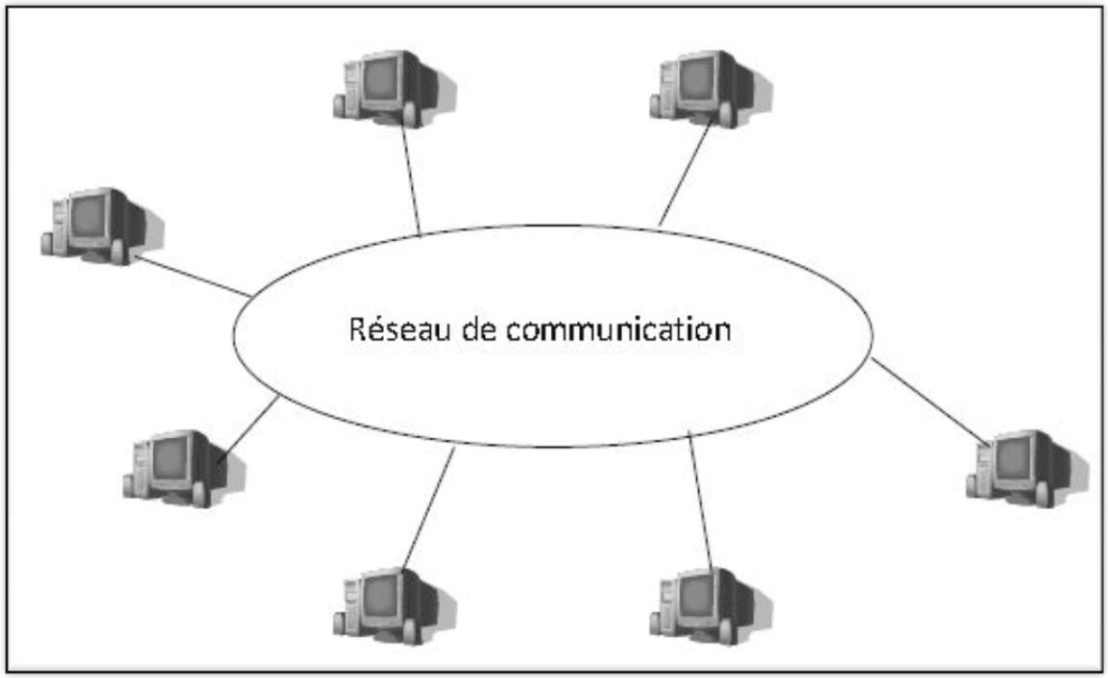


Figure. 1.1 – architecture d'un système distribué.

Exemples de systèmes distribues (SD) [48] :

- Serveur de fichiers : les fichiers se trouvent physiquement sur un serveur, mais virtuellement accessibles à partir de n'importe quelle machine, seulement si le réseau ou le serveur se plante plus d'accès aux fichiers.
- Web : un serveur web auquel se connectent des navigateurs web (clients).
- SD pour calcul scientifiques : plusieurs machines hétérogènes connectées par un réseau local ou internet (grille de calcul). Le principe est le suivant : un ou plusieurs serveurs distribue des calculs aux machines clientes, un client exécute son calcul puis renvoie les résultats au serveur. Avantage : utilisation maximum des ressources de calcul, seulement si le réseau ou le serveur plante le système s'arrête.

### 1.2.2 Caractéristiques des Systèmes distribués

Il n'existe pas de définition rigoureuse de ce qu'un système distribue. On peut essayer de dégager quelques propriétés qui caractérisent les systèmes qui sont universellement considères comme distribués. Un système est considéré comme distribué s'il présente les caractéristiques suivantes [48]:

- Le système est constitué d'un ensemble d'éléments de traitement (processeurs) relies par des organes de communication qui leur permettent d'échanger de l'information.
- Un élément d'un système distribué peut tomber en panne sans affecter nécessairement le fonctionnement de l'ensemble.
- Absence de mémoire commune, ce qui implique l'impossibilité de capter instantanément l'état global du système aux moyens partagés.
- Absence d'horloge physique commune.
- Variabilité des délais de transmission de messages ce qui implique que deux éléments d'un système distribué peuvent avoir des informations différentes sur un troisième élément, et voir des événements s'y produire dans des ordres différents.

### 1.2.3 La sécurité des Systèmes distribués

La nature des SD fait qu'ils sont souvent sujets à des attaques [48] :

- Les communications à travers le réseau peuvent être interceptées.
- On ne connait pas toujours bien un élément distant avec qui on communique.

Des solutions pour ces problèmes :

- Connexion sécurisée par authentification avec les éléments distants.
- Cryptage des messages circulant sur le réseau.

### 1.2.4 Avantage des systèmes distribués

Les avantages des systèmes distribués sont comme suit [48] :

- Partage de données : de multiples utilisateurs peuvent accéder à une base de données partagée.
- Partage de périphériques : de multiples utilisateurs peuvent partager des ressources.

- Communication : facilite la communication interpersonnelle avec le courrier électronique par exemple.
- Rapidité d'exécution : un système distribué peut avoir une puissance de calcul global plus importante qu'un gros ordinateur.
- Fiabilité : le système peut continuer à fonctionner même si une machine tombe en panne.

## 1.3 La sécurité informatique

La sécurité est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité du système contre les menaces. Elle a pour mission principale la protection des informations et des ressources contre toute divulgation ou destruction [7].

L'accès à des ressources doit être également protégé, la sécurité garantie à l'ensemble des machines du réseau un fonctionnement optimal et s'assure que les machines possèdent uniquement les droits qui leur ont été accordés. La notion de sécurité inclut non seulement l'aspect confidentiel, mais également l'authentification.

### 1.3.1 Les aspects de la sécurité

La sécurité participe donc à la sureté d'un système contre des utilisateurs malveillants, par conséquent, tous les mécanismes disponibles doivent être utilisés pour garantir les propriétés suivantes [29] [19] :

- Confidentialité : mécanisme pour transmettre des données de telle sorte que seul le destinataire autorisé puisse les lire.
- Intégrité : mécanisme pour s'assurer que les données reçues n'ont pas été modifiées durant la transmission.
- Non-répudiation : mécanisme pour enregistrer un acte ou un engagement d'une personne ou d'une entité de telle sorte que celle-ci ne puisse pas nier avoir accompli cet acte ou pris cet engagement.
- Authentification :
  - *D'une information :* prouver qu'une information provient de la source annoncée (auteur, émetteur).

- *D'une personne* (ou groupe ou organisation) : prouver que l'identité est bien celle annoncée.

## 1.4 La cryptographie

Depuis sa création, le réseau internet a tellement évolué qu'il est devenu un outil essentiel de communication. Cependant, cette communication met de plus en plus en jeu des problèmes stratégiques liés à l'activité des entreprises sur internet, les transactions faites à travers le réseau ne sont pas sécurisées, car elles peuvent être interceptées, et afin d'éviter ces problèmes, la cryptographie a été mise en place.

La cryptologie est une technique très ancienne. Ainsi, Jules César utilisait déjà un algorithme que nous appelons aujourd'hui le chiffrement par substitution, qui consiste à décaler d'une valeur constante les lettres dans l'ordre alphabétique. Mais la cryptologie est aussi une science qui se renouvelle. Depuis les années 1970, elle est devenue un thème de recherche scientifique académique [11].

### 1.4.1 Définition et terminologie de la cryptologie

La cryptologie (science du secret), est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse [32].

La cryptographie : c'est un ensemble de méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on le transforme pour le rendre incompréhensible, c'est ce qu'on appelle le chiffrement. Inversement, le déchiffrement est l'action qui permet de reconstruire le message en claire à partir du message chiffré. Dans la cryptographie moderne, les transformations en question sont des fonctions mathématiques, appelées algorithmes cryptographiques, qui dépondent d'un paramètre appelé « clé ».

La cryptanalyse : c'est l'analyse des procédés cryptographiques, elle sert à étudier les faiblesses d'un système cryptographique en décryptant les messages sans connaitre la clé de déchiffrement.

### 1.4.2 Techniques de la cryptographie

Il existe deux grandes techniques de cryptographie : la cryptographie symétrique et la cryptographie asymétrique [13].

### 1.4.2.1 La cryptographie symétrique

Le cryptage symétrique est la technique la plus ancienne et la plus connue. Une clé secrète, est appliquée au texte d'un message pour modifier le contenu d'une certaine manière. Cela pourrait être aussi simple que de décaler chaque lettre d'un certain nombre d'emplacements dans l'alphabet. Tant que l'expéditeur et le destinataire connaissent la clé secrète, ils peuvent crypter et décrypter tous les messages qui utilisent cette clé.

Le problème de ce chiffrement est qu'il faut trouver un moyen de transmettre la clé unique entre les deux interlocuteurs [13].


Figure. 1.2 – Chiffrement et déchiffrement à clé symétrique.

Il existe deux modes de chiffrement dans la cryptographie symétrique qui sont : le chiffrement classique et le chiffrement moderne, pour ce dernier on trouve deux grandes catégories qui sont [13]:

- Le système chiffrement par bloc.

- Le système de chiffrement par flot.

#### 1.4.2.1.1 Le système chiffrement par bloc (Bloc cyphers)

L'algorithme d'un chiffrement par bloc est généralement basé sur un modèle itératif. Il utilise une fonction 'F' qui prend une clé secrète 'k' et un message 'M' de 'n' bits. La fonction 'F' est itérée un certain nombre de fois (nombre de tours). Lors de chaque tour, la clé 'K' est différente et on chiffre le message qui vient d'être obtenu de l'itération précédente. Les différentes clés '$K_i$' qui sont utilisées sont déduites de la clé secrète 'K'.

Comme exemple d'algorithmes de chiffrement par bloc [20]:

- **DES** (**D**ata **E**ncryption **S**tandard).
- **3DES** (Triple **DES**).
- **AES** (**A**dvanced **E**ncryption **S**tandard).

#### 1.4.2.1.2Système de chiffrement par flot (de flux ou par stream)

L'algorithme de chiffrement par flot, est un système qui chiffre les données bit par bit quel que soit la longueur du message codé. Ces algorithmes peuvent être assimilés à des algorithmes de chiffrement par blocs, où le bloc a une dimension relativement petite (1 bit). Le principal avantage de cette méthode consiste en sa rapidité de transmission de données.

Comme exemple d'algorithmes de chiffrement par flot [28] [17]:

- **RC4** (utilisé notamment par le protocole WEP du Wi-Fi).
- **E0** (utilisé par le protocole Bluetooth).

La cryptographie symétrique comporte un avantage majeur qui consiste en sa rapidité, car elle ne fait que réarranger les bits d'un message, ce mécanisme est particulièrement adapté à la transmission de grandes quantités de données, grâce à une implémentation matérielle de ces algorithmes les plus connus sous forme de micro-processeurs. Cependant, elle impose d'avoir un canal sécurisé pour l'échange de clés, ce qui cause le problème de distribution des clés, ainsi que l'utilisateur doit avoir autant de clés privées qu'il a d'interlocuteurs.

### 1.4.2.2 La cryptographie asymétrique

La cryptographie asymétrique repose sur un autre concept faisant intervenir une paire de clés, une pour le chiffrement et l'autre pour le déchiffrement. L'une de ces clés est désignée sous le terme de « clé privée », elle n'est jamais transmise à personne. L'autre sous le terme « clé publique » elle est en revanche diffusée publiquement sans problème. Ces clés sont générées en même temps.

L'émetteur chiffre le message à envoyer avec la clé publique du récepteur, et ce dernier le déchiffre avec sa clé privée [8].

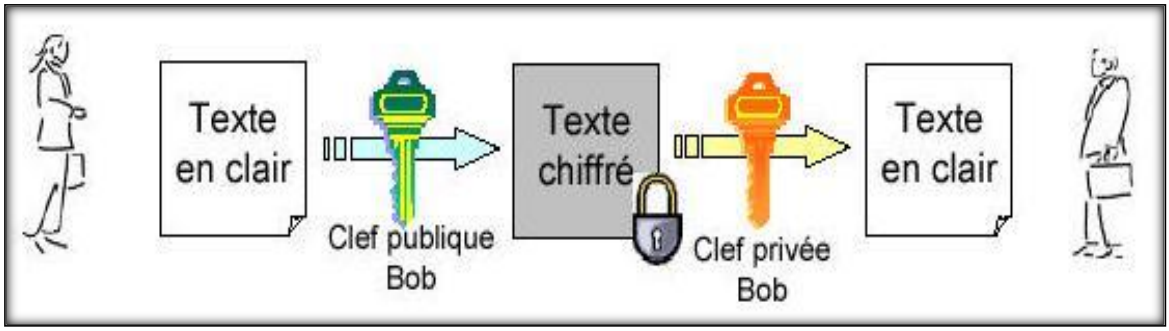


Figure. 1.3 – Chiffrement et déchiffrement à clé asymétrique.

Comme exemple d'algorithmes de chiffrement asymétrique on cite :

- **RSA** (Ron **R**ivset, Adi **S**hamir et Léonard **A**dleman)
- **ElGamal** (Taher Elgamal).

#### 1.4.2.2.1 Signature numérique

Le paradigme de signature numérique correspond à une marque personnelle mise sur un document électronique, qui permet l'authentification, au-terme dit prouver qu'un message provient bien d'un émetteur donné. Ce mécanisme permet aussi de vérifier l'intégrité du message reçu, qui est une preuve que le document n'a pas subi d'altération entre l'instant où il a été signé par son auteur et celui où il a été consulté, ainsi il assure la non-répudiation, afin d'éviter que l'expéditeur nie le fait d'voir chiffré le message avec sa clé privée. Cette méthode consiste à chiffrer le message à envoyer avec la clé privée de l'émetteur. Si ce dernier arrive à le déchiffrer à l'aide de la clé publique de l'émetteur cela veut dire que le message provient bien de lui [21].

#### 1.4.2.2.2 Fonction de hachage

Le principe de cette fonction est qu'un message clair de longueur quelconque doit être transformé en un message de longueur fixe inférieure à celle de départ. Le message réduit porte le nom « haché », « «résumé » ou « condensé ». L'intérêt est d'utilisé ce condensé comme empreinte digitale du message original afin que ce dernier soit identifié de manière univoque. La fonction de hachage doit être telle qu'elle associe un et un seul haché à un texte en clair (cela signifie que la moindre modification du document entraîne la modification de son haché) [25].

## 1.5 Comparaison entre les deux cryptosystèmes existant

En résumé, une synthèse de ces deux méthodes de cryptographie est décrite dans le tableau suivant.

| Type de crypto-système | Avantages | Inconvénients |
|---|---|---|
| Système Symétrique | - Rapide.<br>- Assure<br>la confidentialité. | - La non-répudiation n'est pas assurée.<br>- Il faut autant de paires de clés que des couples de correspondant.<br>- difficulté de distribuer les clés. |
| Système asymétrique | - Utilise deux clés différentes.<br>- Plus fiable et plus sûr.<br>- Assure la confidentialité.<br>- Assure la non-répudiation. | - Lent et demande beaucoup de calcul. |

Tableau. 1.1 – Comparaison des deux systèmes de cryptage.

## 1.6 Système de messagerie instantanée

Un système de messagerie instantanée permet d'échanger des messages textuels ou des fichiers (images, vidéo, son…) en temps réel entre plusieurs utilisateurs connectés à un même réseau. En anglais, on parle de « chat », qui se prononce « Tchat » en français [23].

Malgré que ces systèmes de messagerie aient beaucoup d'avantages, mais les utilisateurs de ces messageries, sont-ils vraiment en sécurité ?

Dans ce qui suit nous allons citer quelques inconvénients de ces messageries en prenant comme exemple *VIBER* et *SKYPE*.

### 1.6.1 Viber

Viber est un logiciel propriétaire pour smartphones, PC et Mac qui permet de passer des appels téléphoniques en utilisant la technique de voix sur IP. Il est développé par la société israélienne Viber Media Inc, rachetée en février 2014 par le groupe japonais Rakuten [14].

#### 1.6.1.1 Collecte et utilisation des informations par Viber

La collection et l'utilisation des informations par Viber sont définies comme suit [14] :

- La copie de votre carnet d'adresses (noms et téléphones) est conservée dans une base de données active.
- Viber conserve également un enregistrement détaillé des appels pour chaque message et appel passant par le système. Nous les conservons 30 mois maximum.
- Si vous décidez de partager votre emplacement avec un certain utilisateur ou groupe, votre localisation sera extraite de votre appareil et envoyée à Viber.

#### 1.6.1.2 Partage et divulgation d'informations

Il peut être amené à divulguer vos Informations personnelles s'il pense que c'est nécessaire pour [14]:

- Obéir à la loi ou appliquer une procédure juridique à laquelle nous devons nous soumettre.
- Protéger et défendre ces droits ou ça propriété (notamment pour appliquer nos contrats).
- Agir dans l'urgence pour protéger des utilisateurs de service Viber ou des membres du public.

Pour pouvoir être en mesure de fournir les produits Viber demandés, Viber pourra parfois être amené à partager vos Informations personnelles et vos données de trafic avec des fournisseurs de service partenaires et/ou agents de confiance tels que : établissements bancaires ou autres fournisseurs de paiement et de services analytiques, support client ou services d'hébergement.

#### 1.6.1.3 La sécurité des serveurs Viber

Les informations personnelles sont conservées sur des serveurs et protégées par des réseaux sécurisés accessibles uniquement de quelques employés autorisés. Toutefois, aucune méthode de transmission par Internet ou méthode de stockage électronique n'est sûre à 100 % [14].

### 1.6.2 Skype

Si vous pensez que les messages privés que vous envoyez sur Skype sont protégés par un chiffrement de bout en bout, réfléchir à nouveau.

Le service appartenant à Microsoft scanne régulièrement le contenu des messages pour des signes de fraude, et les gestionnaires de l'entreprise peuvent enregistrer les résultats indéfiniment.

Avec l'aide d'un chercheur de confidentialité et de sécurité indépendante, Ars(1) utilisé Skype pour envoyer quatre liens Web qui ont été créés uniquement à des fins de cet article. Deux d'entre eux ne sont jamais cliqués, mais les deux autres, celui qui commence en lien HTTP et l'autre HTTPS sont accessibles par une machine à 65.52.100.214, une adresse IP appartenant à Microsoft **[**15].

## 1.7 Conclusion

Au cours de ce chapitre, nous avons défini les systèmes distribués et une vue générale sur la sécurité informatique ainsi que quelques systèmes de messagerie instantanée les plus connus en les critiquant sur leur différentes failles et vulnérabilités.

Dans le chapitre suivant, nous allons citer quelques protocoles d'authentification et de partages de clés les plus connus afin de choisir ceux qui nous conviennent.

(1)Ars Technica-le nom est latin dérivé du «art de la technologie», nous spécialisons dans les nouvelles et commentaires, l'analyse des tendances technologiques, et des conseils d'experts sur des sujets allant des aspects les plus fondamentaux de la technologie pour la technologie de nombreuses façons est d'aider nous découvrons notre monde.

# Chapitre 2

## *Protocoles d'authentifications et partage de clés*

### 2.1 Introduction

Lorsque les réseaux ont commencé à adopter le modèle de communication client-serveur et que les terminaux ont été remplacés par les ordinateurs personnels, les administrateurs ne pouvaient pas avoir confiance aux utilisateurs finaux, car les informations échangées au cours des communications ne sont pas sécurisées et quiconque peut les intercepter, pour cela il a fallu mettre en place un système de partage de clés afin d'assurer un partage de données chiffré et un système d'authentification permettant de rétablir cette confiance dans le réseau. Ce dernier permet à chacun des correspondants de s'assurer que son partenaire et bien celui qu'il prétend être.

Dans ce chapitre, nous présentons et détaillons quelques protocoles d'authentification et de partage de clé qui constituera par la suite la brique de base de notre implémentation.

## 2.2 Authentification

L'Authentification est la vérification d'informations relatives à une personne ou à un processus informatique. L'authentification complète le processus d'identification dans le sens où l'authentification permet de prouver une identité déclarée. Dans un serveur, un processus de contrôle valide l'identité et après authentification, donne l'accès aux données, applications, bases de données, fichiers ou sites Internet. Dans le cas contraire, l'accès est refusé [44].

## 2.3 Le protocole d'authentification RADIUS

Le protocole RADIUS (*Remote Authentication Dial-In User Service*), mis au point initialement par Livingston, est un protocole d'authentification standard, défini par un certain nombre de RFC.

Le fonctionnement de RADIUS est basé sur un système client/serveur chargé de définir les accès d'utilisateurs distants à un réseau. Il s'agit du protocole de prédilection des fournisseurs d'accès à internet car il est relativement standard et propose des fonctionnalités de comptabilité permettant aux FAI (Fournisseur d'Accès Internet) de facturer précisément leurs clients.

Le protocole RADIUS repose principalement sur un serveur (le serveur RADIUS), relié à une base d'identification et un client RADIUS, appelé NAS (*Network Access Server*), faisant office d'intermédiaire entre l'utilisateur final et le serveur. L'ensemble des transactions entre le client RADIUS et le serveur RADIUS est chiffrée et authentifiée grâce à un secret partagé [34].

### 2.3.1 Les différents acteurs de RADIUS

Les acteurs du RADIUS sont [47]:

- **L'utilisateur :** émetteur de la requête d'authentification (poste de travail, un portable...).
- **Client RADIUS :** le point d'accès au réseau (NAS, firewall, point d'accès Wireless, etc...).
- **Serveur RADIUS :** relié à une base d'authentification (Base de données).

### 2.3.2 Principe du protocole RADIUS

Le protocole d'authentification RADIUS se déroule comme suit [2] :

1. Un utilisateur envoie une requête au NAS (Network Acces Server) afin d'autoriser une connexion à distance.
2. Le NAS achemine la demande au serveur RADIUS.
3. Le serveur RADIUS consulte sa base de données d'identification afin de connaitre le type de scénario d'identification demandé pour l'utilisateur.
4. Soit le scénario actuel convient, soit une autre méthode d'identification est demandée à l'utilisateur. Le serveur RADIUS retourne ainsi une des quatre réponses suivantes :
    i. *ACCEPT* : L'identification a réussi.
    ii. *REJECT* : L'identification a échoué.
    iii. *CHALLENGE* : Le serveur RADIUS souhaite des informations supplémentaires de la part de l'utilisateur et propose un « défi ».
    - Une autre réponse est possible : *CHANGE PASSWORD* où le serveur RADIUS demande à l'utilisateur un nouveau mot de passe.
    iv. *CHANGE PASSWORD* est un attribut VSA (Vendor Specific Attributes), c'est-à-dire qu'il est spécifique à un fournisseur.
5. Suite à cette phase dite d'authentification, débute une phase d'autorisation où le serveur retourne les autorisations de l'utilisateur.

La figure suivante explique brièvement le fonctionnement du protocole RADIUS.

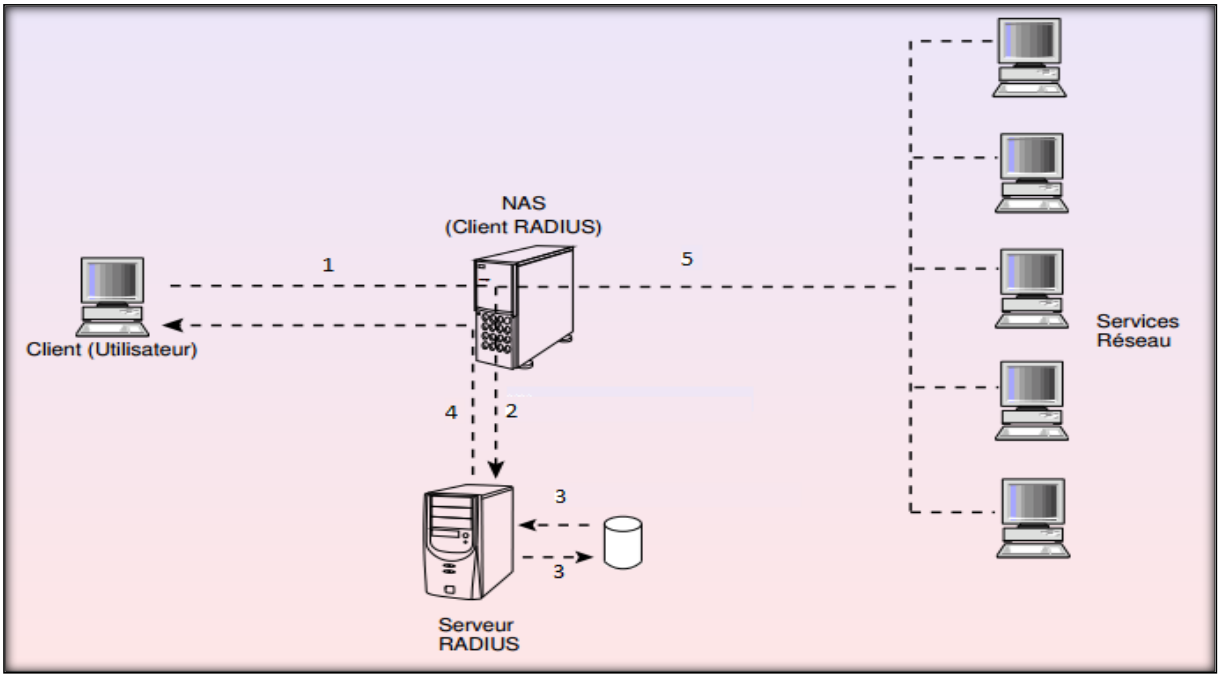


Figure. 2.1 – Fonctionnement du protocole RADIUS.

### 2.3.3 Avantages

Les avantages du protocole RADIUS sont comme suit [2]:

- Fonctionnement basé sur un système client/serveur chargé de définir les accès d'utilisateurs distants à un réseau.
- Le protocole RADIUS permet de faire la liaison entre des besoins d'identification et une base d'utilisateurs en assurant le transport des données d'authentification de façon normalisée.
- L'ensemble des transactions entre le client RADIUS et le serveur RADIUS est chiffré.
- RADIUS permet le respect des trois A : Authentication, Authorization et Accounting (AAA) ou Authentification, autorisation et comptabilisation.
- L'utilisation du protocole UDP qui simplifie la mise en œuvre du serveur.

### 2.3.4 Inconvénients

Les limitations du protocole RADIUS sont comme suit [2]:

- RADIUS a été conçu pour des identifications par modem, sur des liaisons lentes et peu sûres, c'est la raison du choix du protocole UDP.
- Il base son identification sur le seul principe du couple (nom, mot de passe).

- Il assure un transport en claire, seul le mot de passe est chiffré par hachage.
- Il est strictement client-serveur.
- Il n'assure pas des mécanismes d'identification du serveur.

## 2.4 Le protocole d'authentification Kerberos

Kerberos a été conçu dans le but de proposer un protocole d'authentification multi - plate-formes, disposant d'un système de demande d'identification unique, et permettant de contacter ensuite autant de services que souhaité. Il s'agit d'un protocole sécurisé, dans le sens où il ne transmet jamais de mot de passe en clair sur le réseau. Il transmet des messages cryptés à durée de vie limitée. Le terme « *single sign - on* », utilisé en sous - titre de ce document, décrit le fait que l'utilisateur final n'a besoin de s'authentifier qu'une fois pour utiliser toutes les ressources du réseau supportant Kerberos au cours de sa journée de travail (en réalité, au cours du temps de session spécifié par l'administrateur : environ vingt heures, en général). Le système Kerberos repose sur un « *tiers de confiance* » (Trusted third party), dans le sens où il s'appuie sur un serveur d'authentification centralisé dans lequel tous les systèmes du réseau ont confiance [1].

### 2.4.1 Les différents acteurs de Kerberos

Dans le système Kerberos, les acteurs principaux qui interviennent dans son architecture sont les suivants [46] :

- **Client :** Entité pouvant obtenir un ticket auprès du KDC pour utiliser les différents services et ressources d'un réseau.
- **KDC (Key Distributing Center) :** C'es**t** le serveur qui assure l'authenticité des utilisateurs et des services. Il délivre la clé de session pour l'accès au serveur TGS sous forme d'un ticket TGT (Ticket Granting Ticket).
- **TGT :** C'est un ticket délivré par le serveur KDC lors de la première authentification d'un client donné. Sa durée de vie est relativement longue, pour que le client puisse demander accès à plusieurs services différents sans avoir à fournir à plusieurs reprises ses paramètres d'authentification.
- **TGS (Ticket Granting Service) :** C'est le serveur qui assure le contrôle d'accès des utilisateurs. Il délivre la clé de session pour l'accès à un serveur applicatif sous forme d'un ticket.

- **Ticket d'accès :** C'est un ticket délivré à un client donné par le serveur TGS ce qui signifie que le client possède le droit d'accès au service demandé.
- **Clé de session :** C'est une clé symétrique temporaire générée entre un client et un service.

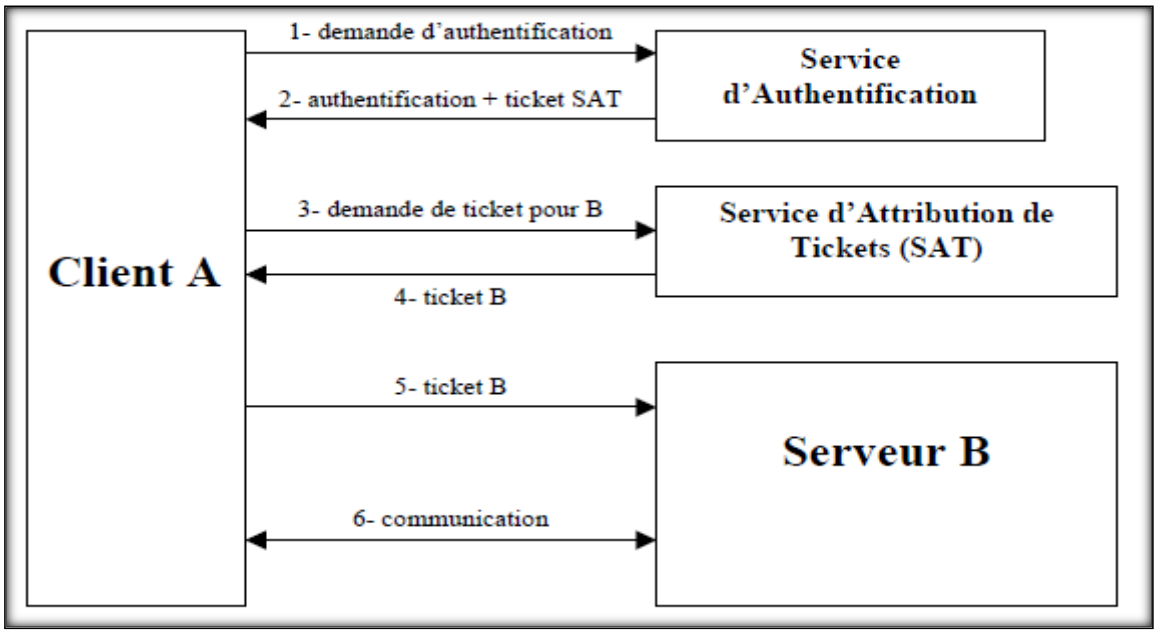


Figure. 2.2 – Processus d'authentification avec Kerberos.

### 2.4.2 Déroulement du protocole Kerberos

Le déroulement du protocole d'authentification se fait suivant les étapes ci-dessous :

**Etape 1 : L'authentification du client**

Le client A envoie une requête contenant son identifiant et sa clé secrète $K_A$(résultat d'une fonction de hachage de son mot de passe) au serveur KDC. Après la vérification de l'identité du client, le serveur lui répond avec une clé de session $K_S$, et un ticket TGT qui comporte la clé de session et l'identifient du client chiffrés par la clé $K_{TGS}$ destiné au serveur TGS. Ces éléments ($K_S$, TGT) sont groupés et chiffrés avec la clé secrète du client, ce qui fait que seul le client peut le déchiffrer.

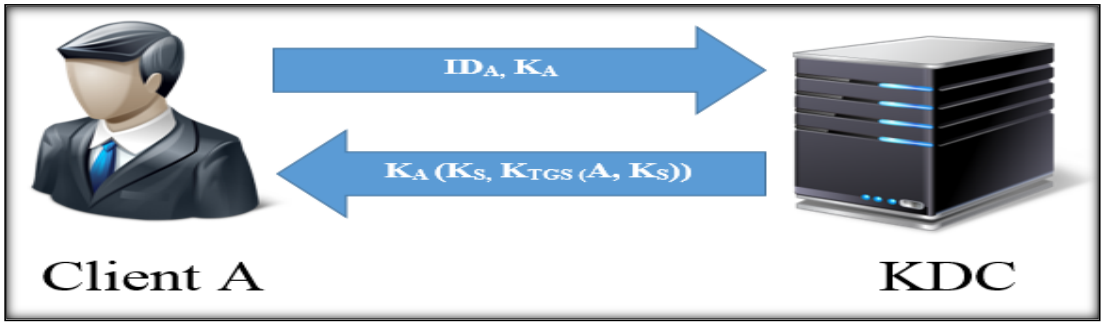


Figure. 2.3 – L'authentification du client.

**Etape 2 : Obtention du ticket d'accès au serveur**

Le client A déchiffre une partie du ticket avec sa clé $K_A$ pour récupérer la clé de session $K_S$. Ensuite, il envoie une requête au TGS en lui demandant le ticket qu'il doit utiliser pour contacter le service du serveur applicatif B. Cette requête comporte le ticket TGT ($K_{TGS}$ (A, $K_S$)), l'identifiant du serveur, ainsi la date d'émission chiffré avec $K_S$. Le serveur TGS déchiffre le ticket TGT avec sa propre clé $K_{TGS}$, récupère la clé de session $K_S$ pour déchiffrer la date et il vérifie si cette date est voisine à sa date courante pour empêcher l'intrus de rejouer l'envoi de la requête. Le serveur TGS vérifie également les droits d'accès. Si le client les possède, le serveur lui délivre le ticket d'accès qui contient deux parties. La première partie se compose de l'identifiant du client A et une clé de session entre le serveur demandé et le client ($K_{AB}$), chiffrés avec la clé du serveur B ($K_B$(A, $K_{AB}$)). La seconde partie se compose de l'identifiant du serveur B ainsi la clé $K_{AB}$, le tout est chiffré avec la clé $K_S$ ($K_S$ (B, $K_{AB}$)).

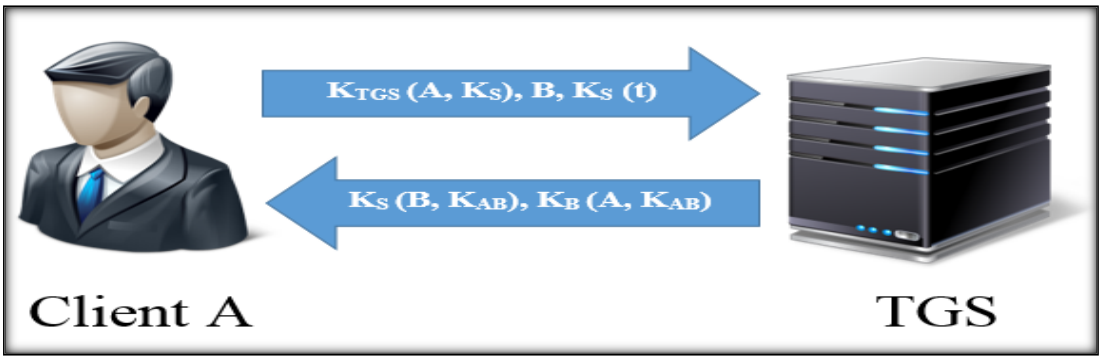


Figure. 2.4 – Obtention du ticket d'accès au serveur.

**Etape 3 : L'accès au service**

Le client reçoit le ticket qui contient la clé de session $K_{AB}$ à partager avec le serveur applicatif. Ensuite, il envoie à ce dernier une requête qui contient son identifiant et la clé $K_{AB}$ chif-

frée avec la clé de serveur $K_B$. Cet échange est lui-même horodaté. Le serveur déchiffre le ticket, récupère la clé $K_{AB}$, vérifie si le ticket est toujours valide, et que la date d'émission de l'utilisateur est voisine à sa date courante, ensuite le serveur lance le protocole pour exécuter le service demandé. Le client peut communiquer avec le serveur applicatif sous le couvert de $K_{AB}$. Si plus tard le client décide de communiquer avec un autre serveur C, il suffira de répéter le message destiné au TGS en remplaçant B par C, ce qui implique que le client peut accéder à tous les serveurs du réseau de façon sécurisée sans que son mot de passe n'ait jamais besoin de circuler sur ce réseau.

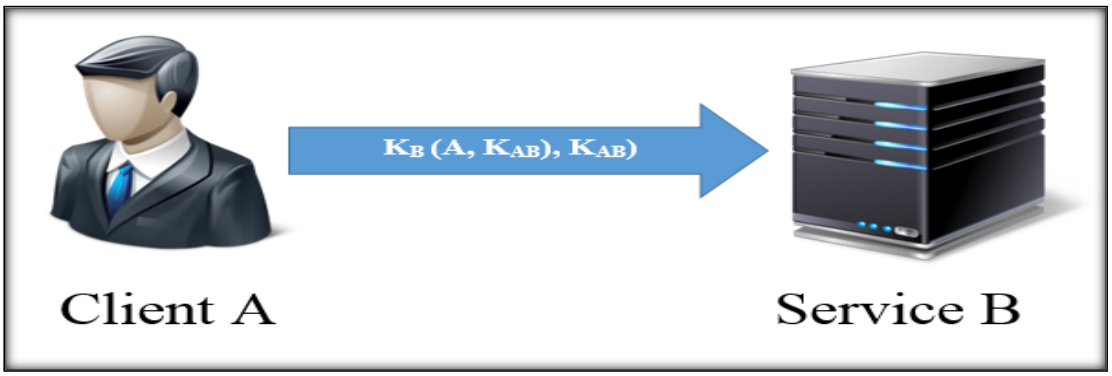


Figure. 2.5 – L'authentification du client.

### 2.4.3 Avantages

Les avantages de Kerberos sont comme suit [37] [26] [5] :

- Très largement déployé.
- Préinstallé sur beaucoup de systèmes d'exploitation.
- Bien intégré au système.
- Kerberos permet en outre de réaliser une authentification mutuelle.
- Il n'y a pas de transmission de mot de passe du service sur le réseau.
- Il offre le mécanisme SSO (Single Signe On), l'utilisateur n'a ainsi qu'un seul mot de passe à se souvenir et ne l'entre typiquement qu'une seule fois par jour.

### 2.4.4 Inconvénients

Les inconvénients de Kerberos sont comme suit [22] :

- Il faut que toutes les machines du réseau soient synchronisées, sinon l'utilisation des timesstamp et la durée de vie des tickets risquent d'être faussées et plus aucune authentification ne sera possible.
- Si l'AS (Authentication Service) de Kerberos est compromis, un attaquant pourra accéder à tous les services avec un unique login.
- Kerberos chiffre uniquement la phase d'authentification, il ne chiffre pas les données qui seront transmises lors de la session.

## 2.5 Partage de clés

Le partage de clé secrète est un algorithme de cryptographie. C'est une forme de partage de secret entre les utilisateurs. Il existe deux modes d'établissement de clé secrète. Distribution de clés où un serveur choisit une clé et le fait parvenir à tout le monde, échange de clés où tout le monde participe à la négociation de la clé.

### 2.5.1 L'algorithme RSA

En 1977, trois mathématiciens, Ronald Rivest, Adi Shamir et Leonard Adleman, ont mis en point une méthode de cryptage à clé publique. La méthode RSA (dont le nom est un acronyme formé des initiales de ses inventeurs) permet à chacun de coder un message à partir d'une clé publique mais n'autorise pas le décodage, qui est conditionné à la connaissance d'une clé privée [42].

#### 2.5.1.1 Le principe de RSA

L'algorithme RSA est basé sur le fait qu'il est beaucoup plus facile de générer deux grands nombres premiers que les retrouver en factorisant leur produit. Le renouvellement des clés n'intervient que si la clé privée est compromise, ou par précaution au bout d'un certain temps (qui peut se compter en années). Les clés se calculent de la manière suivante [36] [45] :

1. Choisir $p$ et $q$, deux nombres premiers distincts.
2. Calculer leur produit $n = p*q$, appelé module de chiffrement.
3. Calculer $\varphi(n) = (p - 1)*(q - 1)$ (c'est la valeur de l'indicatrice d'Euler en $n$).

4. Choisir un entier naturel *e* premier avec *φ(n)* et strictement inférieur à *φ(n)*, appelé exposant de chiffrement.

5. Calculer l'entier naturel *d*, inverse de *e* modulo *φ(n)*, et strictement inférieur à *φ(n),* appelé exposant de déchiffrement.

Comme *e* est premier avec *φ(n)*, d'après le théorème de Bachet-Bézout il existe deux entiers *d* et *k* tels que $e*d + k*\varphi(n)=1$, c'est-à-dire que :

$e*d = 1 \pmod{\varphi(n)}$ : e est bien inversible modulo *φ(n)*.

Le couple (*n, e*) est la clé publique du chiffrement, alors que le nombre *d* est sa clé privée, sachant que l'opération de déchiffrement ne demande que la clef privée *d* et l'entier *n*.

Le chiffrement d'un message M en un message codé C se fait suivant la transformation suivante :

$$C = M^e \bmod n.$$

Pour déchiffrer message codé C, on utilise *d* :

$$M = C^d \bmod n.$$

**2.5.1.2Avantages**

Les avantages de RSA sont comme suit [3] :

- Très rapide, très simple cryptage et de vérification.
- Plus facile à mettre en œuvre que l'algorithme courbe elliptique (ECC).
- Plus facile à comprendre.
- La signature et le décryptage sont similaires; chiffrement et de vérification sont similaires.
- Largement déployé, un meilleur soutien de l'industrie.

**2.5.1.3 Inconvénients**

Les inconvénients de RSA sont comme suit [3] :

- Génération de clés très lente.

- Signature lente et le décryptage, qui sont peu délicate à mettre en œuvre en toute sécurité.

### 2.5.2 L'algorithme Elliptic Curve Cryptography (ECC)

L'algorithme Elliptic Curve Cryptography (ECC) est développé en 1985 par, entre autres, IBM comme alternative au RSA. ECC utilise des courbes elliptiques (grands nombre à déchiffrer en nombres premiers), ainsi les clés sont créées pour l'encodage des données.

#### 2.5.2.1 Principe d'algorithme Elliptic Curve Cryptography(ECC)

Il s'agit d'un échange de clés à la manière de *Diffie Hellman,* c'est-à-dire :

ALICE et BOB se mettent d'accord (publiquement) sur une courbe elliptique *E (a, b,p),* c'est-à-dire qu'ils choisissent une courbe elliptique $y^2 = x^3+ax+b \ mod \ P$. Ils se mettent aussi d'accord (toujours publiquement) sur un point *P* situé sur la courbe.

On définit $n*P=P+P+...+P$ (n fois) où l'opération + correspond à la somme de deux points définie par le symétrique du troisième point d'intersection de la droite définie par les deux points originaux avec la courbe elliptique. Dans le cas où les deux points à ajouter sont identiques, on considère que la droite qui les joint est la tangente à la courbe elliptique passant par l'un d'entre eux. Un tel point est *rationnel*.

Secrètement, ALICE choisit un entier ($d_A$), et BOB un entier ($d_B$). ALICE envoie à BOB le point($d_A*P$), et BOB envoie à ALICE ($d_B*P$).

Chacun de leur côté, ils sont capables de calculer [$d_A(d_B*P) = d_B(d_A*P) = d_A*d_B*P$]qui est un point de la courbe, et constitue leur clef secrète commune.

Si CHARLY a espionné leurs échanges, elle connaît $E(a,b,p),P,d_A*P,d_B*P$. Pour pouvoir calculer $d_A*d_B*P$, il faut pouvoir calculer $d_A$ connaissant *P* et $d_A*P$. C'est ce que l'on appelle résoudre le logarithme discret sur une courbe elliptique. Or, actuellement, si les nombres sont suffisamment grands, on ne connaît pas de méthode efficace pour résoudre ce problème en un temps raisonnable [39].

#### 2.5.2.2 Avantages

Les avantages d'Elliptic Curve Cryptography sont comme suit [39] :

- petites clés, cryptogrammes et signatures.
- génération de clés très rapide.
- signatures rapides.
- chiffrement et déchiffrement sont rapides

#### 2.5.2.3 Inconvénients

Les inconvénients d'Elliptic Curve Cryptography sont comme suit [39] :

- Compliqué et difficile à mettre en œuvre en toute sécurité, en particulier les courbes standard
- algorithmes plus récents pourraient théoriquement avoir des faiblesses inconnues.
- la technologie de cryptographie par courbe elliptique a fait l'objet du dépôt de nombreux brevets à travers le monde. Cela peut rendre son utilisation très coûteuse.

## 2.6 Conclusion

Dans ce chapitre nous avons mentionné quelques protocoles d'authentifications ainsi que quelques protocoles de partage de clés, et nous avons cité leurs avantages et leurs inconvénients, afin de pouvoir choisir les meilleurs protocoles qui conviennent à notre projet.

# Chapitre 3

## *Analyse et conception*

### 3.1 Introduction

Le choix d'un bon protocole d'authentification et de partage de clé et d'un bon cryptosystèm joue un rôle important dans la sécurisation d'un réseau informatique. En effet il indique le niveau de sécurité du ce réseau à chaque action d'un utilisateur, pour cela nous avons mis en place un système d'authentification et un protocole de partage de clé pour protéger les utilisateurs du réseau.

Dans ce chapitre, nous allons détailler le protocole d'authentification et de partage de clé choisi.

## 3.2 Etude préliminaire

Dans la phase de l'étude préliminaire, il convient d'établir les bases permettant de définir les prochaines étapes de la réalisation d'une idée de projet.

### 3.2.1 Présentation du projet

Les approches traditionnelles de transfert de fichiers via e-mail ou ftp posent de nombreux problèmes de sécurité et d'intégrité. Les entreprises recherchent des solutions intégrées, simple à administrer et qui permettent aux utilisateurs de maintenir leur activité et leur productivité en toute sécurité tout en conservant une visibilité et un contrôle sur les échanges effectués.

- **Simplicité d'utilisation, aucune connaissance technique requise**

L'interface Mini-Chat est conçue pour faciliter l'échange de messages et de fichiers quel que soit le niveau informatique. L'utilisation d'icônes et de codes couleurs permettent un confort aux utilisateurs qui prennent en main l'application dès la première utilisation.

- **Gain de temps pour les transferts de fichiers volumineux**

La simplicité d'utilisation du service évite les pertes de temps liées aux soucis techniques fréquemment rencontrés lorsque des solutions inadaptées sont utilisées pour les transferts volumineux.

- Une plate-forme sécurisée pour transférer des fichiers de tout type et de toute taille via une interface utilisateur simple et complète.
- Support listes de contacts
- Communication sécurisée avec capacité de chiffrement intégrée
- Une architecture flexible et extensible, de quelques dizaines à plusieurs centaines d'utilisateurs.
- Architecture technique robuste et évolutive en fonction du nombre d'utilisateurs
- Intégration avec solution d'authentification forte.

Quelques caractéristiques de l'application :

- Le serveur Mini-Chat doit permettre à plusieurs utilisateurs de se connecter et de discuter entre eux à la manière des multiples logiciels de chat disponibles sur Internet.
- Une fois connecté, l'utilisateur peut se déconnecter à tout moment. Au cours d'une session, un utilisateur peut envoyer des fichiers, photos... qui seront enregistrés sur le disque dur, ce qui évite la propagation des virus une fois les clés USB s'échange entre les utilisateurs.
- Le principe de fonctionnement est le suivant : un utilisateur *X* envoie un message ou une pièce jointe à un autre utilisateur sélectionné. Dans le cas où personne n'est sélectionné sur la liste des amis en ligne, le message sera diffusé pour toute personne connectée.
- Ce mini chat comportera une porte d'accès restreinte réservée aux personnes privilégiées et leurs permettra une communication privée et sécurisée.

## 3.3 Problématique

L'utilisation de ce mini chat dans un système distribué signifie qu'il sera à la disposition de plusieurs utilisateurs simultanément, de ce fait nous serons menés à faire face aux problèmes suivants :

- L'authentification des utilisateurs.
- La confidentialité des communications.
- L'intégrité des messages.

## 3.4 Analyse des besoins

Le développement d'un nouveau système, ou l'amélioration d'un système existant, doit répondre à des besoins;

### 3.4.1 Capture des besoins fonctionnels

La capture des besoins fonctionnels consiste à étudier les fonctions du système et prépare la partie suivante «la partie d'analyse ».

#### 3.4.1.1 Les besoins fonctionnels

- Authentification des utilisateurs.
- Intégrité des données.

- Cryptographie.

## 3.5 Solutions proposées

Compte tenu des problèmes rencontrés précédemment et dans le souci de trouver des solutions appropriées. Nous avons proposé d'implémenter un protocole d'authentification pour gérer l'accès des utilisateurs au mini chat, ainsi qu'un système de partage de clés afin d'assurer la confidentialité et l'intégrité des messages.

*Authentification* : Le fait de s'authentifier permet d'accéder à des informations personnalisées en fonction du profil de la personne. Cette personnalisation de l'information est aussi un gage de confidentialité des données. Les informations vous concernant ne sont visibles que par vous.

*Partage de clés* : C'est un moyen pour chiffrer/déchiffrer des données confidentielles entre personnes ayant la clé partagée afin d'assurer un partage de données en toute sécurité. Le chiffrement est un processus qui désigne la transformation de l'information de manière à ce qu'un tiers non autorisé ne puisse pas la lire. Néanmoins, une personne de confiance peut déchiffrer les données et y accéder dans sa forme originale. Il existe de nombreuses méthodes de chiffrement/déchiffrement, mais la clé de la sécurité n'est pas un algorithme propriétaire. La chose la plus importante est de garder la clé de chiffrement pour vous ainsi seulement les personnes de confiance la connaîtront.

### 3.5.1 Les protocoles proposés

- **Protocole d'authentification : Kerberos**

Ce qui nous a encouragés à opter pour le protocole Kerberos, c'est bel et bien sa compatibilité et sa commodité à notre application Mini-Chat. En outre nous y trouvons une implémentation fiable, simple, rapide et correspondante. Jusqu'au aujourd'hui, l'utilisation de Kerberos est d'actualité et d'un large usage.

Par contre le protocole RADIUS ne nous correspond pas parce qu'il fonctionne sous le protocole UDP qui est complètement diffèrent du notre qui est TCP/IP. En plus, le protocole RADIUS est inconnu au public.

- **Protocole de partage de clés : Système hybride (RSA, AES)**

Comme cryptographie asymétrique est la plus fiable et plus sûre, car elle assure la confidentialité, la non-répudiation et un nombre minimum de clés à partager, mais aussi elle demande beaucoup de calculs ce qui la rend lente, alors que la cryptographie symétrique brille par sa rapidité, mais aussi cette dernière souffre d'une grave lacune, on doit transmettre les clés de manière sécurisée (sur un canal authentifié). De ce fait, une combinaison de ces deux systèmes donne naissance à un système hybride plus sûr et plus rapide.

A travers la phase d'initialisation, l'algorithme RSA est utilisé entre les communicateurs pour qu'ils puissent partager une clé secrète AES. Ensuite, pendant la communication, les messages échangés sont chiffrés avec la clé secrète AES pour un chiffrement et un déchiffrement plus rapide.

## 3.6 Présentation d'UML

UML (Unified Modeling Language V2.5) est un langage formel, normalisé et un support de communication performant qui permet grâce à sa représentation graphique, de concevoir des solutions, de faciliter la comparaison et l'évolution de celles-ci. Son caractère polyvalent et sa souplesse ont en fait un langage de modélisation universel [4].

UML propose des diagrammes complémentaires qui permettent la modélisation d'un projet tout au long de son cycle de vie.

Chaque diagramme étant dédié à la représentation des concepts particuliers d'un système logiciel. Les types de diagrammes UML, sont répartis en deux catégories [41]:

- Diagrammes structurels.
  - Diagramme de classes.
  - Diagramme de package.
  - Diagramme d'objets.
  - Diagramme de déploiement.
  - Diagramme de structure composite.

- Diagramme de composants.

❖ Diagrammes comportementaux.

- Diagramme de cas d'utilisation.
- Diagramme d'activité
- Diagramme d'état.
- Diagramme d'interaction.
- Diagramme de séquence.
- Diagramme de communication.
- Diagramme d'interaction.
- Diagramme de vue d'ensemble des interactions.

### 3.6.1 Processus de développement

**Le processus unifié 2TUP :** *2Track Unified Process* (noté 2TUP) est un processus de développement de logiciel qui implémente le Processus Unifié. Il propose un cycle de développement en Y qui sépare les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages que les acteurs échangent avec le système, à produire le cahier des charges et à modéliser le contexte [35].

Les phases de 2TUP : Le schéma suivant décrit plus précisément le processus.

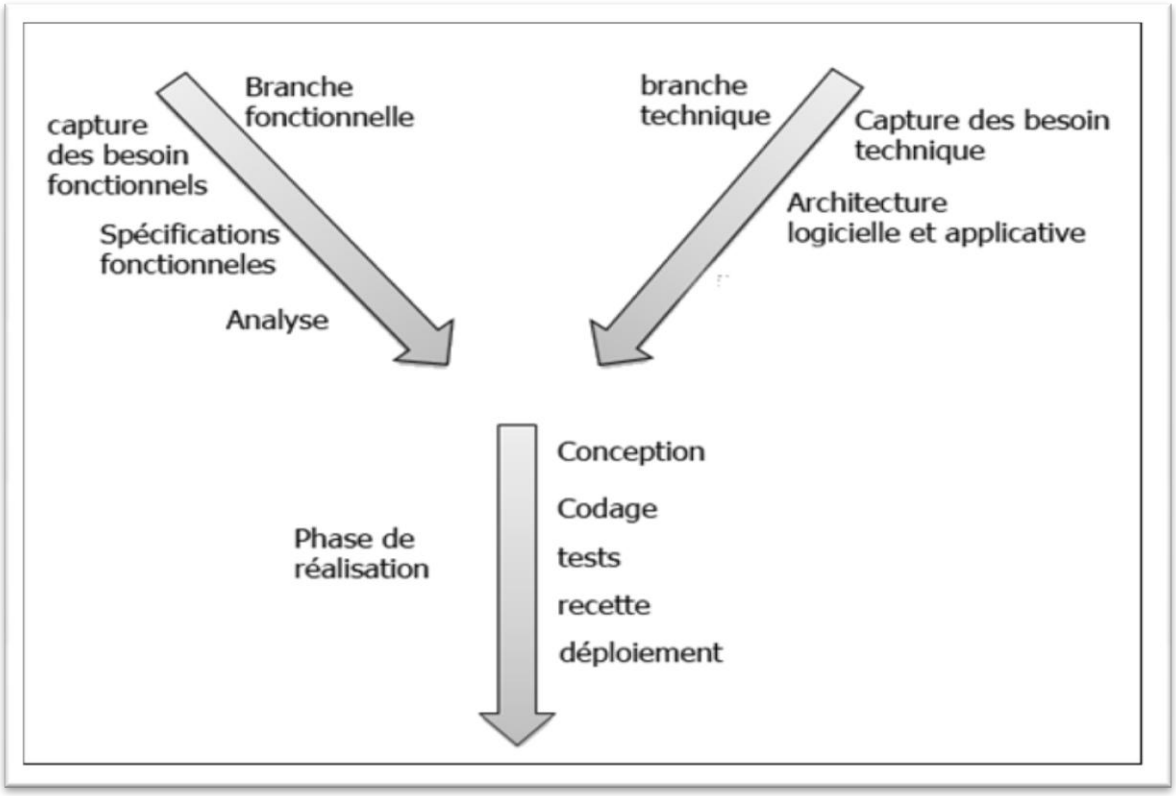


Figure. 3.1– Le processus de développement en Y.

Les contraintes fonctionnelles (La branche gauche**)** comportent :

- *La capture des besoins fonctionnels :* qui produit un modèle des besoins focalisé sur le métier des utilisateurs.

- *L'analyse* : qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée sur ce que va réaliser le système en termes de métier.

L'architecture technique (la branche droite) comporte :

- *La capture des besoins techniques* : recense toutes les contraintes et les choix dimensionnant la conception du système. Les utilisés et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement prés-requis d'architecture technique.

- *La conception générique* **:** définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est la moins dépendante possible des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout le système.

La branche du milieu comporte :

- *La conception préliminaire :* représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- *La conception détaillée* **:** étudie ensuite comme réaliser chaque composant.
- *L'étape de codage :* qui produit ces composants et teste au fur et à mesure les unités de code réalisées.
- *L'étape de recette* **:** qui consiste enfin à valider les fonctions du système développé.

## 3.7 Identification des cas d'utilisations

Un cas d'utilisation décrit sous la forme d'actions, le comportement du system étudié du point de vu des utilisateurs. Il définit les limites du system et ses relations avec son environnement [39].

Nous avons identifié les cas d'utilisations suivants :

| N° | Cas d'utilisation | | Acteur |
|---|---|---|---|
| 1 | s'inscrire | | Client |
| 2 | authentification | Récupérer le ticket TGT | Client |
| | | Récupérer le ticket TGS | |
| 3 | modifier le profil | Modifier le pseudo | Client |
| | | Modifier le mot de passe | |
| 4 | voir la liste des utilisateurs | | Client |

| | | | |
|---|---|---|---|
| 5 | échange de messages | Envoyer un message | Client |
| | | Recevoir un message | |
| 6 | échange de pièces jointes (fichier, image,...) | Envoyer une pièce | Client |
| | | Recevoir une pièce | |
| 7 | générer une clé RSA | | serveur de clé RSA |
| 8 | générer un ticket | Générer un ticket TGT | serveur d'authentification |
| | | Générer un ticket TGS | |

Tableau. 3.1 – Les cas d'utilisations associés au système.

## 3.8 Diagramme de cas d'utilisation

C'est un formalisme permettant de modéliser le fonctionnement d'un système par un découpage en fonctionnalités. Il illustre de plus la nature des interactions avec ces fonctionnalités offertes à titre de services à des acteurs externes au système. Chaque fonctionnalité est appelée un cas d'utilisation [9].

Chaque scénario doit passer par le cas d'utilisation «Authentification», ce qui explique l'utilisation de la relation «include».

Nous illustrons à travers le diagramme suivant, le cas d'utilisation relatif à un client.

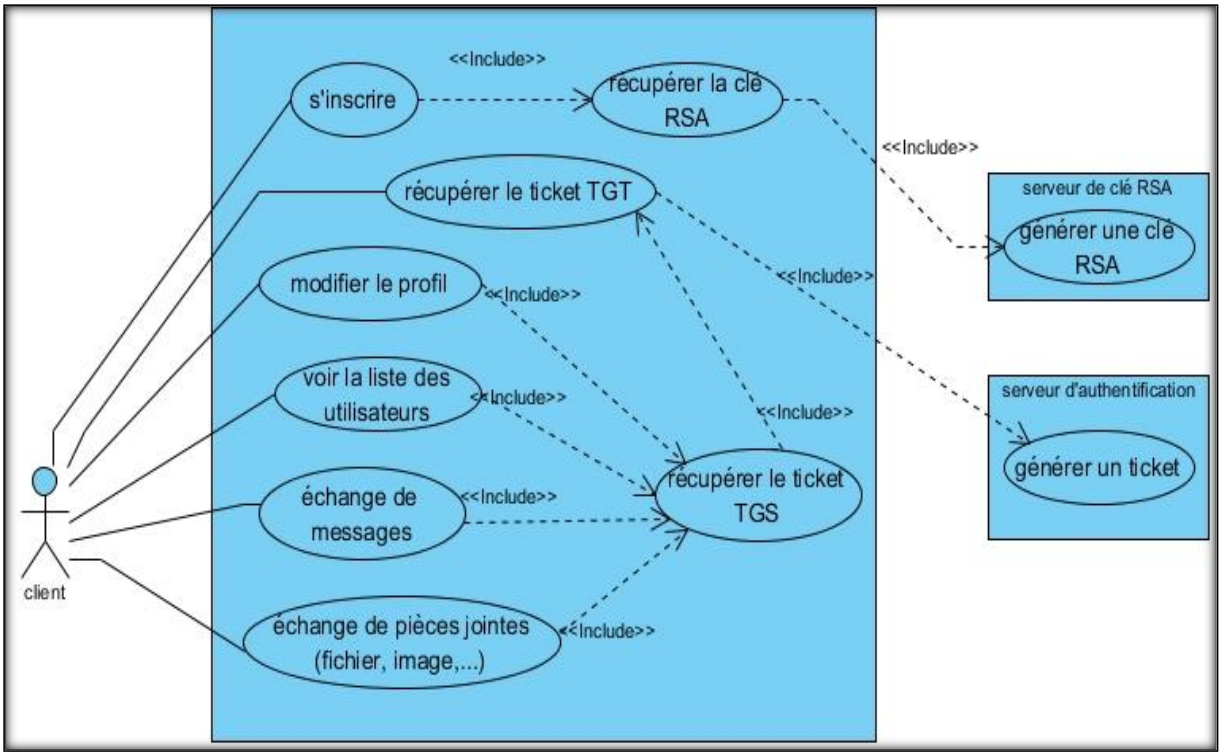


Figure. 3.2– Diagramme de cas d'utilisations associés à un client.

## 3.9 Diagramme de collaboration

Le diagramme de collaboration (appelé également diagramme de communication) permet de mettre en évidence les échanges de messages entre objets, et où la chronologie n'intervient que de façon annexe. Cela nous aide à voir clair dans les actions qui sont nécessaires pour produire ces échanges de messages. Et donc de compléter, si besoin, les diagrammes de séquence et de classes.

Il consiste en un graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) les échanges entre ces objets [12].

### 3.9.1 Diagramme de collaboration « Inscription »

La figure suivante représente la collaboration entre le client et le système ainsi que les messages échangés pour réaliser l'étape de l'inscription.

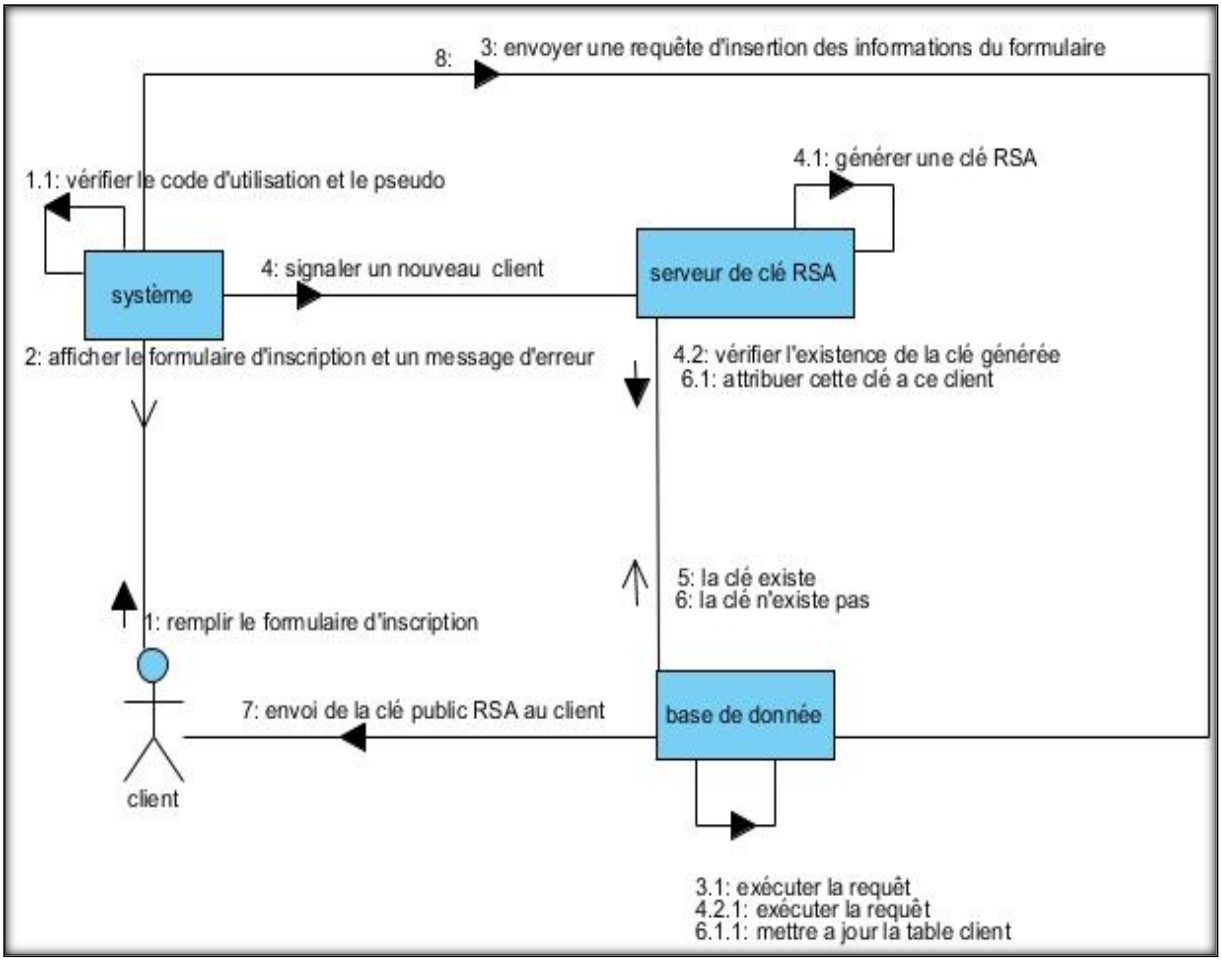


Figure. 3.3– Diagramme de collaboration « Inscription ».

### 3.9.2 Diagramme de collaboration « Authentification »

La figure suivante représente la collaboration entre le client et le système ainsi que les messages échangés pour réaliser l'étape de l'authentification.

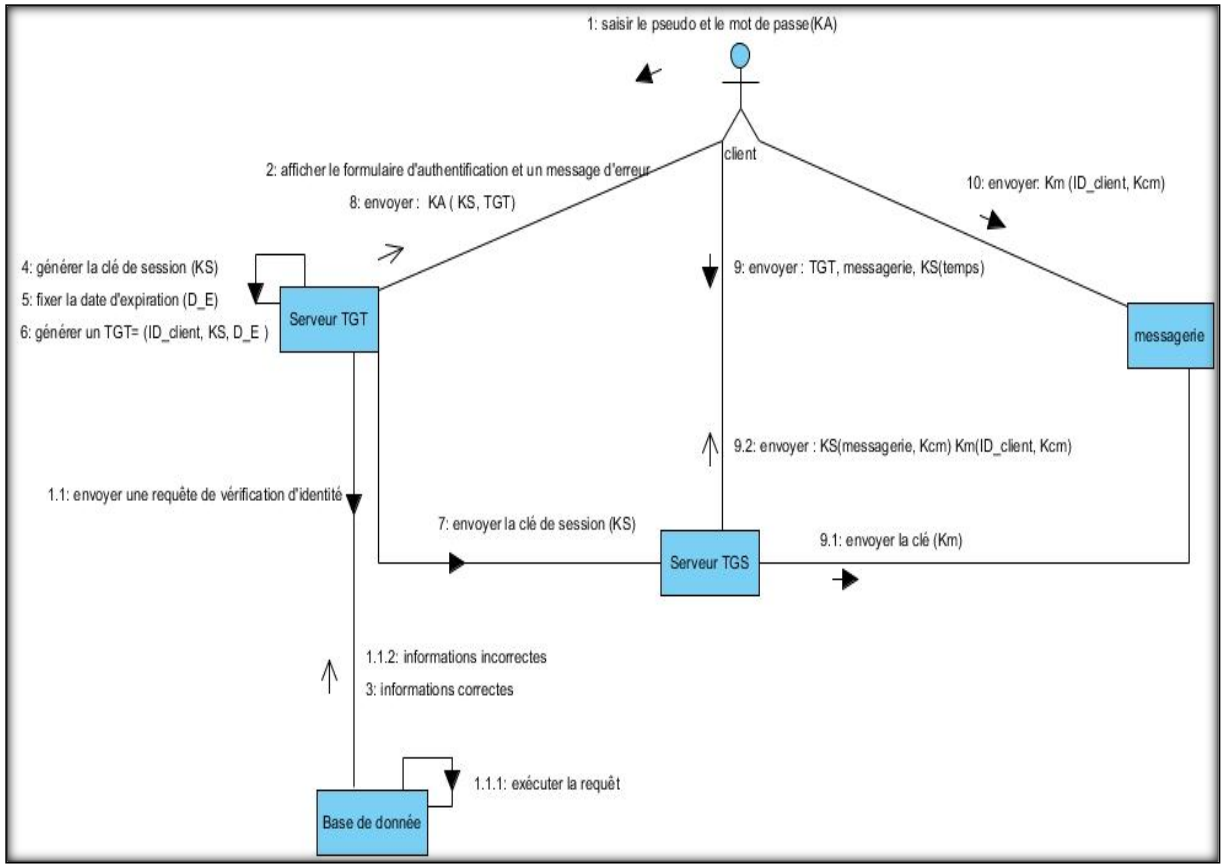


Figure. 3.4– Diagramme de collaboration « Authentification ».

### 3.9.3 Diagramme de collaboration « partage de clé »

La figure suivante représente le diagramme de collaboration entre le client et le serveur ainsi que la façon d'échanger les clés entre eux.

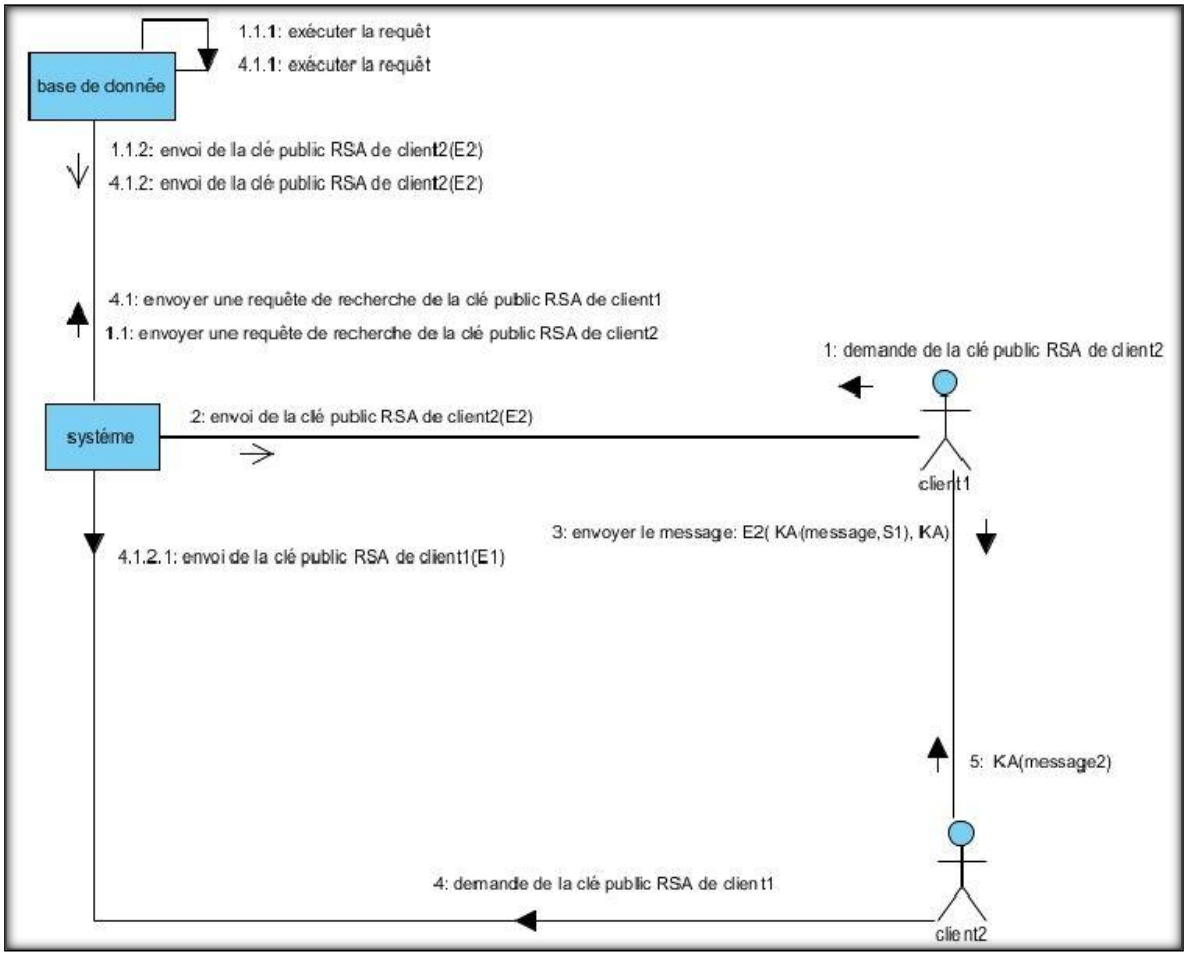


Figure. 3.5– Diagramme de collaboration « partage de clé».

## 3.10 Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique, ils permettent de cacher les interactions d'objets dans le cadre d'un scénario d'un diagramme de cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche et les acteurs secondaires éventuels à droite du système [35].

### 3.10.1 Diagramme de séquence du cas d'utilisation « Inscription »

L'inscription est indispensable pour les clients afin d'avoir les privilèges réservés pour accéder à l'application. Le système affiche le formulaire d'inscription qui sera compléter par le client, ce qui permettra de créer son compte.

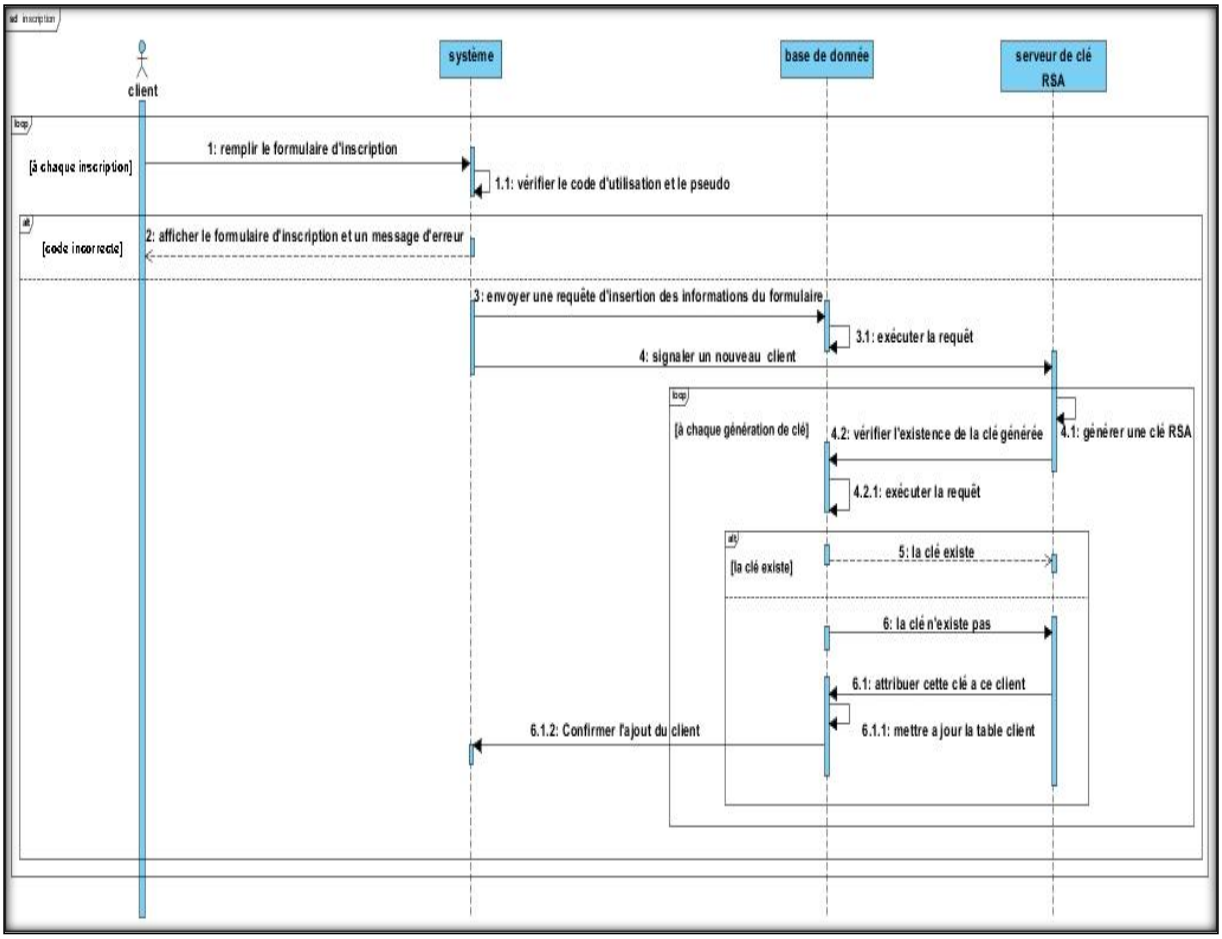


Figure. 3.6 – Diagramme de séquence du cas d'utilisation « Inscription ».

### 3.10.2 Diagramme de séquence du cas d'utilisation « Authentification »

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification des informations associées à un utilisateur (généralement un login et un mot de passe). Ces informations sont préétablies dans une base de données.

Lors de l'authentification d'un utilisateur, deux cas peuvent se présenter : informations correctes ou informations incorrectes, ce qui explique l'utilisation de l'opérateur «alt». Si les informations fournies sont correctes, alors le système accorde l'accès à l'interface appropriée.

En revanche, si l'utilisateur saisit des informations incorrectes, le système génère un message d'erreur et réaffiche la page d'authentification.

Ce procédé est exécuté à chaque fois que l'utilisateur tente de s'authentifier, c'est pourquoi nous avons utilisé l'opérateur « Loop ».

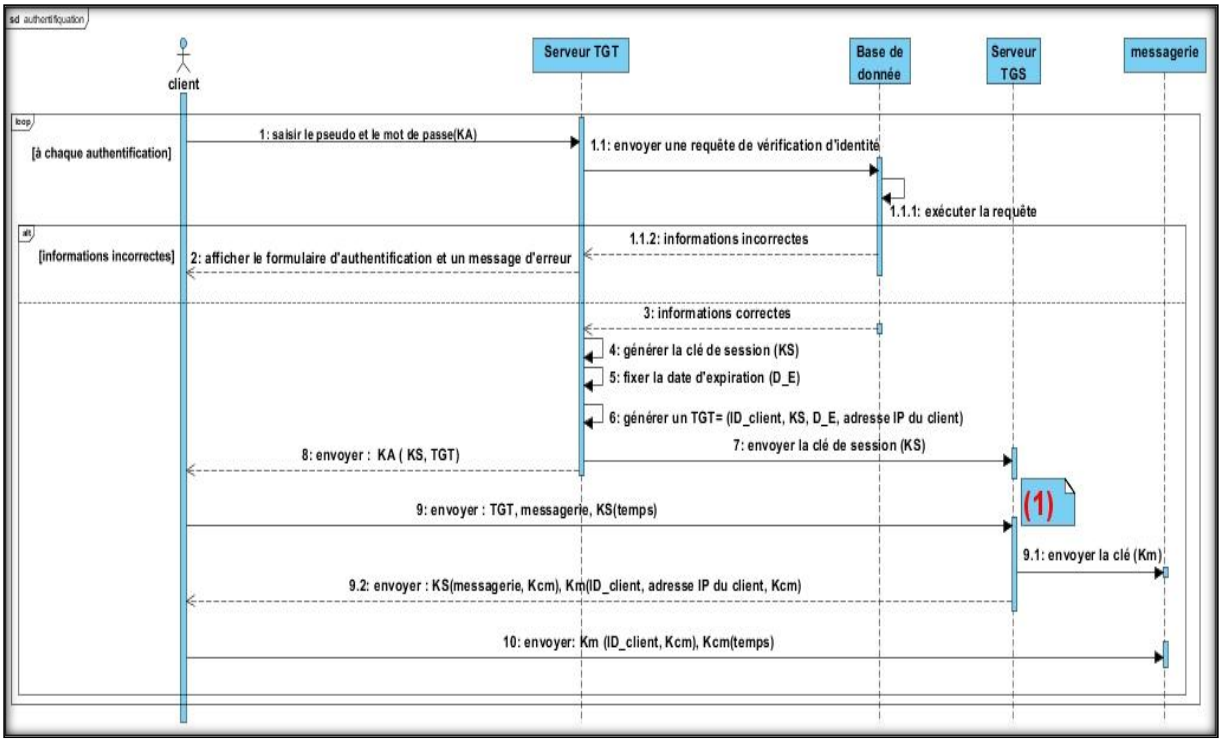


Figure. 3.7 – Diagramme de séquence du cas d'utilisation « Authentification ».

(1) :

- générer une clé secrète (Kcm) entre le client et service messagerie.
- générer une clé (Km) pour le service messagerie

### 3.10.3 Diagramme de séquence du cas d'utilisation « partage de clé »

Après authentification, le serveur enregistre la clé publique de chaque client dans la base de données et il envoi aussi à chaque client sa clé privée chiffrée avec son mot de passe.

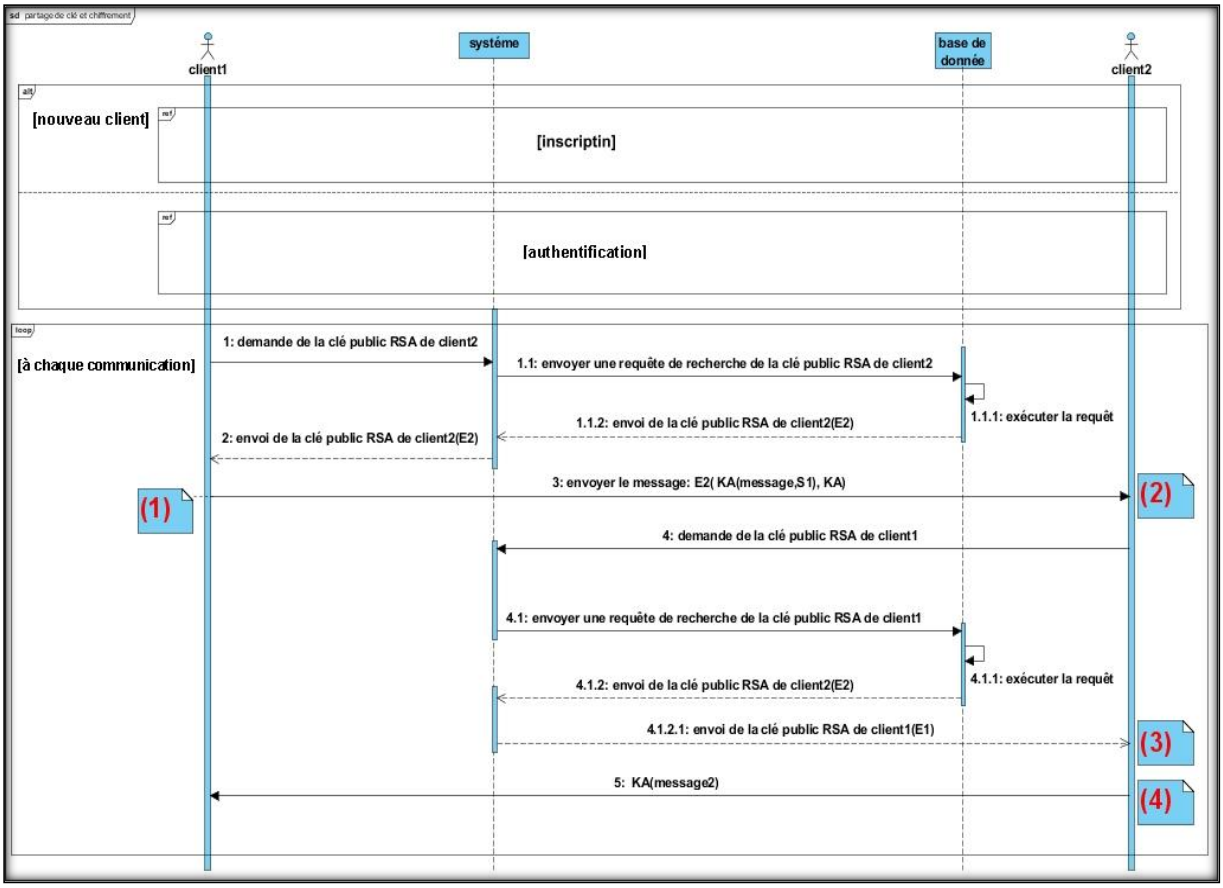


Figure. 3.8 – Diagramme de séquence du cas d'utilisation « partage de clé ».

(1) :

- Rédiger un message.
- Produit un condensat du message par la fonction de hachage H(m1).
- Chiffrer le H(m1) avec sa clé privée (S1).
- Générer une clé AES (KA).

(2) :

- Client2 déchiffre le message avec sa clé privée RSA.
- Récupéré la clé AES (KA).
- Déchiffrer le message avec la clé AES (KA).
- Il produit un condensat du message clair en utilisant la même fonction du hachage (H) : (H(m2)).

(3) :

- Client2 déchiffre la signature (S1) avec la clé publique de client1.
- Récupérer le condensat (H (m1)).
- Comparer H (m1) et H (m2).

(4) :

- Rédiger un message.
- Chiffrer le message avec la clé AES (KA)

## 3.11 Diagramme de classes

Le diagramme de classes est le point essentiel dans un développement orienté objet. En analyse, il a pour but de décrire la structure des entités manipulées par les utilisateurs. En conception, on représente la structure d'un code orienté objet ; ou à un niveau de détail plus important, les modules du langage de développement [40].

### 3.11.1 Diagramme de classes du projet à réaliser

Le diagramme de classe relatif à notre projet, est représenté dans la figure suivante :

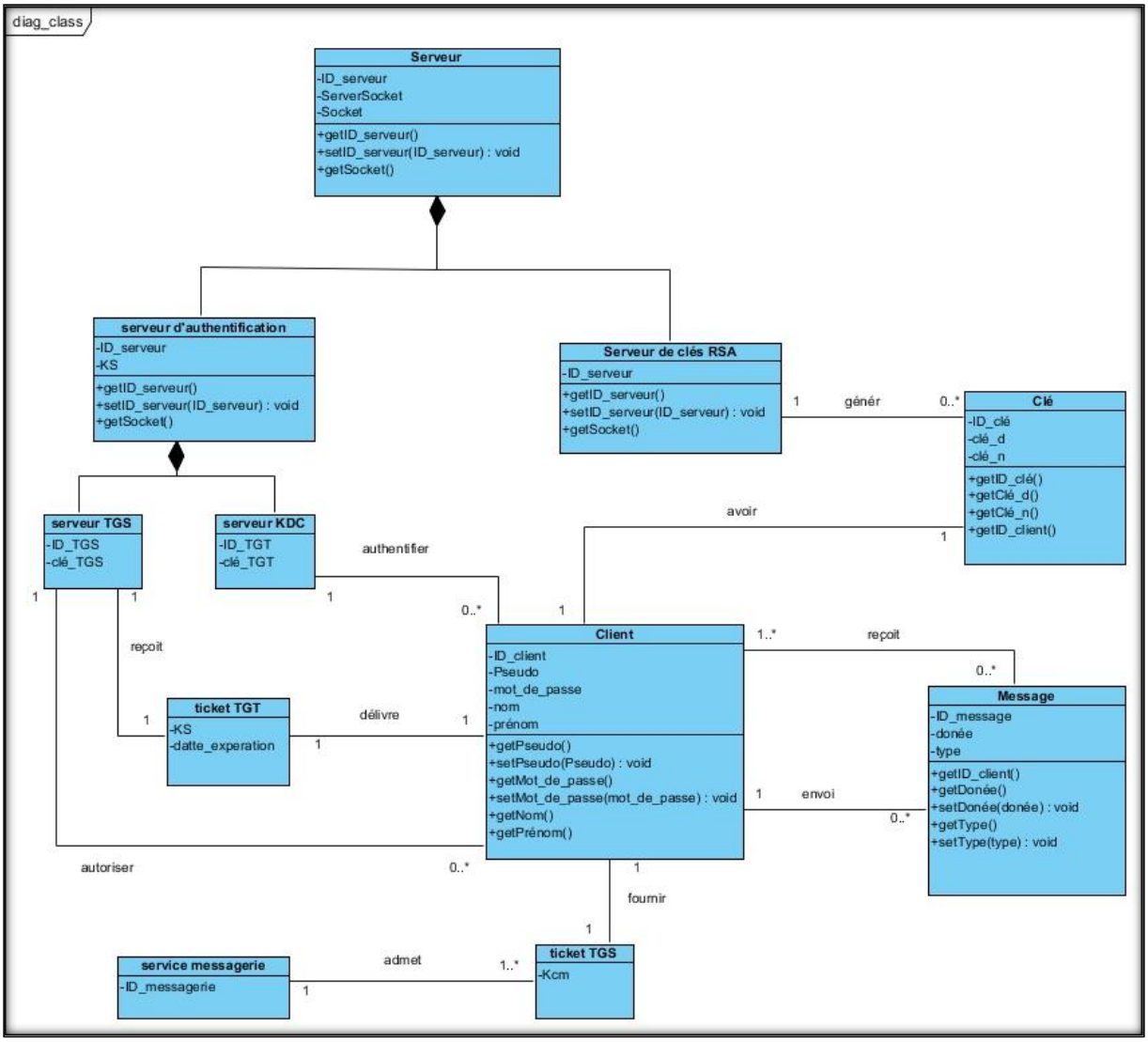


Figure. 3.9 – Diagramme de classe.

## 3.12 Conclusion

Dans ce chapitre nous avons détaillé ceux que nous avons choisis comme protocole d'authentification (Kerberos) et protocole de partage de clé (le protocole hybride RSA, AES). En outre, nous avons présenté la conception du notre application sous forme d'UML.

# Chapitre 4

## *Implémentation*

### 4.1 Introduction

Ce projet s'inscrit dans le domaine de la sécurité informatique, plus précisément la sécurité des systèmes distribués. Il consiste à concevoir et à réaliser un protocole d'authentification sur un système distribué. Il se base sur l'environnement JAVA qui est à la fois un langage de programmation orienté objet et un environnement d'exécution avec la particularité principal, que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation.

Nous présentons dans ce chapitre, les étapes de conception, les outils de réalisation ainsi que des résultats d'implémentation, et quelques scenarios d'attaques.

## 4.2 Environnement et outils de développement

Pour développer le mini chat, on a utilisé :

### 4.2.1 La JDK (JAVA Developement Kit)

Version 1.8.0_91-b14 : l'environnement dans lequel le code JAVA est compilé pour être transformé en bytecode (code intermédiaire) afin que la machine virtuelle de JAVA (JAVA Virtual Machine) puisse l'interpréter [27].

### 4.2.2 Eclipse

Est un IDE, *Integrated Development Environment* (EDI environnement de développement intégré), un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

Au fur et à mesure que vous programmez, éclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problème qu'il décèle [24].

### 4.2.3 WampServer

Est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration des bases MySQL [30].

### 4.2.4 MySQL(*My Structured Query Language*)

Est un système de gestion de bases de données *client/serveur open source*. Son rôle consiste à stocker et à gérer une grande quantité de données en les organisant sous forme de tables. Le système *MySQL* doit aussi permettre la manipulation de ces données à travers le langage standard du traitement des bases de données *SQL* qui est un langage de requêtes vers les bases de données [38].

### 4.2.5 Le serveur apache

Son rôle est de transférer les pages du format texte au format html et les fichiers joints au client (navigateur internet) qui se connecte sur le site web. Ce serveur fonctionne aussi bien sous linux que Windows ou d'autres plates formes [43].

### 4.2.6 Le package JDBC (Java DataBase Connectivity)

Est une API (Application Programming Interface) JAVA permettant d'accéder aux bases de données à partir d'un code JAVA à travers des requêtes SQL [18].

### 4.2.7 Le package Mysql-connector

MySQL fournit des pilotes basés sur les standards pour JDBC, ODBC, et .Net permettant aux développeurs de créer des applications de base de données dans la langue de leur choix. En outre, une bibliothèque native C permet aux développeurs d'intégrer MySQL directement dans leurs applications.

## 4.3 Le modèle client/serveur

***Sockets TCP***

Le protocole TCP offre un service en mode connecté et fiable. Les données sont délivrées dans l'ordre de leur émission.

La procédure d'établissement de connexion est dissymétrique. Un processus, appelé serveur, attend des demandes de connexion qu'un processus, appelé client, lui envoie. Une fois l'étape d'établissement de connexion effectuée le fonctionnement redevient symétrique.

Il est à noter que côté serveur on utilise deux sockets : l'un, appelé socket d'écoute, reçoit les demandes de connexion et l'autre, appelé socket de service, sert pour la communication. En effet, un serveur peut être connecté simultanément avec plusieurs clients et dans ce cas on utilisera autant de sockets de service que de clients [6].

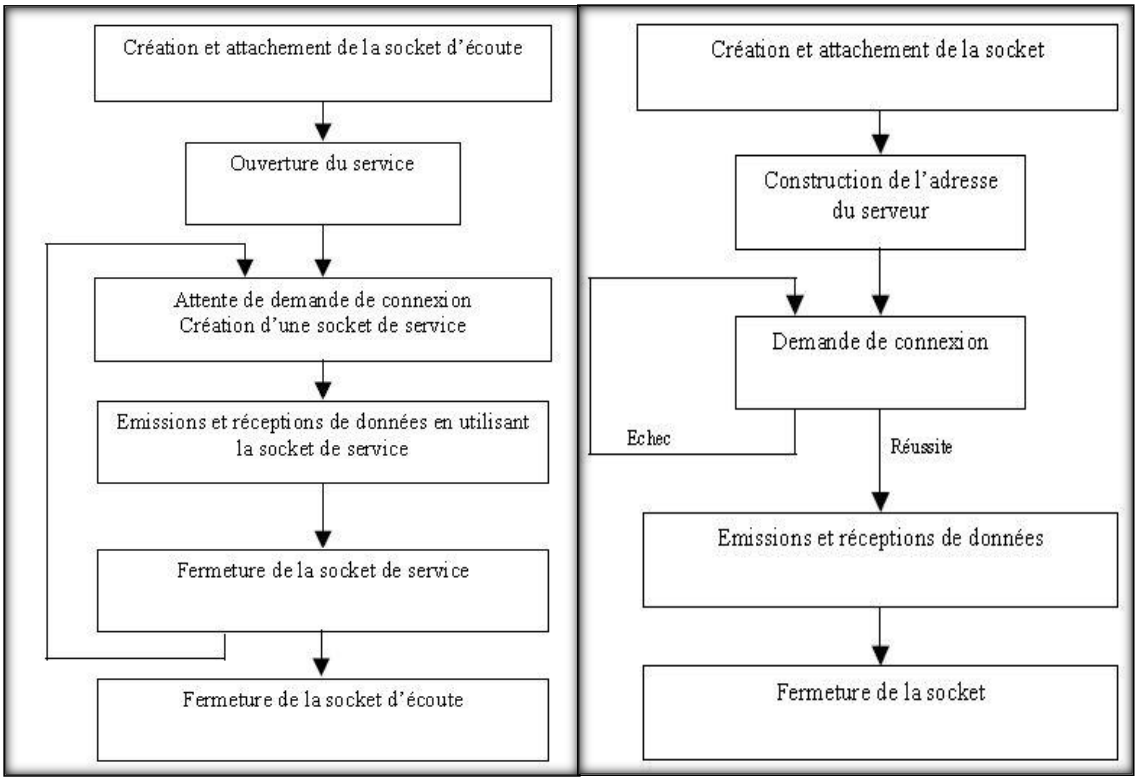


Figure. 4.1 – Fonctionnement d'un serveur et d'un client.

## 4.4 Implémentation du Kerberos

Vu que l'authentification avec le protocole Kerberos est l'aproche la plus utilisée dans les systèmes informatiques, nous avons procedé de la même manière de créer un protocole qui se ressemble à Kerberos en gardant toutes ses principales fonctionnalités.

Nous avons créé deux classes qui représentent les deux serveurs intégrés du Kerberos, classe KDC pour l'authentification et classe TGS pour le contrôle d'accès.

Les tickets échangés entre les trois acteurs (client, KDC, TGS) sont chiffrés avec AES.

Dans notre cas, nous avons un seul service qui est l'application proposée. Une fois le dernier ticket est bien échangé entre le client et le serveur TGS, le client sera redirigé automatiquement vers l'interface de communication.

La figure suivante représente la visualisation des différentes requêtes échangées entre les entités qui compose le système.

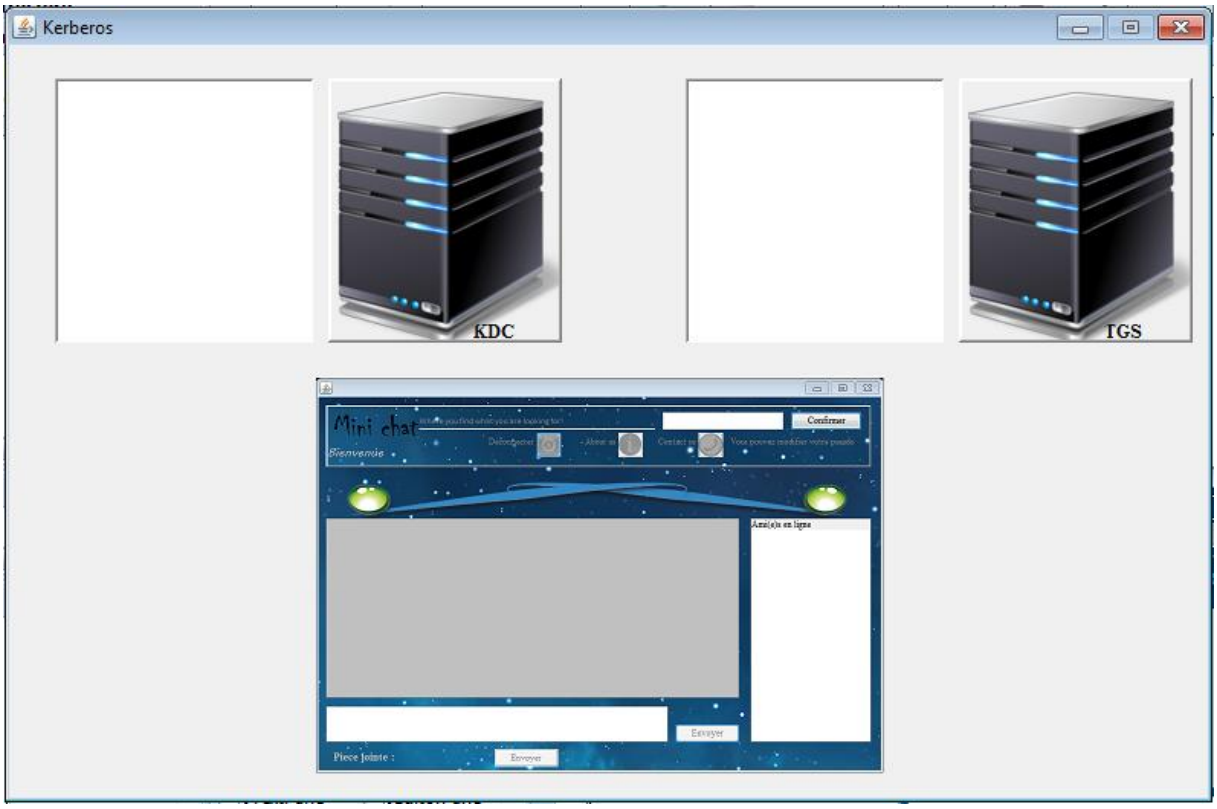


Figure. 4.2 – Interface authentification.

## 4.5 Implémentation du protocole de partage de clés (RSA, AES)

Dans le but d'avoir un échange de données sécurisé entre les utilisateurs, nous avons exploité quelques classes java telles que AES() pour une Génération de clés, chiffrement et déchiffrement AES et RSA() pour une Génération de clés, chiffrement et déchiffrement RSA.

Deux manières d'échange de messages sont utilisées par l'application. La première façon c'est d'envoyer un message à une personne bien précise une fois la sélectionnée sur la liste des amis en ligne. Dans ce cas, le système génère une clé AES et fait un appel à une méthode de chiffrement pour le chiffrer. Une fois le chiffrement AES est terminé, le système récupère la clé publique RSA de la personne choisie pour chiffrer le message précédant et la lui envoyer avec la clé AES.

A la réception, le message sera déchiffré par la clé AES, ensuite par la clé privée RSA. La deuxième façon est de diffuser un message vers tous les amis. Dans ce cas, le système chiffre le message avec la clé AES de la même manière citée déjà auparavant, ensuite il récupère la clé publique commune RSA entre les utilisateurs pour le chiffrement.

A la réception, le message sera déchiffré par la clé AES, ensuite par la clé privée commune RSA.

## 4.6 Scenario d'attaque

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables [16].

### 4.6.1 Attaques sur Kerberos

- *Replay attack* [16]

Cette attaque par rejeu nécessite que l'attaquant mette en place un *Man-In-The-Middle* entre le client et le serveur. Par la suite, il y a deux possibilités :

- Soit l'attaquant effectue une écoute du réseau et renvoie la requête émise par le client afin d'obtenir un accès au service.

- Soit l'attaquant empêche le client d'envoyer la requête au serveur et l'utilise pour obtenir l'accès au service à la place du client.

Nous avons proposé deuxcontre-mesures afin de réduire l'impact de cette vulnérabilité :

- **Timestamp** : La durée d'utilisation de l'AP_REQ est limitée à un certain temps (en général 5 minutes).

- **Adresse IP** : Le ticket fourni par le KDC peut contenir la liste des adresses IP autorisées à utiliser ce ticket.

Le diagramme de séquence suivant montre la reaction du systèm d'authentification contre cette attaque :

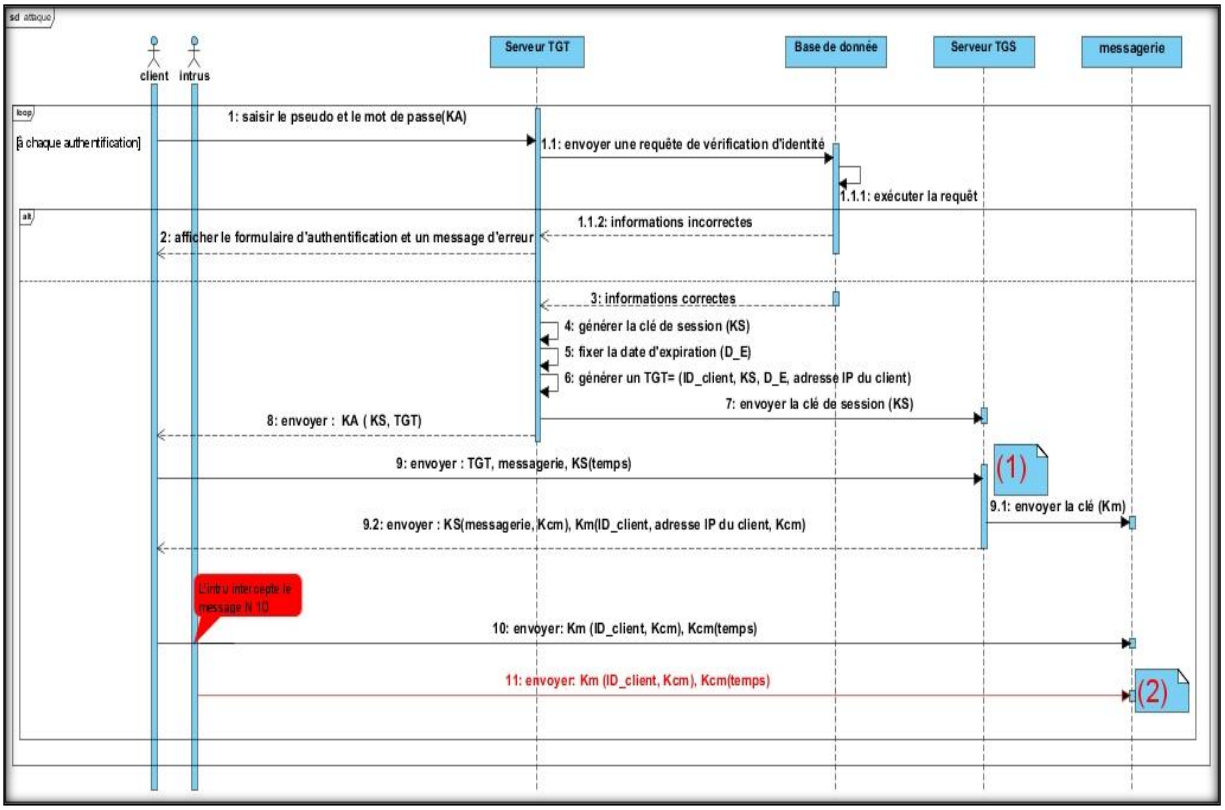


Figure. 4.3 – Réaction du système contre le rejeu.

(1) :

- Générer une clé secrète (Kcm) entre le client et service messagerie.
- Générer une clé (Km) pour le service messagerie.

(2) :

- Le service messagerie découvre que c'est un rejeu, car il a déjà reçu le même message avec le même timestemp et une adresse IP différente.

- *Kerberos et le chiffrement* [16]

Une autre catégorie d'attaque consiste à exploiter les faiblesses des algorithmes de chiffrement utilisés par Kerberos.

Historiquement, Kerberos utilisait uniquement l'algorithme DES (dont la faiblesse n'est plus à démontrer [31]). Le protocole a depuis évolué pour intégrer un mécanisme de négociation de l'algorithme de chiffrement entre le client et le KDC.

Malheureusement, cette négociation n'est pas protégée. Il est donc possible pour un attaquant ayant accès aux flux réseau entre le client et le KDC de modifier ces échanges afin de forcer l'utilisation de l'algorithme de chiffrement le plus faible [10].

Nous avons proposé un chiffrement avec l'algorithme AES comme contre-mesures:

- Taille de bloc de 128 bits.
- Tailles de clé de 128, 192 et 256 bits.

AES repond aux deux principaux paramètres de sécurité mieux que DES qui sont :

– La taille du bloc (e.g. n = 64 ou 128 bits). Les modes opératoires permettent généralementdes attaques quand plus de $2^{n/2}$ blocs sontchiffrés avec une même clé.

– La taille de clé (e.g. k = 128 bits). Pour un bonalgorithme, la meilleure attaque doit coûter $2^{k}$opérations (recherche exhaustive).

### 4.6.2 Attaques sur RSA

- *L'attaque de l'homme du milieu*

L'attaque de « l'homme du milieu » peut être employé de façon évidente pour casser cetalgorithme. Supposons dans notre cas qu'Eve soit une espionne et qu'elle veuille intercepter lesconversations entre Alice et Bob. Il suffit alors à Eve de choisir une clef privée b et une clefpublique qui convient (a, n) qu'elle envoieà Alice en lui faisant croire que ce sont les nouvellesclefs de Bob. Notre espionne peut alors lorsqu'elle intercepte les messages d'Alice, les déchiffrer et les comprendre. Elle lui reste donc à s'assurer que Bob ne se rende compte de rien en rechiffrantles messages d'Alice avec l'ancienne clef publique de Bob qui reçoit le messagecomme si aucune opération n'avait été effectuée dessus. S'il le veut il peut effectuer la mêmeopération en sens inverse, pour récupérer les messages que Bob envoie à Alice.

Pour contrercette attaque, il est très important pour Alice et Bob de toujours s'assurer que personne netente de modifier leurs messages. Pour cela, les méthodesles plus utilisées sont celles d'authentification et de signature [33].

## 4.7 Déscription des interfaces de l'application

### 4.7.1 Interface serveur

Ceci est l'apperçu de l'interface d'un serveur permettant l'ouverture et la fermeture d'une connexion, ainsi que répondre aux requêtes envoyées par des clients.

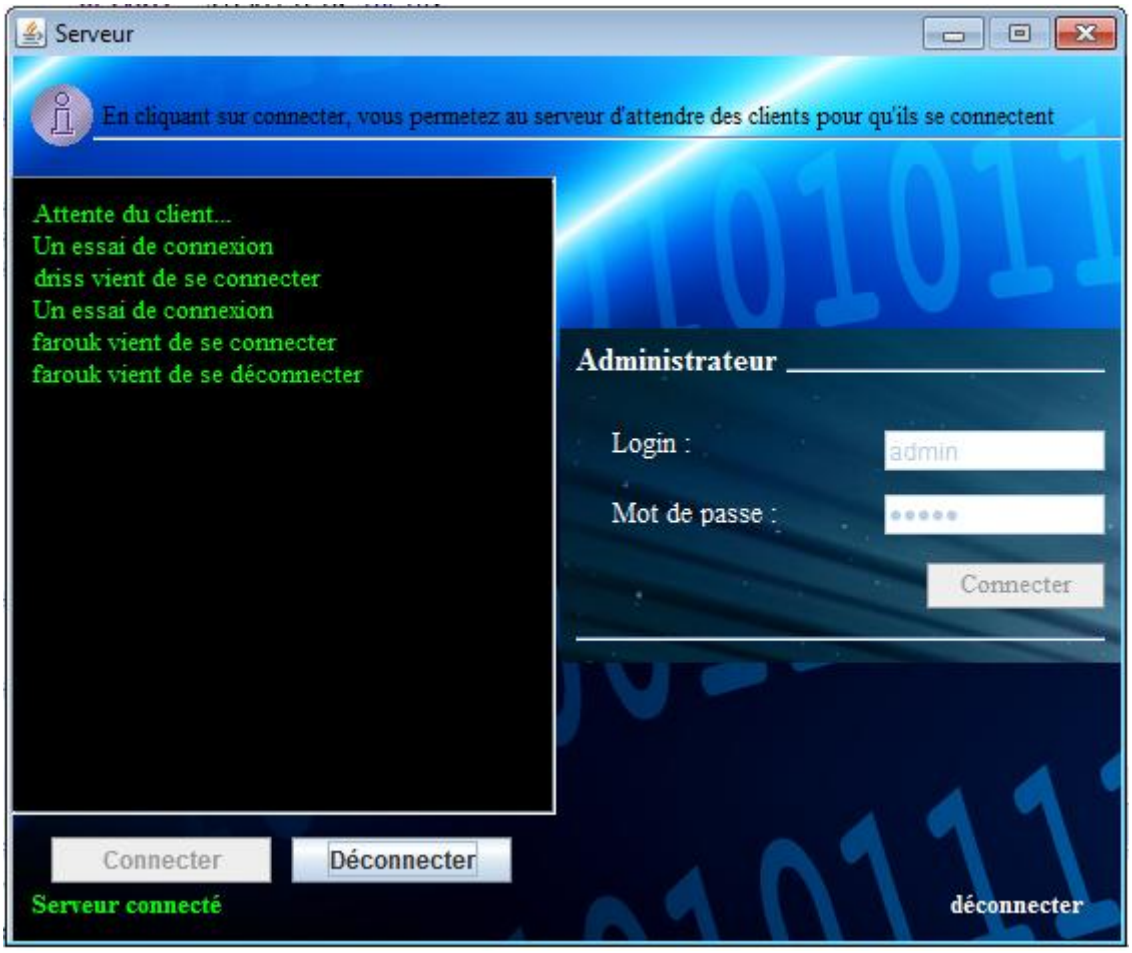


Figure. 4.4 – Interface serveur.

### 4.7.2 Interface client (Inscription / Connexion)

Ceci est l'apperçu de l'interface utilisateur permettant à une personne privilégiée de s'inscrire ou de se connecter avant de commencer la discussion avec d'autres personnes.

Avant de se connecter, il faut d'abord allumer le serveur pour qu'il libère la connexion.

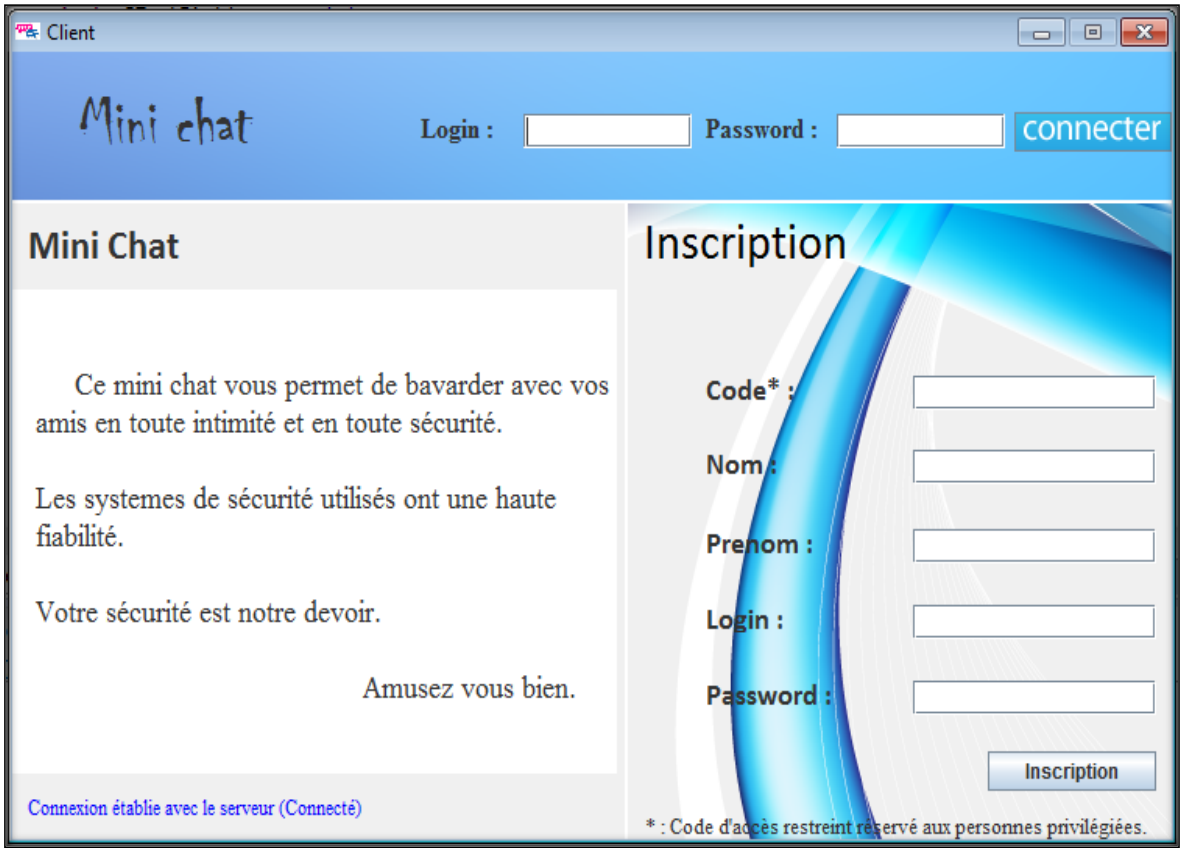


Figure. 4.5 – Interface client (Inscription / Connexion).

### 4.7.3 Interface de discussion

Apres une confirmation du login et du mot de passe de l'utilisateur, une interface de discussion va s'apparaitre. Cette interface contient 4 zones principales.

- Une zone pour recevoir des messages.
- Une zone pour envoyer des messages.
- Une zone pour afficher les amis en ligne.
- Une zone pour désigner si le message est diffusé ou non.

Avant d'envoyer un message à un ami , il faut le séléctioner afin d'eviter la diffusion du message vers toute personne connectée.

Cette interface permet aussi d'envoyer des pieces jointes en cliquant sur « piece jointe » pour choisir la piece à envoyer.

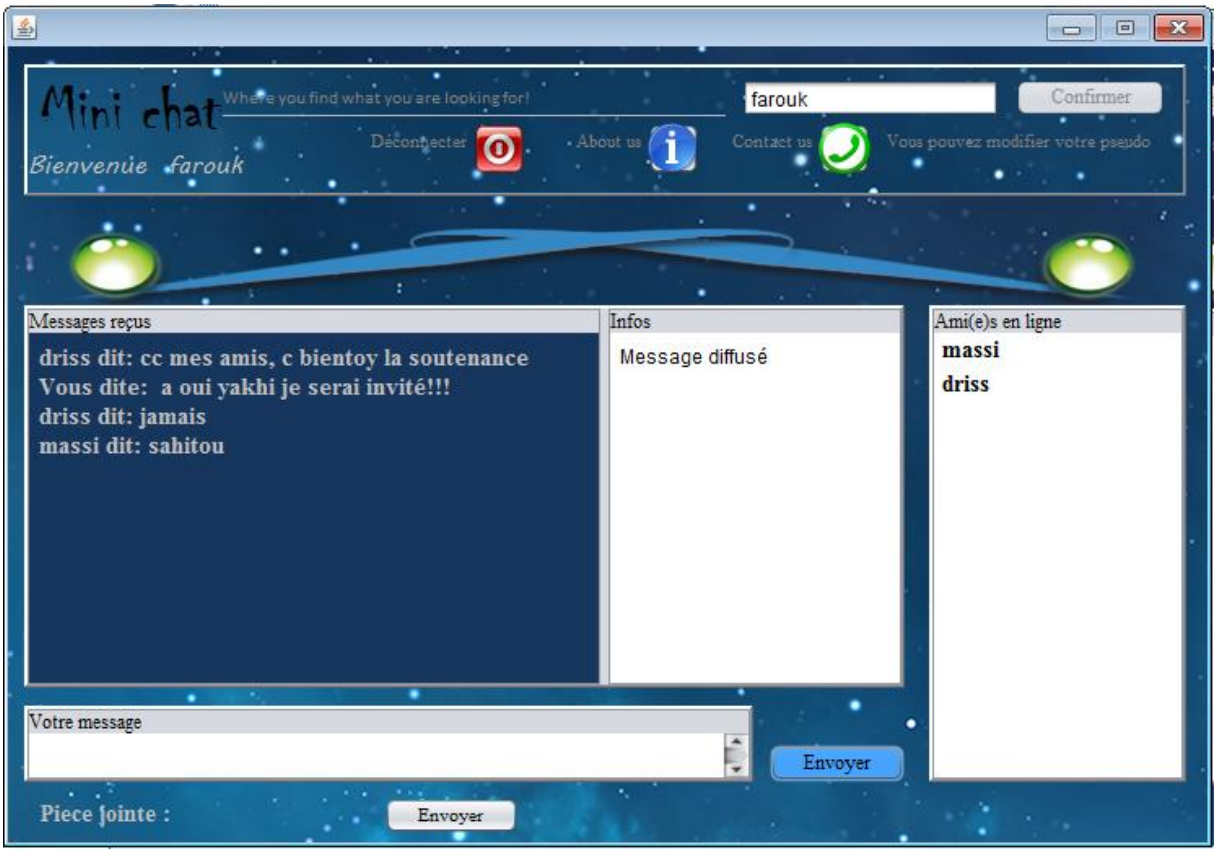


Figure. 4.6 – Interface discussion.

### 4.7.1 Interface information

Cette interface nous donne les differentes information sur l'authentification et les messages chiffrés.

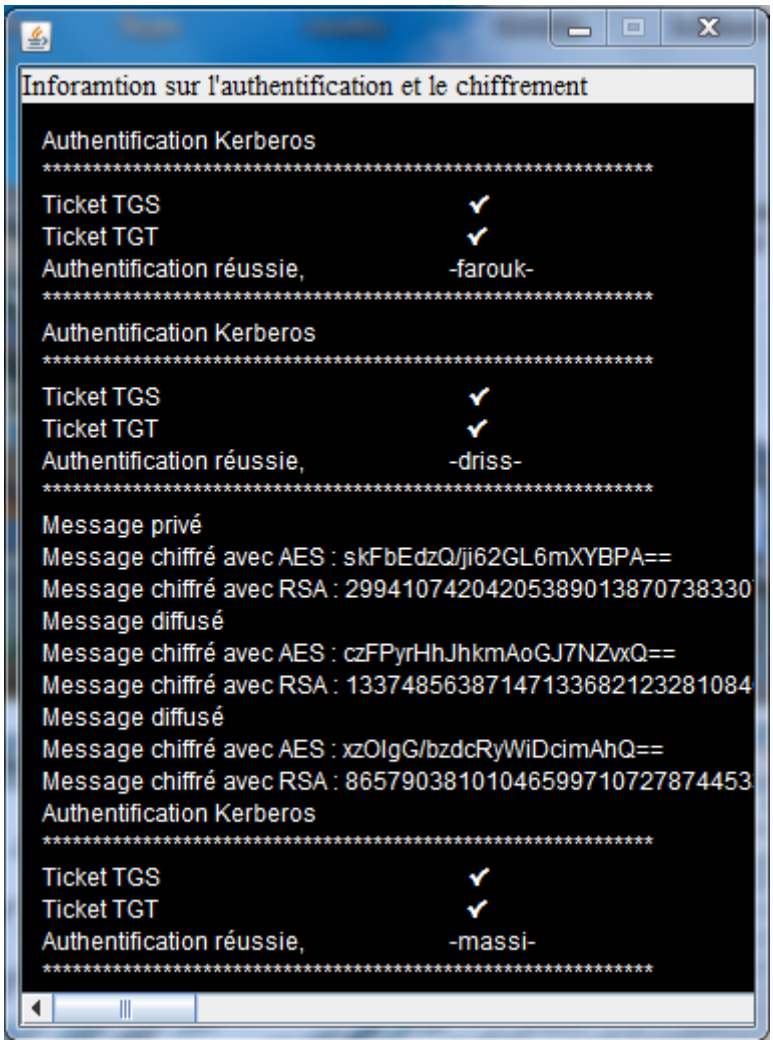


Figure. 4.7 – Interface information.

## 4.8 Conclusion

Au cours de ce chapitre, nous avons décrit brièvement les outils de réalisation de notre application en spécifiant l'environnement et outils de développement, ainsi quelques scénarios d'attaques et des interfaces de Mini-Chat.

# Conclusion générale

Dans ce modeste travail nous avons consacré la première partie à une étude générale sur la sécurité et quelques techniques cryptographiques existantes, où nous avons énuméré quelques systèmes d'authentification et protocoles de partage de clés les plus utilisés. La deuxième partie est, quant à elle, consacrée au système d'authentification Kerberos et le protocole hybride (RSA, AES) de partage de clés et de chiffrement qui est l'objet de notre thématique.

Ensuite, nous avons modélisé ce système et montré ses principales fonctionnalités ainsi que ces différentes composantes en utilisant les diagrammes de modélisation UML. Enfin, nous avons présenté le prototype du système qu'on a implémenté sous JAVA.

# Bibliographies


[1] B. TUNG, Kerberos, a Network Authentication System– Edition Addison –Wesley.

[2] C. DUVALLET, le protocole radius remote authentification dial-in user service.

[3] C. SCHRYVE - L.GAJNY, Cryptographie à clef publique24 mai 2010.

[4] C.MORLEY, J. Hugues, B.LEBLANC, UML2 pour l'analyse d'un système d'information, DUNOD, 4ème édition, 2000.

[5] E.CHEVOIR. «Etude de kerberos 5 », Rapport technique.

[6] E. Rusty Harold, Programmation réseau avec Java, O'Reilly Editions, 1997

[7] G. Florin, S Natkin, LA SÉCURITÉ, CEPADUES Editions, 1995.

[8] Groupe CGI, Cryptographie à clé publique et signature numérique Principes de fonctionnement, Septembre 2002.

[9] G. Roy, Conception de bases de données avec UML, Presses de l'Université de Québec, 2009,1ere édition.

[10] Https://media.blackhat.com/bh-us-10/whitepapers/Stender_Engel_Hill/BlackHat-USA-2010-Stender-Engel-Hill-Attacking-Kerberos-Deployments-wp.pdf (22/06/2016).

[11] Https://interstices.info/jcms/c_30225/nombres-premiers-et-cryptologie-l-algorithme-rsa (22/06/2016).

[12] Https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/les-differents-types-de-diagrammes (22/06/2016).

[13] Https://support.microsoft.com/fr-fr/kb/246071 (22/06/2016).

[14] Https://www.viber.com/fr/privacypolicy.html (22/06/2016).

[15] Http://arstechnica.com/security/2013/05/think-your-skype-messages-get-end-to-end-encryption-think-again/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+arstechnica%2Findex+%28Ars+Technica+-+All+content%29&utm_content=Netvibes (22/06/2016).

[16] Http://connect.ed-diamond.com/misc/misc-054/attaque-sur-le-protocole-kerberos (22/06/2016).

[17] Http://info.crypto.free.fr/wiki.php/2.1)%20Chiffrement%20sym%C3%A9trique (22/06/2016).

[18] Http://java.developpez.com/faq/jdbc/?page=Generalites (22/06/2016).

[19] Http://math.univ-lyon1.fr/~roblot/masterpro.html (22/06/2016).

[20] Http://ser-info-02.ec-nantes.fr/users/info3/weblog/c6a6e/Principes_de_la_cryptographie_moderne.html (22/06/2016).

[21] Http://www.cases.public.lu/fr/publications/dossiers/cryptographie/crypto/cryptographie (22/06/2016).

[22] Http://www.devensys.com/blog/kerberos-principe-de-fonctionnement (22/06/2016).

[23] Http://www.dictionnaireduweb.com/messagerie-instantanee-chat/ (22/06/2016).

[24] Http://www.enseignement.polytechnique.fr/informatique/profs/Julien.Cervelle/eclipse/ (22/06/2016).

[25] Http://www.futura-sciences.com (22/06/2016).

[26] Http://www.id.image.fr/svarrett/download/polys/Tutorial_Kerberos_Html (22/06/2016).

[27] Http://www.oracle.com/technetwork/java/javase/tech/index.html (22/06/2016).

[28] Http://www.sas-informatique.com/index.php/securite-informatique/la-cryptographie/la-cryptographie-symétrique/le-chiffrement-par-flot (22/06/2016).

[29] Http://www.scribd.com (22/06/2016).

[30] Http://www.wampserver.com/ (22/06/2016).

[31] Http://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des

_ faq.html (22/06/2016).

[32] Introduction à la cryptographie, GHISLAINE Labour et HERVE Schauer consultants, 1999-2001, page 3.

[33] J. BLANC, A. DE GEORGES, techniques de cryptographie, 2004

[34] J.F PILLOU, Introduction au protocole RADIUS, Septembre 2015.

[35] J.GABAY, D.GABAY, UML2 Analyse et Conception, DUNOD, 2008, 1ere édition.

[36] MENEZES, van Oorschotet VANSTONE 1996, p. 286; Schneier 1996, Applied Cryptography, p. 467.

[37] M.CERDA Normation Mardi 10 Juillet 2012.

[38] M. Contensin, Bases de données et Internet avec PHP et MySQL, DUNOD, 2004,1ere édition.

[39] N.KOBLITZ, « Elliptic curve cryptosystems », dans *Mathematics of Computation*, n°48, 1987, p.203–209.

[40] Pascal roques, UML 2 par la pratique, Eyrolles, 2006, 5ème édition.

[41] P. ROQUES, Les cahiers du programmeur UML 2 modélisé une application web, Eyrolles, 2007.

[42] R. L RIVEST and A.SHAMIR and L.M ALDEMAN, A method for obtaining digital signature and public-key cryptosystems, communications of the ACM, 1978.

[43] R. LENTZNER , « 300 astuces pour SQL et MYSQL »,OSMAN EYROLLES MULTIMEDIA, Edition :2001.

[44] R. Smith, Authentication, Edition Addison Wesley.

[45] Stinson 2006, Cryptography: Theory and Practice, p. 175.

[46] S.VARRETTE, Tutorial Kerberos, Comprendre et mettre en place une architecture Kerberos. Avril 2004, Page 3 / 12.

[47] T. BOUDHAMAA, S. YAHYAOUI, Sécurisation des systèmes, Protocoles utilisant des mécanismes d'authentification : TACAS+ RADIUS et Kerberos, université de Reims CHAPAGNE – ARDENNE, 2008

[48] Z.tahakourt, introduction aux systemes distribués, 2013.

## Résumé

L'objectif de cette thématique est d'optimiser la sécurité des systèmes distribués, afin d'assurer une confidentialité, une intégrité, et une authentification des biens matériels et moraux des personnes.

Suite à des analyses, recherches, études et des comparaisons, nous avons opté pour le système KERBEROS comme protocole d'authentification et le protocole hybride (RSA, AES) pour le partage de clés pour leur fiabilité, haute sécurité et large usage.

La phase de conception est réalisée en utilisant les diagrammes de modélisation UML, qui ont produit des modèles pour la phase de l'implémentation sous JAVA.



## Abstract

The objective of this thematic is to improve the security of distributed systems, to ensure confidentiality, integrity, and authentication of material and moral property of people.

After analyzes, research, studies and comparisons, we opted for the KERBEROS system as the authentication protocol and the hybrid protocol (RSA, AES) for sharing key for their reliability, high security and extensive use.

The design phase is performed using the UML diagrams, which have produced models for the phase of the implementation under JAVA.