République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de béjaia
Faculté des sciences exactes
Département informatique

Mémoire de fin de cycle
En vue d'obtention du diplôme de master professionnel en informatique
Option : Administration et Sécurité des Réseaux

Thème :

Proposition et Implémentation d'un Protocole D'authentification Unique

Mémoire soutenu le 27/06/2016 par :

M^lle KHERBACHE Meriem
M^lle LETAT Zina

Devant le jury composé de :

Président :	*M. FARAH Zoubeyr*	*M.C.B, U.A.M Béjaia*
Examinatrice :	*M^lle BENMERBI Samah*	*DOCTORANTE, U.A.M Béjaia*
Encadreur :	*M. DJEBARI Nabil*	*M.A.A, U.A.M Béjaia*
CoEncadreur :	*M. ALBANI Mourad*	*Chef département informatique, NAFTAL GPL Béjaia*

Année univerrsitaire 2015/2016

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

AES	**A**dvanced **E**ncyption **S**tandard
DES	**D**ata **E**ncyption **S**tandard
FTPS	**F**ile **T**ransfer **P**rotocol **S**ecurity
HDM	**H**omme **D**u **M**ilieu
HID	**H**uman **I**nterface **D**evice
HTTPS	**H**yper **T**ext **T**ransfer **P**protocol **S**ecure
IETF	**I**nternet **e**nginnering **T**ask **F**orce
IDS	**I**ntrusion **D**etection **S**ystem
IP	**I**nternet **P**rotocol
IPsec	**I**nternet **P**rotocol **S**ecure
JSP	**J**ava **S**erver **P**age
KDC	**K**ey **D**istriution **C**enter
MAC	**M**essage **A**uthentification **C**odes
MDC	**M**odification **D**etection **C**ode
MIT	**M**assachusetts **I**nstituteof **T**echnology
NIST	**N**ational **I**nstitute **S**tandards **T**echnologie
OSI	**O**pen **S**ystemes **I**nterconnection
OTP	**O**ne **T**ime **P**assword
PGP	**P**retty **G**ood **P**rivacy
PHP	**P H**ypertext **P**preprocessor
PIN	**P**ersonal **I**dentification **N**umber
PKI	**P**ublic **K**ey **I**nfrastructure
RSA	**R**ivest **S**hamir et **A**dlman
SHA	**S**ecure **H**ash **A**lgorithme
SSH	**S**ecure **SH**ell
SSL	**S**ecure **S**ocket **L**ayer
SSO	**S**single **S**sign- **O**n
TA	**T**rusted **A**uthority
TCP	**T**ransmission **C**ontrol **P**rotocol
TGS	**T**icket **G**ranting **S**ervice
TGT	**T**icket **G**ranting **T**icket
UIT	**U**nion **I**nternationale des **T**élécommunication
UML	**U**nified **M**odeling **L**anguage
UP	**U**nified **p**rocessus
XML	**EX**tensible **M**arkup **L**anguage

Introduction Générale

De nos jours, la sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs accèdent à leur comptes en toute sécurité .

L'universalité des protocoles HTTP et Client/serveur a depuis longtemps séduit les développeurs car les applications portées sur le web sont de plus en plus nombreuses et l'accès à ces applications est engendré par l'authentification, avec un nombre croissant d'applications à leur disposition, et donc avec autant de mots de passe à mémoriser, les utilisateurs et les administrateurs font de leur mieux : ils inscrivent ces codes secrets dans leur agenda papier, les notent sur des Post-it qu'ils collent autour de leur écran ou, plus simple, laissent leurs connexions ouvertes lorsqu'ils quittent leur poste de travail.

Après certaines recherches sur les différents standards d'authentification existants (Kerberos ,SSO...) et grâce à des informations collectées nous avons élaborée notre vision de l'application, son architecture, ainsi que les outils et les technologies utiles. D'où l'intérêt du notre projet de fin d'études consiste à réaliser un système d'authentification unique.

Dans ce contexte le mémoire est organisé sous forme de quatre chapitres, dans lesquels, nous présenterons l'organisme d'accueil Naftal Bejaia GPL qui nous a conduit à tirer la problématique et proposer une solution a cette dernière, par la suite, définir quelques protocoles d'authentification existants qui nous permettront de justifier notre proposition dans ce projet. En plus, nous présenterons le protocole proposé et nous insisterons sur son mécanisme de fonctionnement et nous détaillerons les différents diagrammes qui nous ont permis la réalisation de ce projet. Enfin, le dernier chapitre sera spécifiquement centré sur la réalisation et l'implémentation du système d'authentification proposé.

Nous terminerons notre mémoire par une conclusion générale et les perspectives concernant le renforcement de la sécurité, pour des raisons de temps, de moyens et de compatibilité entre les systèmes, ces perspectives n'ont pas fait parti de nos objectifs. Mais cela reste possible selon nos pronostics.

Chapitre 1

Présentation de l'entreprise d'accueil

1.1 Introduction

Pour mieux cerner la problématique soulevée. Nous nous sommes adressés a un des leaders industrielle de la région : il s'agit de NAFTAL ; et plus exactement l'unité "Gaz Pétrolier Liquéfié (GPL) de Bejaia". Cette enquête nous a permis de déceler les différentes difficultés reliées à notre thématique et de les côtoyer dans un contexte purement industriel.
Dans ce qui suit, on présentera notre entreprise d'accueil ainsi que les problèmes rencontrés en matière de réseau et sécurité puis on abordera les solutions proposées pour palier à ces derniers.

1.2 Historique

Jusqu'au 24 février 1971, date de nationalisation de ces derniers, SONATRACH apparue en plusieurs directions (18 entreprises).
Issue de SONATRACH, l'entreprise ERDP (entreprise de raffinage et de distribution du pétrole) a été créé par le décret 80/10 du 06 avril 1980, entré en activité le 01 janvier 1982, elle est chargée de l'industrie, de raffinage et de la distribution des produits pétroliers sous le sigle NAFTAL (NAFT est le pétrole et AL vient de l'Algérie).
Le 04 mars 1985, les anciens districts CLP (carburant, lubrifiant, pneumatique et bitumes) et GPL ont été regroupé sous le nom de UND (Unité NAFTAL de Distribution).
En 1987, l'activité de raffinage est séparée de l'activité de distribution, le raffinage est confié à NAFTEC (entreprise national de raffinage).
A partir de 1989, NAFTAL change de statut et devient SPA filiale à 100% de SONATRACH en 2000, l'entreprise est restructurée en divisions et zones : CLP, GPL et BITUMES.

1.3 Capacité, missions et objectifs de NAFTAL

1.3.1 Capacité de l'entreprise NAFTAL

— Le chiffre d'affaire : 15650000000 DA

— Transport : 3000 véhicules des distributions ; constitue de tracteurs routiers, de semi-remorques plateaux, de camions citernes, de camions plateaux, camions porte palettes et de moyens de transport tiers. 07 barges pour le soutrage des navires.

— Stockage

- Capacité de stockage totale

  * Carburants (terre, aviations, marines) :800000M3.
  * GPL conditionne : 3.8 millions B13.

- Capacité d'enfutage GPL : 1.2 millions tonnes/an.
- Capacité de formulations bitumes : 400000 tonnes/an.
- 67 centres de députés de distribution et de stockage de carburant lubrifiant et pneumatiques.
- 30 dépôts aviation marines.
- 40 usines d'emplissages GPL.
- 49 dépôts relais GPL.
- 15 unités bitumes.

— Distribution :

- 1755 stations-services dont 284 stations GPL/C (Sirghaz).
- 118 stations essences sans plomb et 14550 point de vente GPL

1.3.2 Missions de l'entreprise NAFTAL

NAFTAL a pour mission principale, la commercialisation et la distribution des produits pétroliers raffinent sur le marché national, notamment : les GPL, les carburants et lubrifiants y compris ceux destinés à l'aviation et à la marine, solvants, aromatiques, paraffines, bitumes et pneumatiques, elle intervient dans les domaines suivants :

- Enfutage GPL.
- Formulation de bitumes.
- Distribution, stockage et commercialisation des carburants, GPL, lubrifiants, bitume, pneumatiques, GPL/carburants et produit spéciaux.
- Transport des produits pétroliers.

1.3.3 Objectifs de l'entreprise NAFTAL

Pour remplir sa mission principale, NAFTAL s'est tracée les objectifs suivants :

- Organiser et développer la commercialisation et la distribution des produits pétroliers et dérivés.
- Stocker, transporter et/ou faire transporter tous produits pétroliers commercialisés sur 1e territoire national,développer les infrastructures de stockage et de distribution pour assurer une meilleure couverture de marché.
- Élaborer des plans en liaison avec les organismes concernés visant la couverture de marche national en produits pétroliers.
- Promouvoir, participer et veiller à l'application de la normalisation et du contrôle de la qualité des produits pétroliers en vue de planifier et d'assurer l'approvisionnement du marché.
- Développer et mettre en œuvre les actions visant l'utilisation optimale rationnelle des infrastructures et moyens.
- Participer et veiller à la mise en œuvre des actions visant le renforcement de l'intégrité économique.
- Concourir à la formation, au recyclage et au perfectionnement des travailleurs.
- Procéder à toute étude de marché de consommation.
- Assurer la maintenance des équipements et matériels relevant de son patrimoine.

1.4 Présentation générale du district GPL Bejaia

Le district GPL Bejaia se compose d'une direction de 6 départements, d'un centre remplissent de deux mini centres et d'un dépôt relais.

1.4.1 Missions

La zone GPL de Bejaia comprend géographiquement : Bejaia, Jijel et quelques communes de la wilaya de Bouira parmi ses principales missions : :

- L'approvisionnement, le stockage, le conditionnement et la distribution des produits GPL. Pour la satisfaction des besoins des clients affectés à sa zone.
- Assurance de la gestion des activités et des patrimoines de l'entreprise qui lui est rattachée.
- Application des politiques et procédures de l'entreprise en matière de gestion, de maintenance et de sécurité.
- Le bon suivi du développement des infrastructures.
- Réalisation des plans directeurs de distribution arrêtés par l'entreprise et assister les autres unités en matière de besoin.

1.4.2 Objectifs

• Organiser et développer les produits GPL.

• Connaître les différentes marches GPL et les besoins actuels et potentiels de sa clientèle.

• Satisfaire sa clientèle dans les meilleures conditions d'efficacité et de cout.

• Promouvoir participer et veiller à l'application de la normalisation et de contrôle de la qualité des produits.

• Organiser et coordonner les activités de programmation des approvisionnements, de ravitaillement et distribution des différents centres de stockages repartis à travers trois wilayas (Bejaia, Bouira, Jijel).

• Assurer l'approvisionnement et la commercialisation des produits GPL sur l'ensemble des trois wilayas.

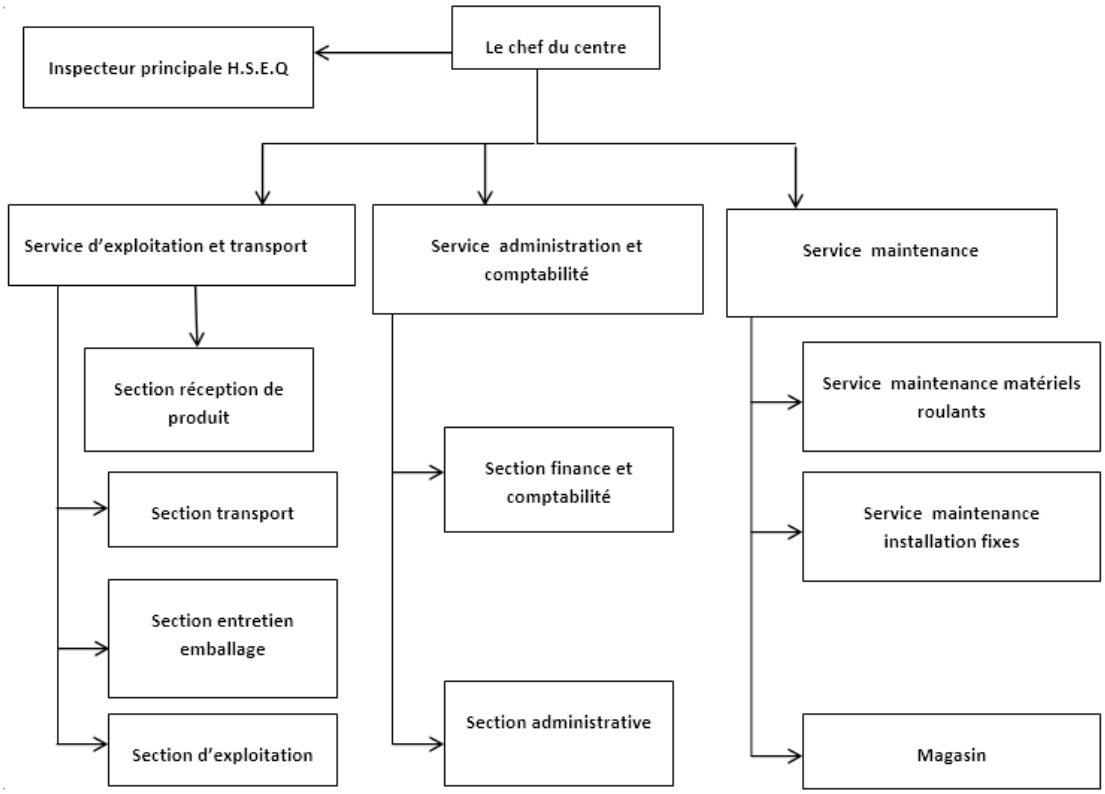

FIGURE 1.1 – l'organigramme NAFTAL GPL Bejaia.

1.5 Présentation du réseau NAFTAL Bejaia

1.5.1 Architecture de réseau NAFTAL Bejaia

Le réseau de ladite entreprise est de topologie étoile, il contient 62 ordinateurs connectés entre eux à l'aide des switchs. 1l est partagé en quatre zones :

•La zone 1 :elle recouvre le département informatique qui matérialise l'administration de réseau ;

•La zone 2 : attribuée au département finances ;

•La zone 3 ; reliant le département ressources humaines, les deux dernières sont reliées au département informatique par des câbles FTP, qui sont efficaces pour les petites distances ;

•La zone 4 ; correspondant au département commercial, elle est reliée avec celle de département informatique, par l'intermédiaire de fibre optique. Le choix est imposé par la grande distance les séparant (150m).

Toute l'administration du réseau se passe dans le département informatique, puisqu'il héberge le routeur d'accès vers l'internet. Ses bases de données et sa configuration sont réalisées sur un serveur réservé pour cet usage. Chaque zone dispose d'un switch, chargé de la communication des paquets jusqu'aux blocs constituants cette zone, ensuite ces switchs sont connectés aux terminaux dans chaque zone.

1.6 Les Equipements du réseau NAFTAL Bejaia

•62 ordinateurs ;

•Serveur Dell powerledge ;

•4 Switchs Cisco 24 ports ;

•4 Points d'accès (wifi) ;

•4 onduleurs apc 450va ;

•Un routeur vers l'extranet.

1.7 Problématique

La sécurisation d'un réseau informatique est de plus en plus épuisante surtout à l'heure actuelle où, le nombre d'applications et le niveau d'ouverture des systèmes croient exponentiellement.

Définir les personnes autorisées à accéder à un réseau, et leurs degrés de libertés respectives constitue la pièce maitresse de la sécurité, c'est ce que on désigne par « l'authentification des utilisateurs ». Ces derniers sont amenés à fournir en plusieurs reprises auprès de chacune de ces applications un couple login/mot de passe ce qui est aberrant. Menant ainsi à l'augmentation des phases d'authentification, chose qui peut conduire à l'usurpation de l'identité des utilisateurs, et l'interception des données échangées entre eux par une personne malveillante.

1.8 Solution proposée

À travers notre étude, nous proposeront une conception judicieuse d'un système d'authentification permettant à chacun des utilisateurs de s'authentifier uniquement une seule fois, pour avoir un accès libre à toutes les ressources (applications, données, ...etc.) qui lui sont autorisées ce qui

va permettre d'alléger le processus.

Ce système sera réalisé sur une plateforme « JAVA », ou nous allons implémenté le modèle client/serveur en utilisant « SOCKET ».

1.9 Conclusion

Dans ce chapitre, nous avons présenté l'entreprise NAFTAL, son architecture réseau et le problème d'authentification multiple. Dans ce qui suit, nous étudieront les différentes techniques et standards d'authentification qui feront office de sources d'inspiration.

Chapitre 2

Généralités sur la sécurité et l'authentification

2.1 Introduction

L'utilisation temporelle des applications par les entités, nécessite une sécurité informatique assurant la protection des interactions entre les différentes composantes, ainsi que la protection contre toute sorte d'usurpation d'identité.

2.2 Définition de la sécurité informatique

La sécurité informatique est l'ensemble des moyens mises en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles.

La sécurité a pour objectif : la protection des données et des ressources en mettant en place des mécanismes de contrôle permettant d'assurer le bon fonctionnement du système dans l'objectif d'assuré [17] :

•**Intimité et confidentialité :** empêcher la divulgation d'informations à des entités non-habilitées à les connaitre ;

•**Authentification :**
- Authentification d'une information : prouver qu'une information provient de la source annoncée (émetteur).
- Authentification d'une entité : prouver que l'identité est bien celle annoncée.

•**Intégrité des informations :** c'est assurer que les informations n'ont pas été altérées par des personnes non autorisées ;

•**La signature :** le moyen de lier l'information à une entité ;

•**La validation :** les moyens de fournir l'autorisation d'utiliser ou de manipuler les informations ;

•**Contrôle d'accès :** c'est limiter l'accès de ressources aux personnes privilégiées.

•**Certification :** c'est la probation de l'information par une entité de confiance

•**Réception :** c'est approuver la réception de l'information ;

•**Anonymat :** c'est cacher l'identité d'une entité impliquée dans un processus de communication ;

•**Non-répudiation :** empêcher le démenti d'engagement ou d'action précédents.

2.3 Vulnérabilités de système informatique

Dans un système informatique, les menaces peuvent toucher les composantes matérielles, logicielles ou informationnelles.
Il existe principalement deux types de menaces [11][17] :

2.3.1 Les menaces accidentelles (non-intentionnelles)

Dans cette catégorie sont repris les bugs logiciels et les pannes matérielles et autres défaillances incontrôlables sont présentés.

2.3.2 Les menaces intentionnelles (Passive - Active)

Ces types de menaces représentent l'action d'une personne désirant s'introduire dans le système et relever les informations qui peut se résumé comme suit :

•**Interruption :** problème de disponibilité des données.

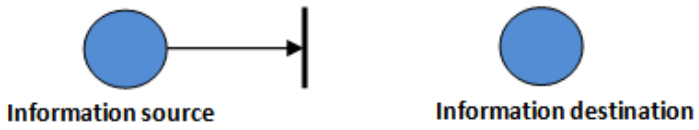

FIGURE 2.1 – Menace intentionnelles : interruption.

•**Modification :** Problème d'intégrité des données.

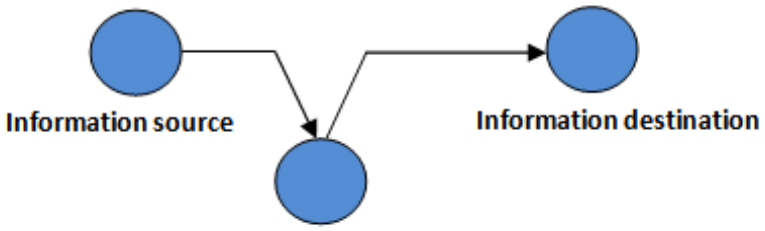

FIGURE 2.2 – Menace intentionnelles : modification.

•**Interception :** Problème lié à la confidentialité des données.

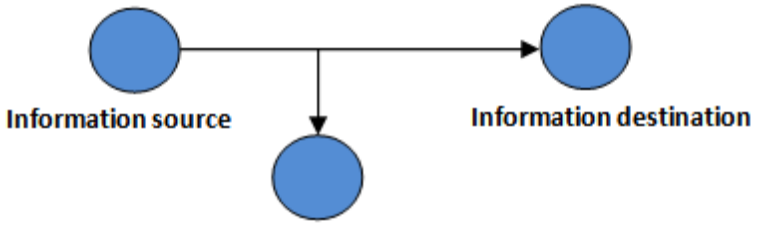

FIGURE 2.3 – Menace intentionnelles : interception.

•**Fabrication :** Problème d'authentification.

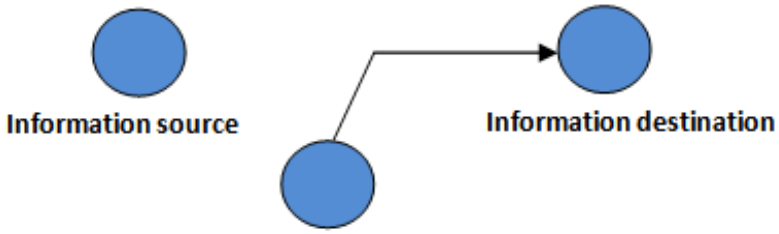

FIGURE 2.4 – Menace intentionnelles : fabrication.

2.4 Principales technologies de défense

Il existe plusieurs moyens pour se protéger contre les menaces de sécurité potentielle [21][04] :

•Authentification : vérifier la véracité des utilisateurs, du réseau et des documents.

•Autorisation : L'autorisation permet de déterminer ce que l'entité a le droit de faire.

•Cryptographie : pour la confidentialité des informations et la signature électronique.

•Contrôles d'accès aux ressources.

•Firewalls : filtrage des trames entre le réseau externe et le réseau interne.

•Audit : études des fichiers de log pour repérer des anomalies.

• Programmes de tests de vulnérabilité et d'erreurs de configuration.

•Détection d'intrusions : détection des comportements anormaux d'un utilisateur ou des attaques connue : développement d'IDS très en vogue et d'actualité.

2.5 Mécanismes cryptographiques

2.5.1 Chiffrement

Le chiffrement est un processus de transformation des informations de façon à les rendre invisibles à toute personne autre que le destinataire. Le déchiffrement est le processus inverse du chiffrement, il sert à transformer les informations de façon à les rendre à nouveau visible.

Un algorithme de cryptographie, également appelé chiffre, est une fonction mathématique utilisée pour le chiffrement ou le déchiffrement. Dans la plupart des cas, deux fonctions complémentaires sont employées, une pour le chiffrement et la seconde pour le déchiffrement.

Dans la cryptographie moderne, la possibilité de conserver des informations secrètes ne se fait pas à partir de l'algorithme de cryptographie seul, mais avec un nombre appelé clef afin de produire un résultat chiffré, ou pour déchiffrer des informations précédemment chiffrées.[25][17]

a) Chiffrement symétrique

Chiffrement symétrique, algorithme à clé secrète, cryptographie à clé secrète : tous ces termes désignent la même technique. Ici, l'émetteur et le destinataire du message disposent de la même clé secrète «k ». L'émetteur va utiliser cette clé secrète k pour chiffrer le message « M ». Le message chiffré est « C ». Le récepteur utilisera cette même clé secrète pour déchiffrer C, et retrouver ainsi le message en clair M.
Ce type de chiffrement a l'avantage d'être rapide, mais présente un problème de sécurité d'échange de la clé partagée, en plus le nombre nécessaire pour faire communiquer N interlocuteur est très grand (N*(N-1)/2). Parmi les algorithmes de chiffrement symétrique les plus connus, on trouve DES, 3 DES, et AES[26][25].

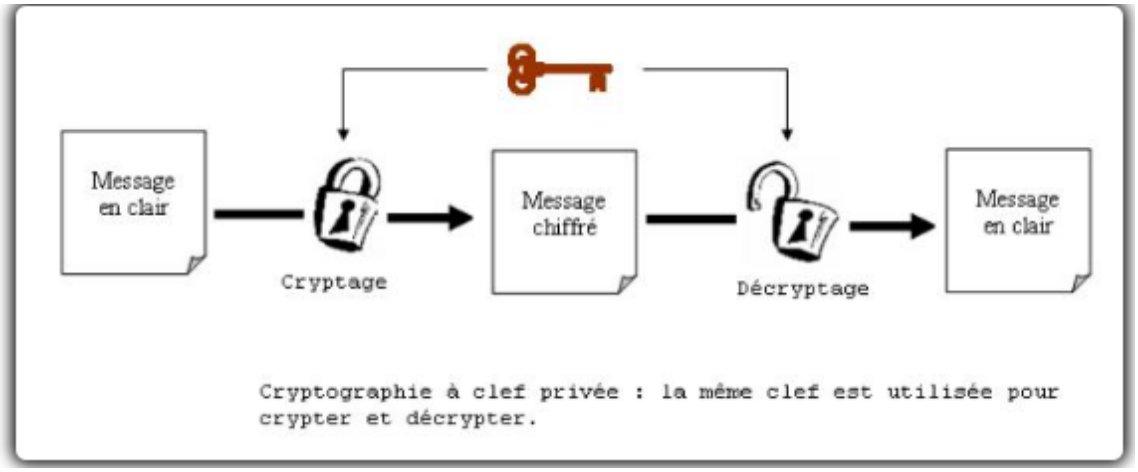

FIGURE 2.5 – Chiffrement symétrique [5].

b) Chiffrement asymétrique

Chiffrement asymétrique, algorithme à clé publique, cryptographie à clé publique : tous ces termes désignent la même technique. Cette technique repose sur le fait que : la clé de chiffrement est différente de la clé de déchiffrement. La clé de chiffrement, qui est appelée clé privée, est gardée secrète, tandis que la clé de déchiffrement, qui est appelée clé publique, est destinée à être divulguée. Les algorithmes à clé publique permettent d'assurer la confidentialité d'un message. Dans ce cas, la procédure à suivre est la suivante :
- L'émetteur doit récupérer du destinataire, la clé publique k1 avec laquelle il va chiffrer le message en clair M. Le message chiffré résultant ; C sera envoyé au destinataire ;
- Le destinataire pourra ainsi déchiffrer ce message, et ce grâce a sa clé privée k2. il retrouve le message en clair M d'origine.
Ce type de chiffrement résout le problème rencontré dans le cas symétrique, car il permet une communication sans échange de clé.RSA est un exemple d'algorithme de chiffrement asymétrique. La combinaison des deux techniques (chiffrement hybride) permet a la fois d'obtenir la rapidité de chiffrement et de résoudre le problème de la clé secrète. Diffie Hellman est un exemple d'algorithme de chiffrement hybride[26][27].

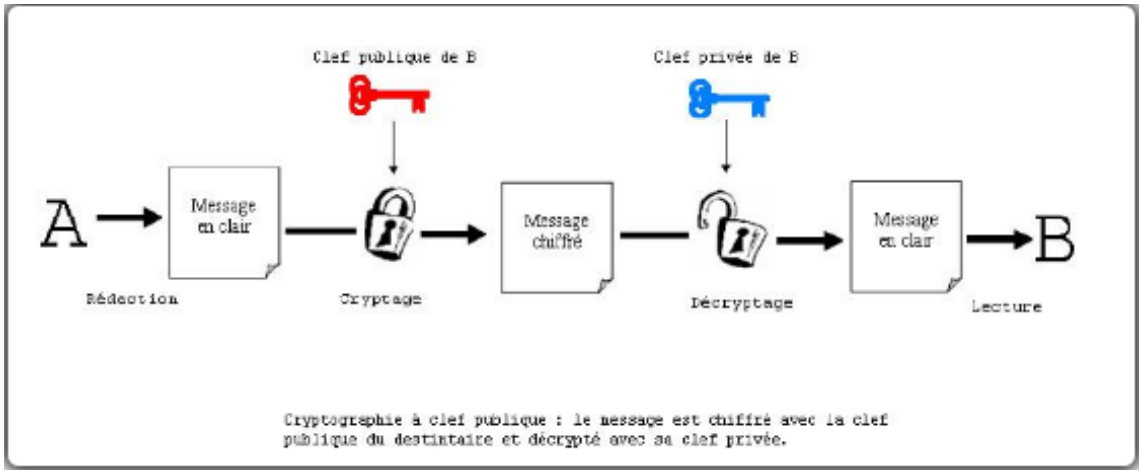

FIGURE 2.6 – Chiffrement asymétrique [5].

2.5.2 Fonction de hachage

Une fonction à sens unique qui a pour entrée une donnée de taille arbitraire appelée message, dont elle associe une valeur de sortie appelée valeur de hachage, empreinte ou haché.Une bonne fonction de hachage possède les propriétés suivantes [26] :

- La valeur de hachage d'un message se calcule très rapidement ;
- Difficile de calculer le message à partir de sa valeur de hachage ;
- Impossible de modifier un message sans changer sa valeur de hachage ;
- Impossible de trouver deux messages différents ayant la même valeur de hachage.

Les fonctions de hachage cryptographiques ont de nombreuses applications en sécurité informatique, notamment dans les signatures numériques, les codes d'authentification de message et les

autres formes d'authentification. Les fonctions Secure Hash Algorithme du NIST sont des exemples de fonctions de hachage cryptographiques. Il existe deux types de fonction de hachage : -Codes détection de modification (MDC : Modification Détection Code) : utilise des fonctions de hachage sans clé pour vérifier l'intégrité des données . -Codes d'authentification de messages (MAC : Message Authentification Codes) : utilise des fonctions de hachage avec clé pour vérifier l'intégrité d'une donnée et authentifier la source de cette même donnée[20].

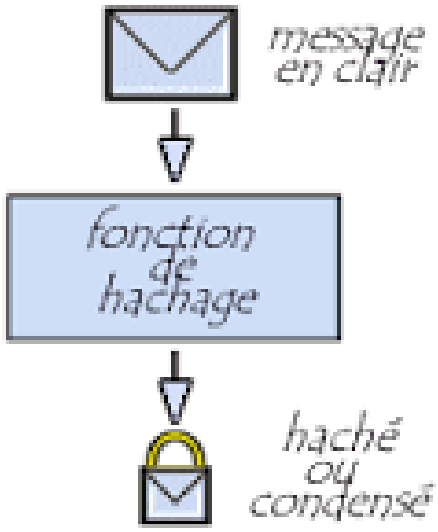

FIGURE 2.7 – Fonction de hachage [5].

2.5.3 Message authentification code (MAC)

Un code d'authentification de message (MAC, Message Authentification Code) est un code accompagnant des données dans le but d'assurer l'intégrité de ces dernières, en permettant de vérifier qu'elles n'ont subies aucune modification, après une transmission par exemple.

Le concept est relativement semblable aux fonctions de hachage. Il s'agit ici aussi d'algorithmes qui crée un petit bloc authentificateur de taille fixe. La grande différence est que ce bloc authentificateur ne se base plus uniquement sur le message, mais également sur une clé secrète.

Tout comme les fonctions de hachage, les MAC n'ont pas besoin d'être réversibles. En effet, le récepteur exécutera le même calcul sur le message et le comparera avec le MAC reçu.

Le MAC assure non seulement une fonction de vérification de l'intégrité du message, comme le permettrait une simple fonction de hachage mais de plus authentifie l'expéditeur, détenteur de la clé secrète. Il peut également être employé comme un chiffrement supplémentaire (rare) et peut être calculé avant ou après le chiffrement principal, bien qu'il soit généralement conseillé de le faire avant[27].

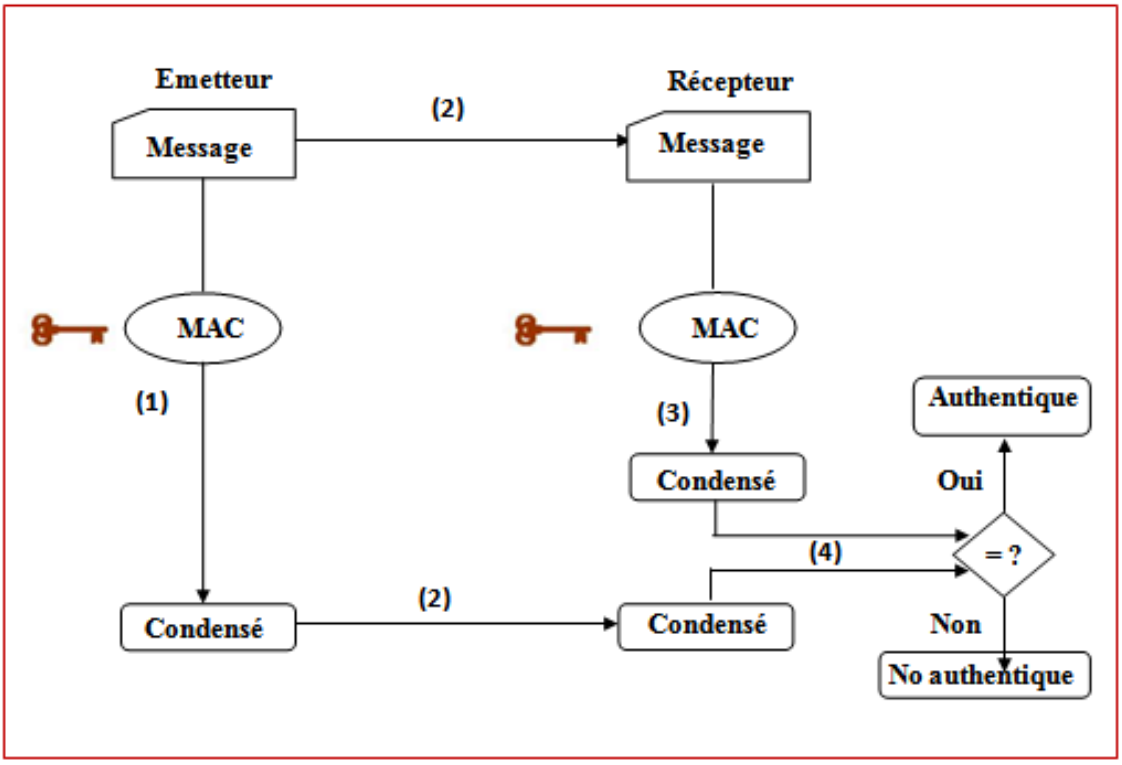

FIGURE 2.8 – Message authentification code.

Si deux interlocuteurs désirant authentifier le message échangé entre eux, doivent partager une clé secrète k, l'émetteur calcule le condensé h=MAC (k, m) du message (1), qui sera ajouté au message à envoyer(2). A l'arrivée, le récepteur vérifie l'origine du message reçu comme suit : Il calcule le condensé du message reçu en utilisant la même clé k (3), puis, le compare avec le condensé reçu (4). Si les deux condensés sont égaux, le message est dit authentique sinon le message reçu a été changé ou a été fabriqué par un autre expéditeur[20].

2.5.4 Signature numérique

La signature numérique fournis les services d'intégrité des données, d'authentification de l'origine des données et de non-répudiation. La façon la plus simple de signer un message est d'utiliser la cryptographie asymétrique pour le chiffrer en utilisant sa clef privée : seul le possesseur de cette clef peut générer la signature et toute personne ayant accès à la clef publique correspondante peut la vérifier[25].

La figure ci-dessous (2.9) illustre les différentes étapes pour assurer le no répudiation avec l'origine de la donnée transmise.

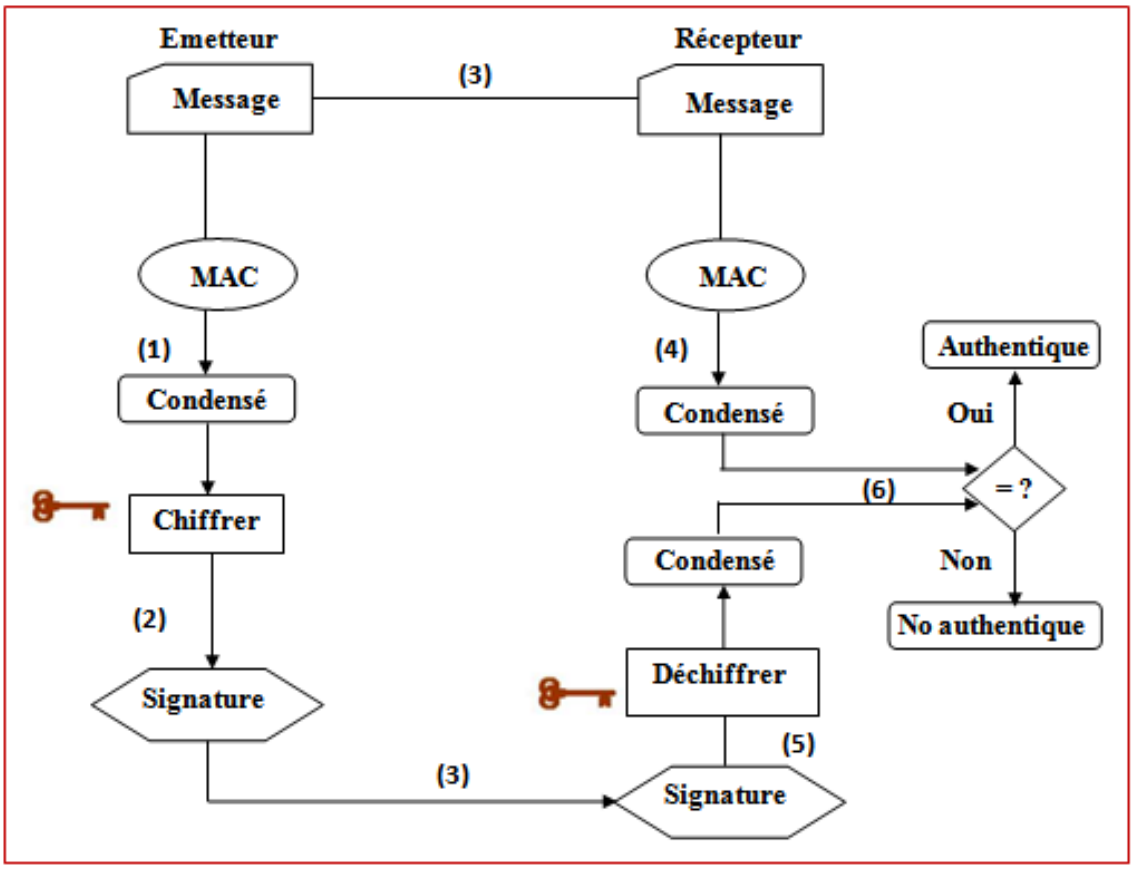

FIGURE 2.9 – Signature numérique.

— L'émetteur calcule le condensé du message en utilisant une fonction de hachage (1) , puis chiffre ce dernier avec sa clé privée (2). Cela constitue la signature du message. Il procède ensuite à l'envoi du message et de la signature (3).

— A la réception, le destinataire calcule le condensé du message reçu en utilisant la même fonction de hachage (4), déchiffre la signature en utilisant la clé publique de l'émetteur (5). Les deux condensés ainsi obtenus sont comparés (6). S'ils concordent, la signature est de ce fait vérifie et le destinataire peut alors avoir la certitude que le message a été envoyé par l'émetteur et n'a pas été altéré[20].

2.6 Politique de sécurité

La mise en œuvre d'une politique de sécurité globale, est assez pénible comme tache, essentiellement pour raison de multiplicité des aspects à considérer. Une politique de sécurité se caractérise par : les niveaux d'intervention, ses objectifs et enfin les outils utilisés.

Chaque aspect doit être pris en compte différemment, de façon à atteindre les objectifs désirés, toute en utilisant de façon agencée les différents outils mis à disposition.

Nous abordons les différents aspects d'une politique de sécurité, avant de définir les objectifs visés, et nous verrons ensuite les outils dédiés à l'application de celle-ci. Une politique de sécurité

s'élabore à plusieurs niveaux [20] :

• sécuriser l'accès aux données de façon logicielle (authentification, contrôle d'intégrité).

• sécuriser l'accès physique aux données : serveurs placés dans des salles blindées avec badge d'accès.

• Un aspect très important pour assurer la sécurité des données d'une entreprise est de sensibiliser les utilisateurs aux notions de sécurité, de façon à limiter les comportements à risque : si tout le monde peut accéder aux salles de serveurs, peut importe qu'elles soient sécurisées.

• De même, si les utilisateurs laissent leur mot de passe écrit à côté de leur PC, son utilité est limitée.

• Enfin, il est essentiel pour un responsable de sécurité de s'informer continuellement, des nouvelles attaques existantes, des outils disponibles de façon à pouvoir maintenir à jour son système de sécurité et à combler les brèches de sécurité qui pourraient exister.

2.7 Protocoles de communication sécurisés

2.7.1 SSL (Secure Socket Layer)

Le logiciel SSL développé par Netscape, c'est un logiciel qui a pour rôle de chiffrer les messages échangés entre un navigateur et le serveur interrogé.Il permet d'assurer[08][23] :

• Confidentialité : Il est impossible d'espionner les informations échangées.

•Intégrité : Il est impossible de truquer les informations échangées.

•Authentification : Il permet de s'assurer de l'identité du programme, de la personne ou de l'entreprise avec laquelle on communique.

SSL est un complément à TCP/IP et permet (potentiellement) de sécuriser n'importe quel protocole ou programme utilisant TCP/IP.
Certains protocoles ont été spécialement modifiés pour supporter SSL :

• HTTPS : c'est HTTP+SSL. Ce protocole est inclus pratiquement dans tous les navigateurs, et vous permet de consulter vos comptes bancaires par le web de façon sécurisée.

• FTPS est une extension de FTP (File Transfer Protocol) utilisant SSL.

• SSH (Secure Shell) : c'est une sorte de Telnet (ou login) sécurisé. Cela permet de se connecter à un ordinateur distant de façon sûre et d'avoir une ligne de commande. SSH possède des extensions pour sécuriser d'autres protocoles (FTP, POP3 ou même X Windows).

2.7.2 IPSec (Internet Protocol Security)

Défini par l'IETF comme un cadre de standards ouverts pour assurer des communications privées et protégées sur des réseaux IP, par l'utilisation des services de sécurité cryptographiques , l'IPsec est un ensemble de protocoles utilisant des algorithmes permettant le transport de données sécurisées sur un réseau IP. IPsec se différencie des standards de sécurité antérieurs en n'étant pas limité à une seule méthode d'authentification ou d'algorithme et c'est la raison pour laquelle il est considéré comme un cadre de standards ouverts. De plus, IPsec opère à la couche réseau (couche 3 du modèle OSI) contrairement aux standards antérieurs qui opéraient à la couche application (couche 7 du modèle OSI), ce qui le rend indépendant des applications, et veut dire que les utilisateurs n'ont pas besoin de configurer chaque application aux standards IPSec[08].

2.8 Définition de l'authentification

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne ou d'un ordinateur) afin de lui autoriser ou pas, l'accès à des ressources bien spécifiées[03].

L'authentification permet donc, de valider l'authenticité de l'entité en question. La notion d'authentification s'oppose à celle de l'identification d'une personne physique ou morale (dirigeant et/ou toute personne y autorisée). Cette distinction est importante puisque par abus de langage, on parle d'authentification alors qu'il s'agit d'identification. Lorsqu'une personne présente sa pièce d'identité lors d'un contrôle, elle est identifiée grâce à un document officiel, mais n'est pas authentifiée, car le lien entre la pièce d'identité et la personne n'est pas établie de façon indiscutable, irrévocable et reconnue par les tribunaux en cas de litige[04].

Par opposition, lorsqu'une personne est authentifiée, cette authentification doit-être apportée par un tiers de confiance, ou par une autorité de certification délivrant une preuve au sens juridique reconnue devant les tribunaux (ex : la signature électronique de la carte bancaire)[20].

2.9 Techniques d'authentification

Pour s'authentifier, un utilisateur fournit en général au moins 2 éléments[21] :

- son identifiant qui permet son identification.
- un ou plusieurs éléments permettant d'assurer l'authentification elle-même.

Nous retrouvons ainsi ces éléments sous des formes diverses. Voici les plus largement utilisés :

2.9.1 L'identifiant et le mot de passe

L'identifiant et le mot de passe sont le couple d'authentification le plus connu. Simple, robuste, voire même rustique, son plus gros défaut est que le niveau de sécurité dépend directement de la complexité du mot de passe. Des mots de passes simples sont faibles, et des mots de passes trop complexes conduisent les utilisateurs à mettre en œuvre des stratégies de contournement pour les gérer : Post-it, liste dans un fichier Excel ou dans le Smartphone,....

2.9.2 L'identifiant et le mot de passe OTP (One-Time Password)

L'OTP permet de sécuriser l'utilisation du mot de passe sur le réseau. En effet avec un système OTP, l'utilisateur possède un calculateur spécialisé qui lui fournit à la demande un mot de passe. Ce mot de passe est valide pendant une durée limitée seulement, et pour une seule utilisation. Cette solution est en général mise en œuvre pour le processus d'authentification initiale pour les accès externes via IP/VPN.

2.9.3 Les certificats PKI sur carte a puce ou clef USB

Les certificats X.509 mettent en œuvre une technologie avancée de chiffrement qui permet de chiffrer ou signer des messages sans avoir à partager de secret. L'identifiant est un certificat publique qui est signé et donc garanti par une autorité de certification reconnue. L'utilisateur doit fournir un secret pour pouvoir utiliser les différents éléments cryptographiques : « le code PIN de sa carte ou de sa clef USB». Cette solution est en général mise en œuvre pour le processus d'authentification initiale ou pour les connexions aux applications Web ou de messagerie.

2.9.4 Clef « confidentiel défense »

Il s'agit d'une déclinaison particulière de l'exemple précédent. C'est en général une clef multi-fonctions : stockage de certificat X.509, stockage de données, ressource cryptographique etc.. . .

2.9.5 L'identifiant et le mot de passe sur une carte a puce

Le stockage de l'identifiant et du mot de passe sur une carte à puce permet de compléter la sécurisation du processus d'authentification. Le mot de passe peut ainsi être très complexe et changé régulièrement de manière automatique et aléatoire. Sans la carte, et sans son code PIN, il n'y a plus d'accès au mot de passe. Cette solution est généralement mise en œuvre pour le processus d'authentification initiale.

2.9.6 Biométrie

L'authentification par biométrie s'appuie sur la vérification d'un élément du corps de l'utilisateur (le plus souvent l'empreinte digitale). Elle peut s'appuyer sur un serveur central, sur le poste ou sur une carte à puce pour stocker les données biométriques de l'utilisateur. Cette solution est en général mise en œuvre pour le processus d'authentification initiale et/ou pour protéger l'accès à des applications très sensibles.

2.9.7 L'identification sans contact

Le RFID est une technologie qui aujourd'hui se déploie dans les projets 'identification/autorisation. Une puce RFID est encastrée dans un badge et porte un numéro d'identification. Ce numéro est ensuite associé à un utilisateur dans un système informatique. A la base c'est une technologie d'identification qui peut, en étant couplée à un mot de passe fourni par l'utilisateur par exemple, être utilisé dans des procédures d'authentification. Il existe 2 déclinaisons de cette technologie :

- **Le RFID passif ou HID,** qui suppose que la carte ne possède pas d'alimentation propre. La carte est alimentée lors de la lecture par un champ électromagnétique généré

par le lecteur. Ce système est communément utilisé pour le contrôle d'accès physique par badge ou le paiement au restaurant d'entreprise. La détection d'une carte HID se fait à quelque centimètre.

•**Le RFID actif** s'appuie sur les protocoles de communication RFID mais associe à la carte une alimentation propre. Cette alimentation permet une détection de la carte à plus longue portée (par exemple dès l'entrée dans une salle ou un bureau). L'intérêt principal du RFID actif est de permettre un constat d'absence pour les postes de travail dans des zones accessibles au public

2.10 Types d'authentification

2.10.1 Authentification simple

Soit deux entités A et B, A veut prouver sont identité à B, et B à son tour vérifie l'identité de A.

a) Protocole à chiffrement symétrique

Dans ce cas de figure, A présente à B un message d'authentification chiffré avec la clé secrète qu'ils partagent, cette dernière constitue la preuve que le message a bien été envoyé par A. la vérification du message d'authentification par B varie selon sa nature ou son contenu :
- Protocole avec horodatage : le message d'authentification est composé, comme le montre la figure, de [04][03] :
-Une partie en claire : l'identité de A.
-Une partie chiffré par Kab : contient l'instantané "ta" de l'envoi du message accompagné par l'identité de B.

Lors de la réception du message, B le déchiffre avec la clé Kab et vérifie si "ta" appartient a l'intervalle d'acceptation des estampilles, si c'est le cas, l'authentification est réussie.

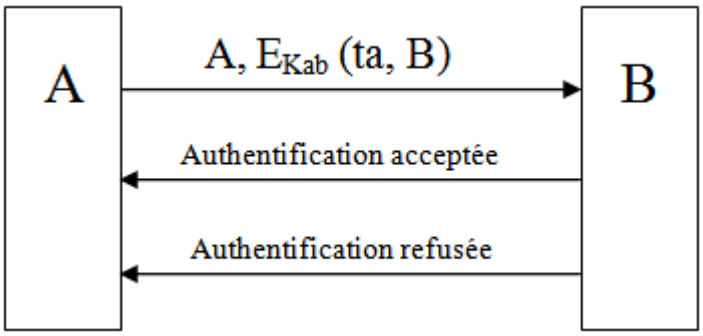

FIGURE 2.10 – Protocole d'authentification simple symétrique : avec horodatage.

- Protocole avec numéro de séquence : la différence avec l'horodatage est qu'un numéro de séquence est utilisé a la place de l'instantané. Après le déchiffrement du message d'authentification, B modifie son compteur locale des numéros de séquence (en l'incrémentant) commun avec A, s'il ne retrouve pas la même valeur, le message est rejeté et l'authentification échoue[20][03].

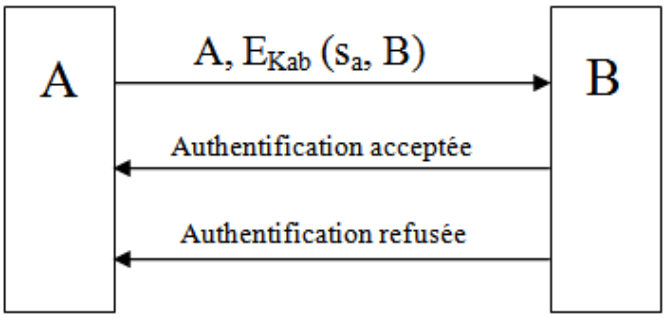

FIGURE 2.11 – Protocole d'authentification symétrique : avec numéro de séquence.

) Protocole à chiffrement asymétrique

Les techniques d'authentification illustrées ici sont basées sur la signature numérique et le message d'authentification est constitué de :
-Une partie en clair : l'identité de A, un certificat qui permet à A de prouver son identité.
-Une partie signée par A avec sa clé privée.
- Protocole avec horodatage : B extrait la clé publique de A de certificat CertA, déchiffre le message et compare le résultat avec le hache de (ta, B) qu'il recalcule. Si les deux correspondent, l'identité de A est vérifiée[14].

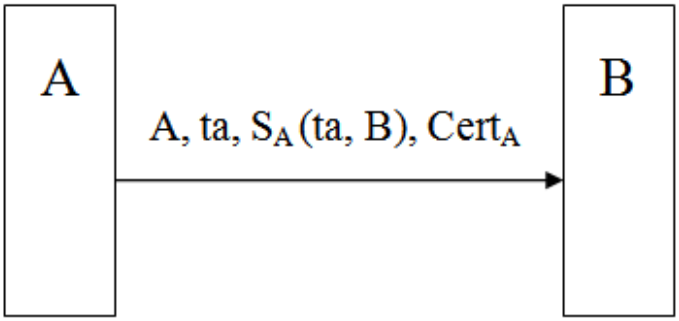

FIGURE 2.12 – Protocole d'authentification simple asymétrique : avec horodatage.

- Protocole avec numéro de séquence : la différence avec l'horodatage est qu'un numéro de séquence est utilisé a la place de l'instantané. Après le déchiffrement du message d'authentification, B modifie son compteur locale des numéros de séquence (en l'incrémentant par exemple) commun avec A, s'il ne retrouve pas la même valeur, le message est rejeté et l'authentification échoue[14].

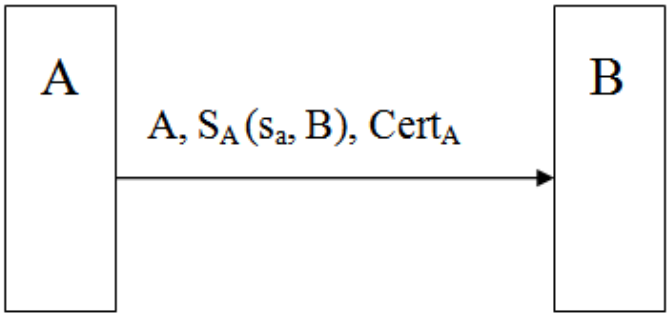

FIGURE 2.13 – Protocole d'authentification simple asymétrique : avec numéro de séquence.

2.10.2 Authentification mutuelle

Le protocole de base pour ce modèle de communication est le protocole challenge/response combiné à la technique des nombres aléatoires. Les autres paramètres tel que l'estampille ou le numéro de séquence peuvent être également ajoutés pour renforcer l'authentification.

Les deux figures suivantes illustrent une session d'authentification entre A et B à base du chiffrement symétrique figure (2.14) et asymétrique figure(2.15)[20].

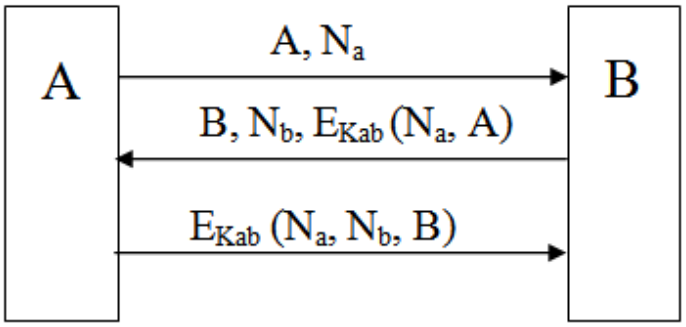

FIGURE 2.14 – Protocole d'authentification mutuelle : chiffrement symétrique.

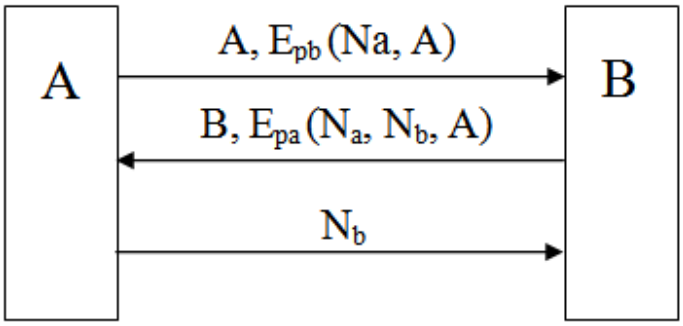

FIGURE 2.15 – Protocole d'authentification mutuelle : chiffrement asymétrique.

L'ajout de l'identifiant de B au message d'authentification, permet d'éliminer tout rejeu de ce message par un intru, qui voudrait se faire passer pour B auprès de A. dans ce cas,l'entité A pourra déchiffrer le message grâce a la clé Kab, et pourra tout de suite déduire qu'il s'agit d'un rejeu d'un message d'authentification qu'elle a précédemment envoyé[20].

2.11 Les différents standards et outils d'authentification

2.11.1 Définition de système kerberos

Le protocole Kerberos[12] a été normalisé par l'IETF dans les RFC 1510 (septembre 1993) et 1964 (juin 1996). Le nom « Kerberos » provient de la mythologie grecque et correspond au nom du chien (en français « Cerbère ») protégeant l'accès aux portes d'Hadès.

Développé par le MIT, Kerberos est un système d'authentification sécurisé et centralisé, a tierce personne de confiance (ou TA pour Trusted Authority) s'inspirant de celui de Needham Schroeder, qui utilise une cryptographie à clés symétriques conçu pour les réseaux TCP/IP.

Le système kerberos se compose d'un serveur d'authentification AS, qui permet à l'utilisateur de s'authentifier une seule fois pendant la durée d'une session ; d'un serveur de tickets TGS, qui génère le ticket de service demandé par l'utilisateur pour se connecter au service demandé. Et enfin

d'un centre de distribution de clés, qui assure la liaison entre les deux serveurs AS et TGS.

Le fonctionnement[05]

• À l'aide d'un mot de passe ou d'une carte à puce, l'utilisateur du système client s'authentifie auprès du KDC.

• Le KDC émet un ticket d'accord de ticket (TGT, Ticket Granting Ticket), approprié au client. Grace a ce TGT, le système client accède au service d'accord de ticket (TGS, Ticket Granting Service), qui fait partie du mécanisme d'authentification Kerberos V5 du contrôleur de domaine.

• Ensuite, le TGS émet un ticket de service à l'intention du client.

• Le client présente ce ticket de service au service réseau demandé. Ce ticket de service prouve à la fois l'identité de l'utilisateur au service et l'identité du service à l'utilisateur.

Les avantages[31]

• Transmission des mots de passe cryptés à travers le réseau ;

• Un espion sur le réseau ne doit pas pouvoir obtenir l'information nécessaire pour se faire passer pour un utilisateur ;

• Kerberos doit pouvoir se reposer sur une architecture de serveur distribuée avec des systèmes interchangeable ;

Les inconvénients[09]

• Tous les services du réseau doivent être « Kerberisé », c'est-à-dire compatible avec le protocole Kerberos, sinon aucune authentification ne sera possible.

• Il faut que toutes les machines du réseau soient synchronisées, Si l'on peut tromper un ordinateur quand 'a l'heure réelle, alors les anciens authentifiants peuvent être rejoués.

• Kerberos introduit un SPOF (Single Point Of Failure) dans le réseau. Si le serveur Kerberos tombe, il n'y aura plus aucun accès aux différents services du réseau. La machine serveur de Kerberos doit être parfaitement sure.

• Si l'AS 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.

2.11.2 Systéme SSO(Single Sign-On)

Le SSO[18] (Single Sign on, authentification unique et une seule fois) est un système, qui permet à l'utilisateur de saisir son mot de passe une seule fois pour accéder à toutes les applications web, améliorant ainsi l'ergonomie d'accès aux applications et la sécurité du système d'information,

tout en limitant la circulation des mots de passe.

Lors de l'unique saisie du mot de passe, l'identifiant de l'utilisateur et ses attributs sont transmis vers les différentes applications.

Certains logiciels SSO assurent la fermeture des sessions applicatives de l'utilisateur lorsqu'il se déconnecte.

Le fonctionnement[18]

Il se déroule comme suit :

- l'authentification est assurée par un serveur dédié, transparent pour l'application (pas de recueil du couple identifiant + mot de passe).
- Des tickets sont ensuite délivrés au client (maintien de la session d'authentification), et aux applications (transmission de l'identité de l'utilisateur).
- Etablissement des relations de confiance entre les applications et le serveur d'authentification (cryptographie symétrique ou asymétriques, certificats X509).

Les avantages[12]

Le système offre :

- Simplicité de l'authentification de l'utilisateur : Cela lui évite de retenir ou de taper un mot de passe, à chaque fois qu'il lance une application ;
- Sécurité renforcée : Toute la sécurité est assurée par un serveur spécialise et non par des applications ;
- Cohérence dans la gestion des comptes utilisateurs (allocation, expiration, mises à jour), Ergonomie utilisateur, rationalisation dans la gestion des comptes.

Les inconvénients[12]

- Une seule authentification pour accéder a toutes les ressources, donc en cas d'obtention d'un mot de passe par une personne mal intentionnée, elle aura l'accès à tous les services réseaux.

2.11.3 Le Certificat X.509

La X.509[16] est une norme de cryptographie asymétrique, dédiée aux infrastructures à clés publiques (PKI), définie par l'Union internationale des télécommunications (UIT).

La X.509 établit entre autres un format standard de certificat électronique et un algorithme pour la validation de chemin de certification. Ladite norme fait également l'objet de nombreuses RFC de l'IETF.
Fut créée en 1988 dans le cadre de la norme X.500. Elle repose sur un système hiérarchique d'autorités de certification, à l'inverse des réseaux de confiance (telle la toile de confiance OpenPGP), où n'importe qui peut signer les certificats d'autrui.

Un certificat est utilisé principalement, pour identifier et authentifier une personne physique ou morale, mais aussi pour chiffrer des échanges. Il est signé par un tiers de confiance qui atteste un lien entre l'identité physique et l'entité numérique.

Le fonctionnement[28]

• L'utilisateur récupère le certificat crée par l'autorité de certification.

•Suite à chaque authentification l'utilisateur doit présenter le certificat.

Les avantages[23]

• Une implication minime des utilisateurs ;

• Aucun équipement supplémentaire n'est nécessaire (Aucun jeton d'authentification à ni distribuer ou a gérer) ;

• faciles à émettre ou révoquer selon le renouvellement du personnel ;

• Solution évolutive s'adaptant à la croissance de l'entreprise ;

• Solution idéale pour les employés travaillant à distance ou sur leurs appareils portables.

Les inconvénients[23]

Il présente des inconvénients en matière de :

• La sécurité : si un serveur hébergeant un certificat est compromis, tous les autres serveurs utilisant ce certificat sont en danger (même clef privée) ;

• La gestion : si un certificat doit être révoqué, il faudra le changer sur tous les serveurs qui l'utilisent ;

• La compatibilité : pour éviter les soucis, il faut considérer que l'étoile ne remplace qu'un seul niveau de domaine (pas de correspondance sur le point).

2.11.4 Infrastructure de clé publique PKI

PKI[02] (Public Key Infrastructure) est un système, qui permet de gérer des listes importantes de clefs publiques et d'en assurer la fiabilité. Elle offre un cadre global, permettant d'installer des éléments de sécurité tels que la confidentialité, l'authentification, l'intégrité et la non-répudiation, tant au sein de l'entreprise que lors d'échanges d'information avec l'extérieur.
Une infrastructure PKI fournit donc quatre services principaux :
-La fabrication des clés.
- La certification des clés publiques et publication des certificats.
- La révocation des certificats.
- La gestion de la fonction de certification.

Le fonctionnement[12]

Dans une infrastructure à clé publique ; pour obtenir un certificat numérique, l'utilisateur fait une demande auprès de l'autorité d'enregistrement. Celle-ci génère un couple de clé (clé publique, clé privée), envoie la clé privée au client, certifie la clé publique et appose sa signature sur le certificat en appliquant une procédure bien définie.

Les avantages[24]

Le système assure :

- Une véracité des informations communiquées par l'utilisateur lors de la demande de certificat ;
- Une délivrance du certificat au bon propriétaire ;
- La validité d'un certificat à un instant donné.

Les inconvénients[24]

- La méthode est relativement lente ;
- La cryptographie à clé publique peut être vulnérable à usurpation d'identité.

2.11.5 Le système Pretty Good Privacy(PGP)

PGP (Pretty Good Privacy) est un crypto-système (système de chiffrement) inventé par Philip Zimmermann, un analyste informaticien. Philip Zimmermann a travaillé de 1984 à 1991, sur un programme permettant de faire fonctionner RSA sur des ordinateurs personnels (PGP). Cependant, étant donné que celui-ci utilisait RSA sans l'accord de ses auteurs, cela lui a valu des procès pendant 3 ans, il est donc vendu environ 150$ depuis 1993. Il est très rapide et sûr ce qui le rend quasiment impossible à cryptanalyse[22].

Le fonctionnement[12]

PGP est un système de cryptographie hybride, utilisant une combinaison des fonctionnalités de la cryptographie à clé publique et de la cryptographie symétrique.
La plupart des cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement. La compression réduit ces modèles dans le texte en clair, améliorant par conséquent considérablement la résistance à la cryptanalyse. Ensuite, l'opération de chiffrement se fait principalement en deux étapes :

- PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
- PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

L'opération de décryptage se fait également en deux étapes :

- PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée ;
- PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

Les avantages[29]

PGP possède plusieurs avantages :

- La rapidité : le message est chiffré par un cryptage symétrique. La clef IDEA est chiffrée de façon asymétrique. Toutefois le volume de données que représente cette clef est négligeable par rapport au volume de données que représente le message. Par conséquent, le temps de chiffrage global est proche de celui d'un système symétrique ;
- Une plus haute sécurité qu'un système à clef symétrique.

Les inconvénientS[29]

• L'expéditeur et le destinataire doivent disposer, des versions compatibles de logiciels PGP ;

• La courbe de la complexité et de l'apprentissage de l'utilisation de PGP, peut être intimidante ;

• Pas de récupération, ou les administrateurs informatiques sont souvent confrontés à des situations d'urgence, impliquant la perte ou l'oublie des mots de passe.

2.12 La différence entre les standards

Les standards cités peuvent être comparés, suivant la technique de génération des paramètres variables qu'ils utilisent et leurs caractéristiques, types d'authentification, moyens et durée d'authentification et le type de chiffrement.

	kerberos	PGP	certificat
Types d'authentification	centralisée	distribuée	centralisée
Moyens d'authentification	Ticket	aucun	certificat
Durée d'authentification	Dépend de la durée de vie du ticket et la durée e vie de la session	Dépend de la durée e vie de la session	Dépend de la validité du certificat
Type de chiffrement	Clé privée et clé de session	Clé privée, clé publique et clé de session	Clé privée et clé publique

TABLE 2.1 – Tableau de différence entre les standards

2.13 Conclusion

Dans ce chapitre, nous avons présenté les différents mécanismes de sécurité, ainsi que les techniques d'authentification. Enfin nous avons étudié quelques standards d'authentifications, pour essayer d'en tirer les solutions les mieux adaptées pour une conception judicieuse d'un système d'authentification, permettant à un utilisateur de s'authentifier une seule fois pour bénéficier d'un accès multiple.

Chapitre 3

Conception du système

3.1 Introduction

Ce chapitre est consacré, pour l'analyse et à la conception du projet qui consiste au développement, d'un système d'authentification sécurisé, permettant à un utilisateur d'accéder à toutes les applications qui lui sont dédiées, et ce rien qu'après une seule authentification.

Pour la modélisation de notre système, nous avons opté pour le langage : « UML : unified modeling language ».

Le choix de ce dernier, se justifie par une panoplie d'avantages. Parmi tant d'autres, nous citons le fait qu'il permet de développer des programmes et des systèmes très fiables, en des temps de développement records, du coups la productivité est améliorée et le coût a diminue,et nous avons suivi la démarche de développement logiciel UP (Unified Processus).

3.2 Modèles des systèmes distribués

Un système distribué est un ensemble d'entités autonomes de calcul (ordinateurs, PDA, processeurs, processus, processus léger etc.) interconnectées et qui peuvent communiquer[32].

3.2.1 Le modèle Client/serveur

Désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur sur lequel est exécuté le logiciel client, le serveur, et l'ordinateur sur lequel est exécuté le logiciel serveur, Le rôle de chaque entité est le suivant[06] :

•Le client

— Effectue une demande de service auprès du serveur (Requête).

— Initie le contact (parle en premier), ouvre la session.

•Le serveur

— Est la partie de l'application qui offre un service.

— Est à l'écoute des requêtes clientes.

— Répond au service demandé par le client (réponse)

3.2.2 Le modéle Peer to Peer

Le terme peer-to-peer abrégé en P2P est un modèle de réseau informatique proche du modèle client-serveur mais qui est distribué de manière à ce que les entités appelées peers jouent le double rôle client et serveur. Tous les ordinateurs récupèrent de l'information et la retransmettent, en fait ils interagissent afin d'offrir à une communauté un service de manière décentralisé[32].

3.3 Outils de communication

3.3.1 Les sockets

Les sockets sont des interfaces standards pour les communications avec le protocole TCP. Elles ont été introduites à l'origine dans le monde Unix et sont à présent un standard de fait. Dans ce mode de communication (au travers des classes Socket et ServerSocket), le protocole TCP assure la gestion de la connexion, l'envoi, la réception, le séquencement et l'assemblage des paquets transmis. L'établissement de la connexion et la mise en forme ou l'extraction des données envoyées ou reçues sont à la charge du programmeur (ces dernières opérations sont largement facilitées par les classes ObjectInputStream et ObjectOutputStream)[19].

3.3.2 Les Servlets

Les servlets sont utilisées avec le protocole HTTP. Ce sont des petites applications téléchargées du serveur qui s'exécutent sur le client. Elle sont téléchargées via le protocole HTTP à l'initiative d'un navigateur web. Cependant, elles ne possèdent aucun mécanisme permettant de communiquer avec un serveur. Elle sont simplement stockées coté serveur, au même titre que des images ou des pages HTML mais s'exécutent sur le client de façon autonome. La servlet HTTP vient alors comme partie serveur naturelle de l'applet. Elle est une application s'exécutant coté serveur uniquement et pouvant intervenir entre le client HTTP (le navigateur) et le serveur HTTP (Apache, IIS...) pour contrôler les requêtes et effectuer des actions que le protocole HTTP ne sait pas faire. L'une des applications les plus répandues des servlets est la génération de pages HTML dynamiques. L'intérêt des servlets est alors évident vis à vis de solutions distribuées type RMI / Corba ou sockets : les servlets possèdent toutes les méthodes pour gérer facilement le HTTP[19].

3.4 Analyse des besoins

A fin que ce logiciel soit applicable et atteint tout les objectifs soulignés, il doit garantir :

— La génération des certificats à partir du serveur d'identité lors de l'ajout d'un utilisateur ou bien d'un fournisseur de service ;

— La vérification des certificats des utilisateurs et des fournisseurs de services auprès du serveur d'identité ;

— La révocation des certificats lorsqu'ils ne sont pas intègre ;

— L'authentification unique des clients, auprès de fournisseur de services et avoir accès à ses déférents applications permises.

3.5 Démarche suivie pour la conception du protocole

UML (Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet. Le processus suivre pour développer ce projet est UP, qui est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel[15].

3.6 Architecture du protocole

La figure suivante illustre l'architecture globale du système proposé

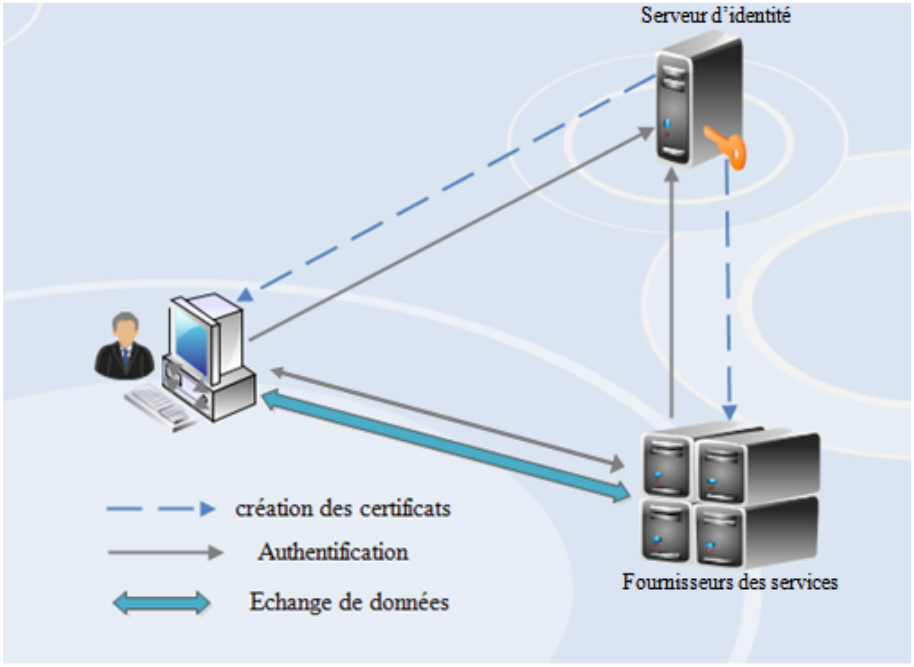

FIGURE 3.1 – Architecture globale du protocole.

Il comprend quatre entités principales qui sont :

— **Le serveur d'identité :** cette entité est le responsable de démarrage de l'environnement, il a également pour rôle de crée les certificats, aux utilisateurs et aux fournisseurs des services enregistrés,comme il permet aussi d'authentifier ces derniers, et de générer des clés de session pour chaque communication.

— **Le fournisseur des services :** c'est un ensemble des services, intégrés dans un seul serveur. ce serveur s'authentifie auprès de serveur d'identité pendant une durée donnée, afin de déterminer les fournisseurs des services fonctionnels.

— **L'application :** cette dernière permet de récupérer les services disponibles à partir du serveur d'identité, puis les affiche aux utilisateurs accédant à cette application.

— **L'utilisateur :** c'est l'entité qui demande un service offert par un fournisseur des services,pour cela il est demandé de s'authentifier en envoyant son certificat, son login et mot de passe au serveur d'identité. Si ce dernier accepte l'authentification, l'utilisateur sera authentifié auprès du service et pourra alors l'utiliser.

3.7 Diagramme de collaboration

3.7.1 Définition

Le diagramme de collaboration permet de mettre en évidence, les interactions entre les différents objets du système étudié, ainsi que les messages qu'ils échangent entre eux.
la figure suivante montre le diagramme de collaboration du protocole proposé[15] :

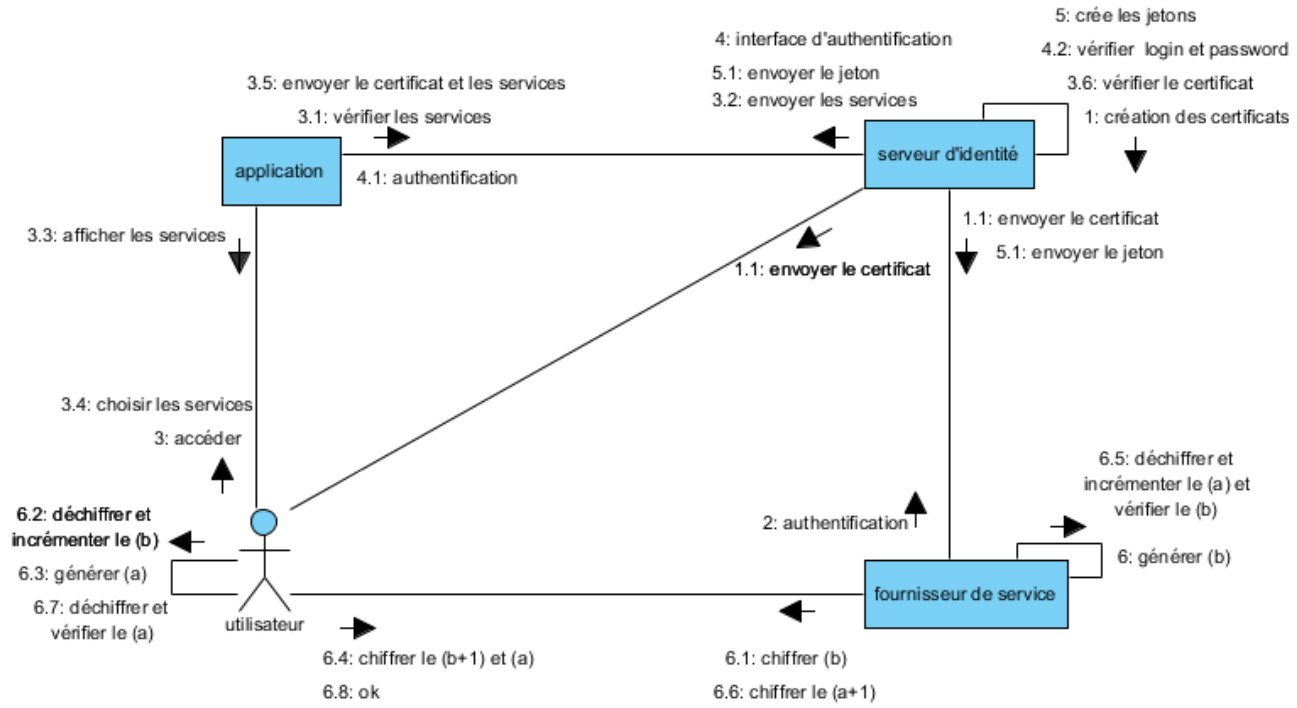

FIGURE 3.2 – Diagramme de collaboration"les interactions entre les différentes entités du système".

3.8 Fonctionnement du protocole

Le fonctionnement du protocole proposé, se déroule en plusieurs étapes :

1. Le serveur d'identité génère les clés asymétriques, et cela en utilisant l'algorithme RSA, et des certificats pour chaque employé et chaque fournisseur des services enregistrés.
2. Les fournisseurs des services s'authentifient au serveur d'identité pendant un délai donné par le serveur.
3. l'utilisateur accède à son application, cette dernière vérifie le certificat de l'utilisateur.
4. L'application récupère les services disponibles à partir de serveur d'identité.
5. L'application affiche les services à l'utilisateur.
6. L'utilisateur choisis au moins un service désiré.
7. L'application envoi une requête qui contient le certificat et les services choisis au service d'identité.
8. A la réception de la requête le serveur d'identité vérifie l'intégrité et la validité du certificat, et récupère les adresses des fournisseurs des services choisit par l'utilisateur. Le serveur d'identité génère un nombre aléatoire (nb) unique, le signe avec sa clé privée puis il le chiffre avec la clé publique de l'utilisateur ensuite ce nombre est envoyé à l'utilisateur.
9. L'utilisateur récupère ce nombre, vérifie la signature avec la clé publique du serveur, et le déchiffre avec sa clé privée.il incrémente se nombre (nb+1) puis il l'envoi avec le couple login/mot de passe : le tout chiffré avec la clé publique du serveur d'identité.
10. Le serveur d'identité déchiffre le message avec sa clé privée, et vérifie : le nb+1, le login et le mot de passe.il vérifie aussi si c'est le même client qui a envoyé le certificat. Le serveur d'identité génère la clé secrète en utilisant DES, pour chaque session entre un client et un fournisseur des services. Il crée par la suite des jetons (l'adresse du client, l'adresse de fournisseur et la clé secrète), un jeton sera chiffré d'une part ; avec la clé publique de l'utilisateur et lui sera envoyé. Et d'autre part ; avec la clé publique du fournisseur et lui sera a son tour envoyé.
11. A la réception du jeton, l'utilisateur et le fournisseur le déchiffre chacun avec sa clé privée, Et récupèrent la clé secrète. Puis Le fournisseur des services génère un nombre aléatoire unique (b) qu'il chiffre avec la clé secrète, et l'envoie à l'utilisateur.
12. A la réception du nombre ; l'utilisateur le déchiffre avec la clé secrète, et l'incrémente (b+1), puis génère un autre nombre aléatoire (a) et envoie le tout chiffré avec la clé secrète au fournisseur de services.
13. A son tour, le fournisseur des services déchiffre le message avec la clé secrète et comparé le nombre incrémenté (b+1) avec le nombre qu'il a généré (b), puis il incrémente le nombre aléatoire généré par l'utilisateur, il devient (a+1) puis il le chiffre avec la clé secrète, il le transmit ainsi au client.
14. L'utilisateur le déchiffre avec la clé secrète et vérifie s'il s'agit du même nombre (a). A cette étape les deux sont authentifiés l'un auprès du l'autre.

3.9 Diagramme de cas d'utilisation

Les cas d'utilisations décrivent sous la forme d'actions et de réactions, le comportement d'un système du point de vue utilisateur. ils servent à structurer les besoins des utilisateurs et les objectifs correspondants du système[10].
la figure suivante définie le cas d'utilisation du protocole proposé :

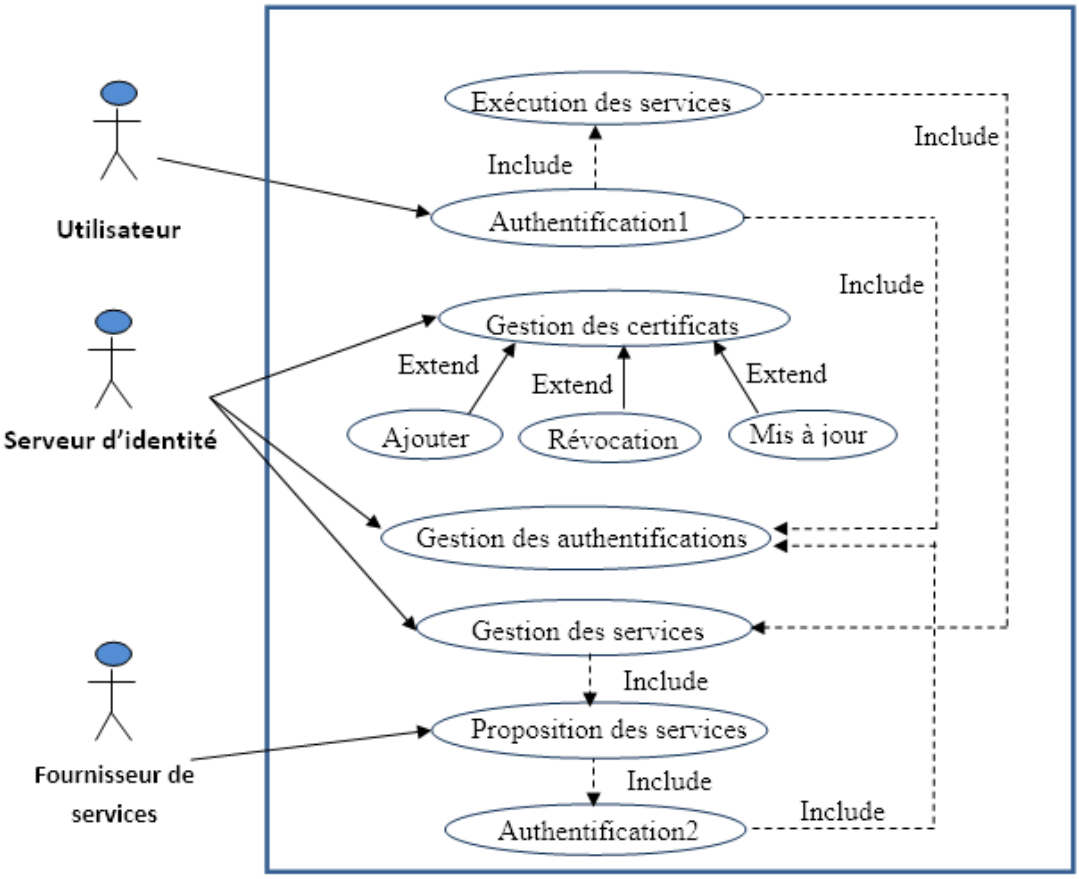

FIGURE 3.3 – Diagramme de cas d'utilisation.

3.10 Description des cas d'utilisation

on retrouve les cas suivants :

— **Exécution des services :** ce cas d'utilisation offre à l'utilisateur la possibilité d'exécuter les services choisis.

— **Authentification 1 :** l'exécution des services ne sera enclenchée qu'après une authentification de l'utilisateur auprès du serveur d'identité.

— **Gestion des certificats :** c'est un cas propre au serveur d'identité, qui se charge de créer des certificats aux utilisateurs et aux fournisseurs des services enregistrés. le serveur permet aussi la mise à jour des certificats dans le cas d'expiration de leurs durées de vies, et révoque les certificats qui ne sont pas intégrés.

— **Gestion des authentification :** dans ce cas le serveur d'identité vérifie l'authentification de deux acteurs(l'utilisateur et le fournisseur des services)soit valider soit refuser.

— **Gestion des services :** après l'authentification des fournisseurs des services, le serveur d'identité récupère les services disponibles et les affiche a l'utilisateur.

— **Proposition des services :** le fournisseur des services après son authentification,propose ces services disponibles au serveur d'identité.

— **Authentification 2 :** dans ce dernier les fournisseurs des services s'authentifient périodiquement au serveur d'identité.

cas d'utilisation	acteur	Objectif	Scénario	Extension
Authentification 1	utilisateur	authentification de l'utilisateur auprès du serveur d'identité pour exécuter les services choisis	-l'utilisateur envoie son certificat et les services désiré au serveur d'identité.-le serveur d'identité vérifie la validité et l'intégrité de certificat, et génère un nombre aléatoire qu'il signe avec sa clé privée et aussi le chiffre avec la clé publique de l'utilisateur. -l'utilisateur déchiffre le nombre aléatoire et le incrémente, il envoi au serveur d'identité son login et mot de passe et le nombre incrémenter le tout chiffré avec la clé publique du serveur.-le serveur d'identité compare le nombre incrémenté avec le nombre qu'il a généré et vérifie s'il s'agit le détendeur du certificat.	-si le certificat n'est pas intègre, le serveur d'identité envoi un message d'erreur a l'utilisateur.-si l'utilisateur n'est pas ce lui qui a envoyé le certificat, le serveur d'identité lui refuse l'accès au service désiré
Authentification 2	Fournisseur de services	authentification des fournisseurs des services auprès de serveur d'identité	-le fournisseur des services envoi son certificat au serveur d'identité. -le serveur d'identité vérifie la validité et l'intégrité de certificat, et aussi l'identité de fournisseur des services.	-si le certificat n'est pas intègre, le serveur d'identité envoi un message d'erreur au fournisseur des services.

TABLE 3.1 – Taleau de description des cas d'utilisation

3.11 Diagramme de séquence

3.11.1 Définition

Le diagramme de séquence représente les échanges des messages entre objet, d'un point de vue temporel. Il montre ainsi la chronologie d'envoi des messages, et décrit les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets[30].

3.11.2 Diagramme de séquence :"authentification des fournisseurs des services"

Ce diagramme représente la phase d'authentification, du fournisseur de services au prés de serveur d'identité pour récupérer les services disponibles.

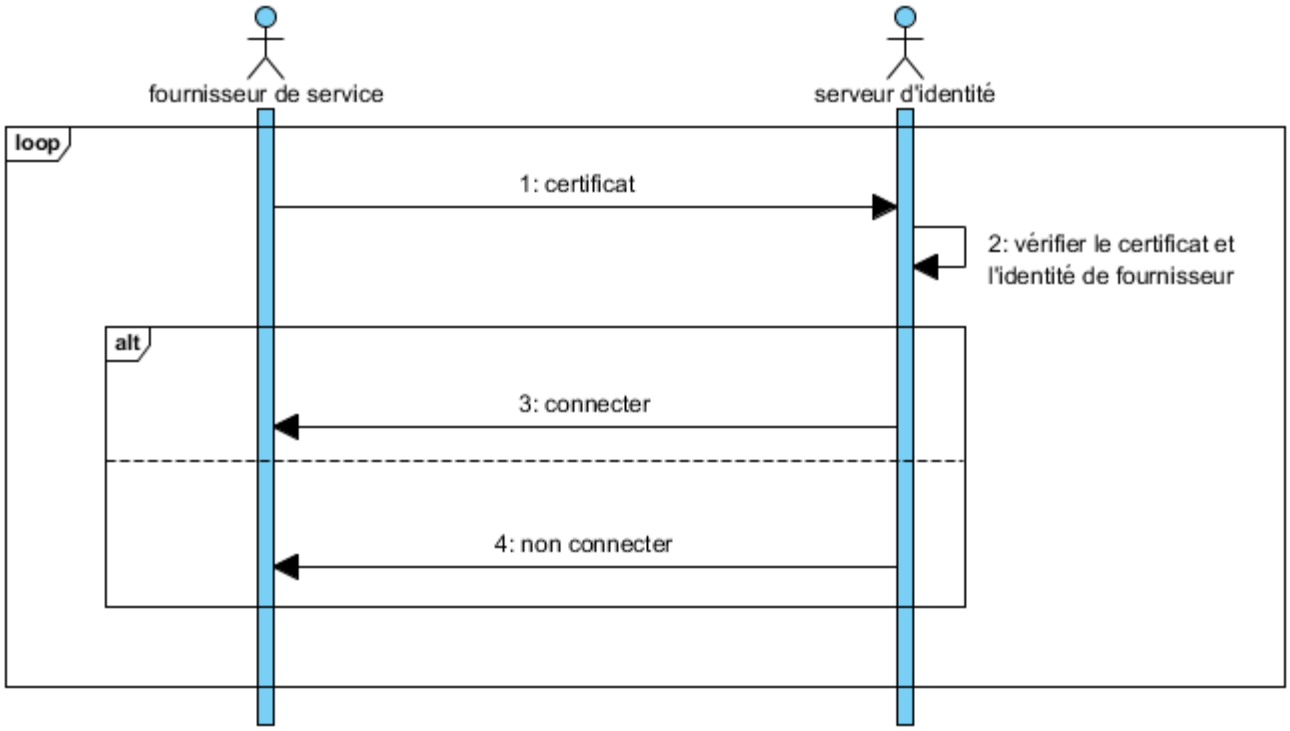

FIGURE 3.4 – Authentification des fournisseurs des services.

3.11.3 Diagramme de séquence : "récupération de certificat"

Dans ce diagramme, l'application vérifie l'existence du certificat de l'utilisateur connecter.

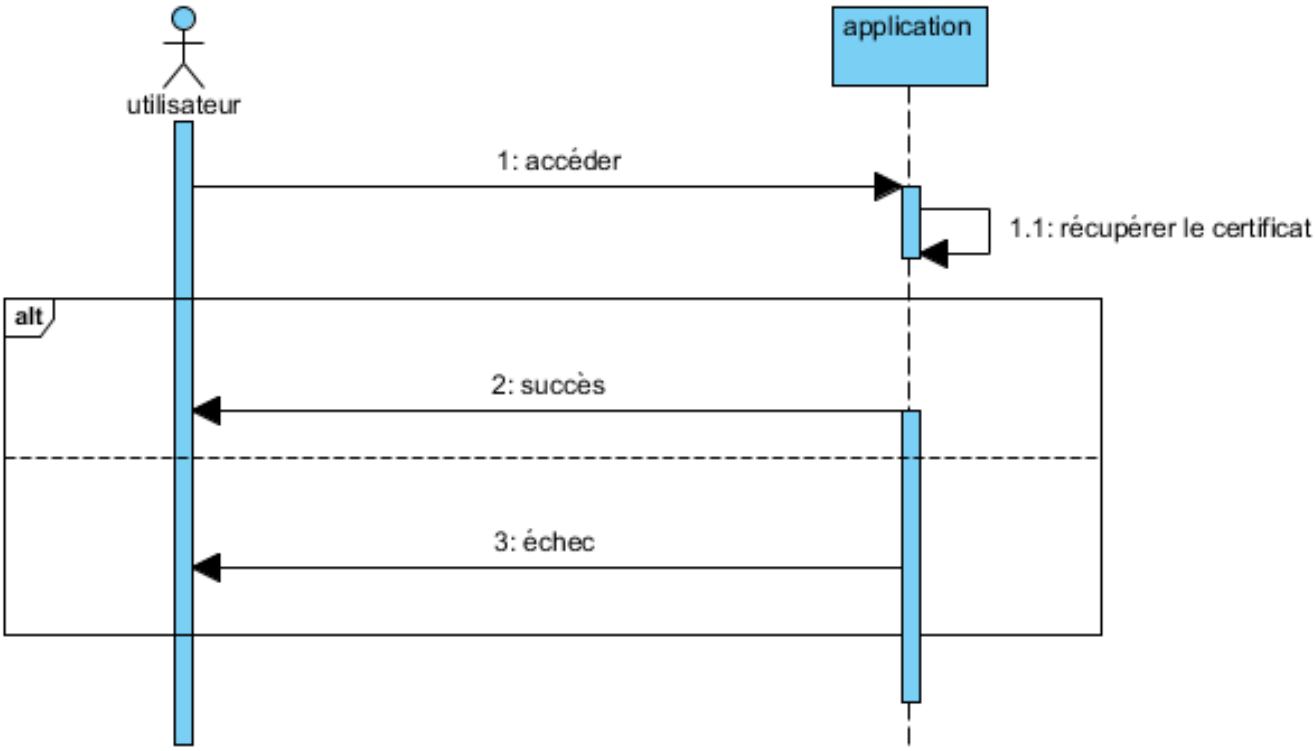

FIGURE 3.5 – Récupération du certificat.

3.11.4 Diagramme de séquence : "authentification de l'utilisateur"

Ce diagramme représente la phase d'authentification de l'utilisateur au prés de serveur d'identité afin d'exécuter les services disponible.

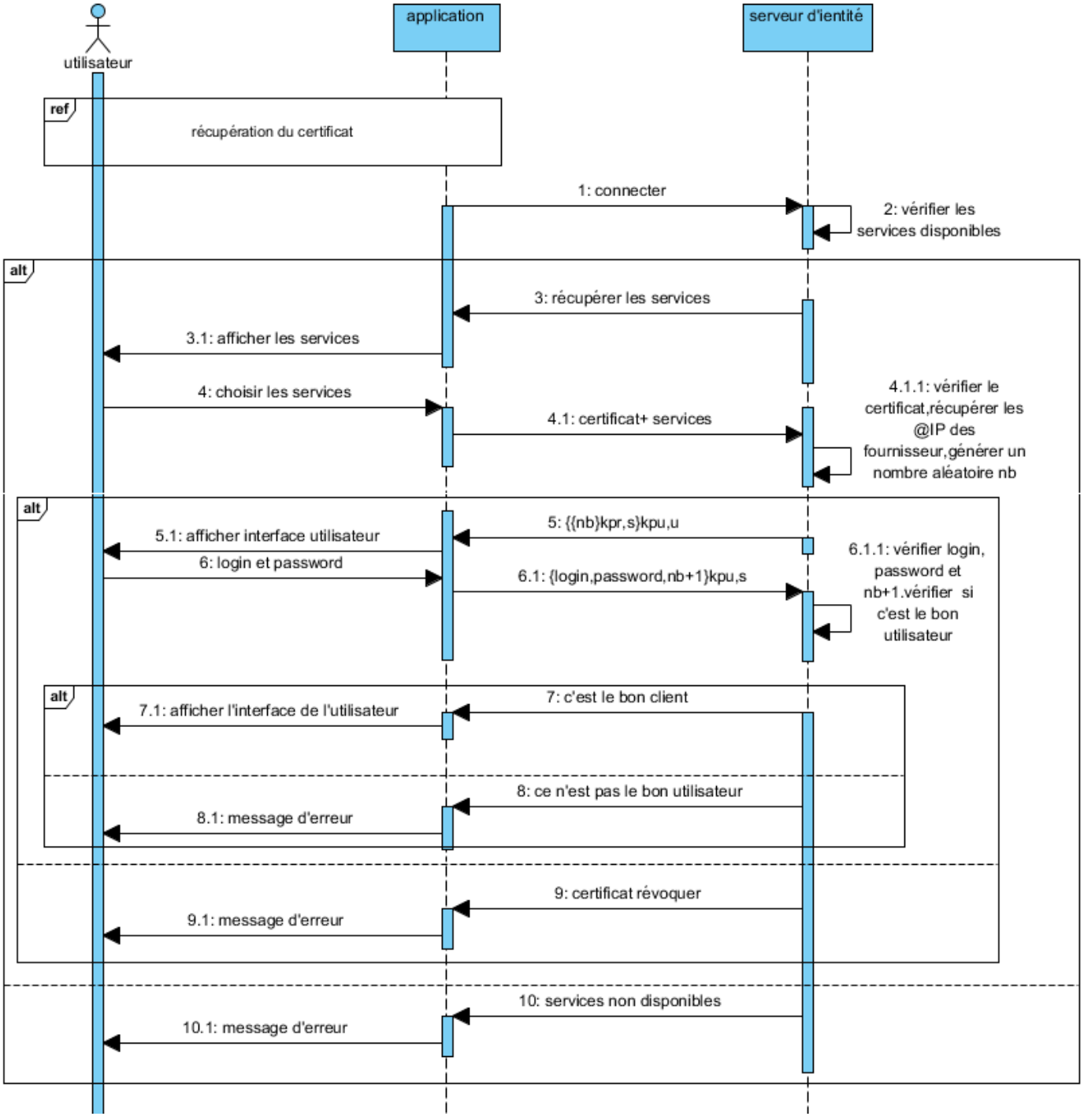

FIGURE 3.6 – Authentification de l'utilisateur".

— Le nombre aléatoire =le haché de la date courante.
— kpu,u = la clés publique de l'utilisateur.
— kpu,s = la clé publique de serveur d'identité.
— login = le nom concaténé avec le prénom.
— mot de passe = l'identifiant de l'utilisateur.

3.11.5 Diagramme de séquence : "création des jetons"

Le diagramme suivant illustre la phase de création des session entre l'utilisateur et les fournisseur des services.

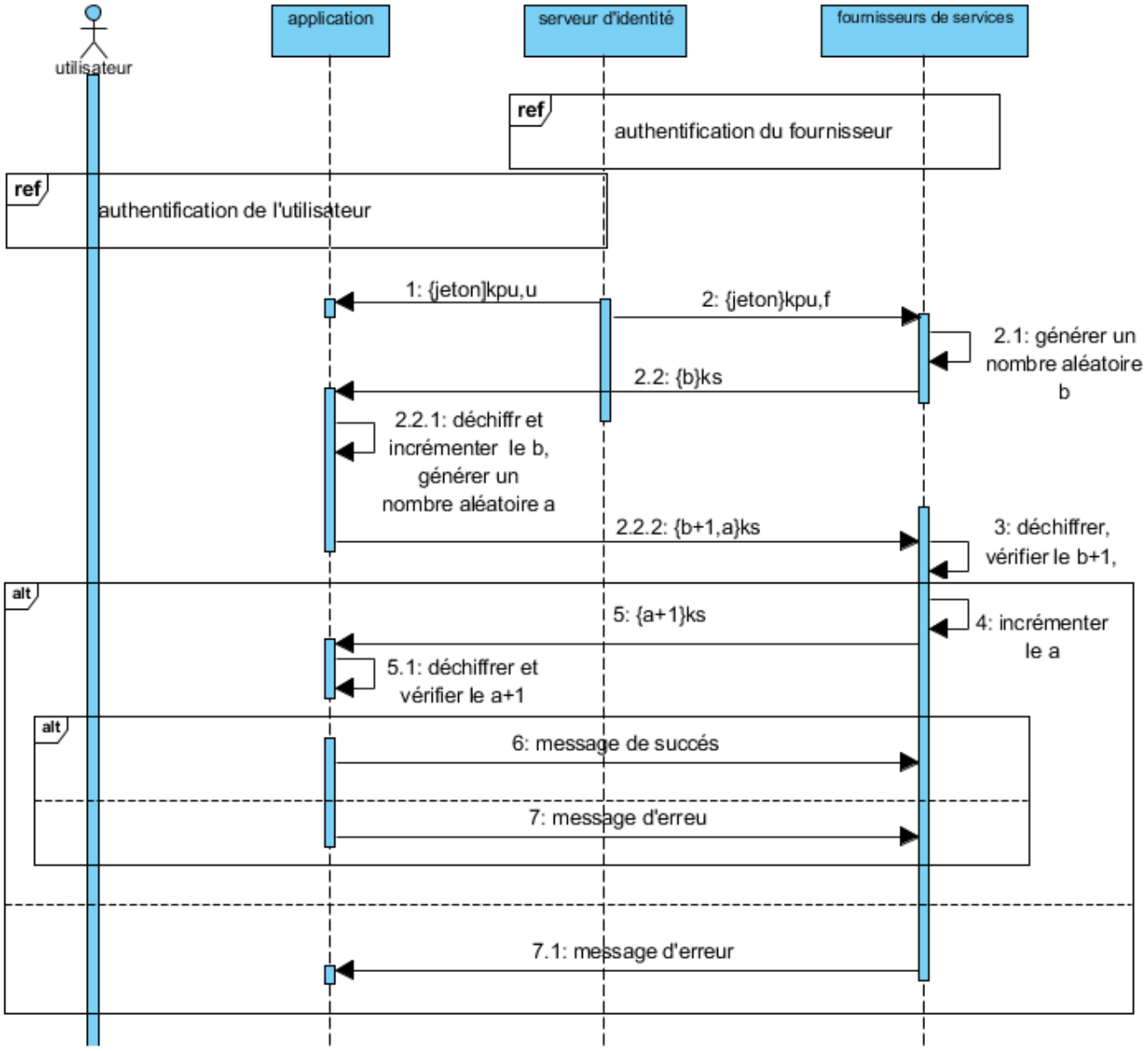

FIGURE 3.7 – Création des jetons.

— b= le nombre aléatoire généré par le fournisseur de services.
— a = le nombre aléatoire généré par l'utilisateur.
— kpu,f = la clé publique du fournisseur de services.
— ks = la clé secrète partagée entre l'utilisateur et le fournisseur de services.
— jeton = @ip du fournisseur de services,@ip de l'utilisateur,la clé secrète.

3.12 Diagramme d'activité

3.12.1 Définition

Un diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. Les actions sont les unités de base du comportement du système. Un diagramme d'activité permet de grouper et de dissocier des actions[1].

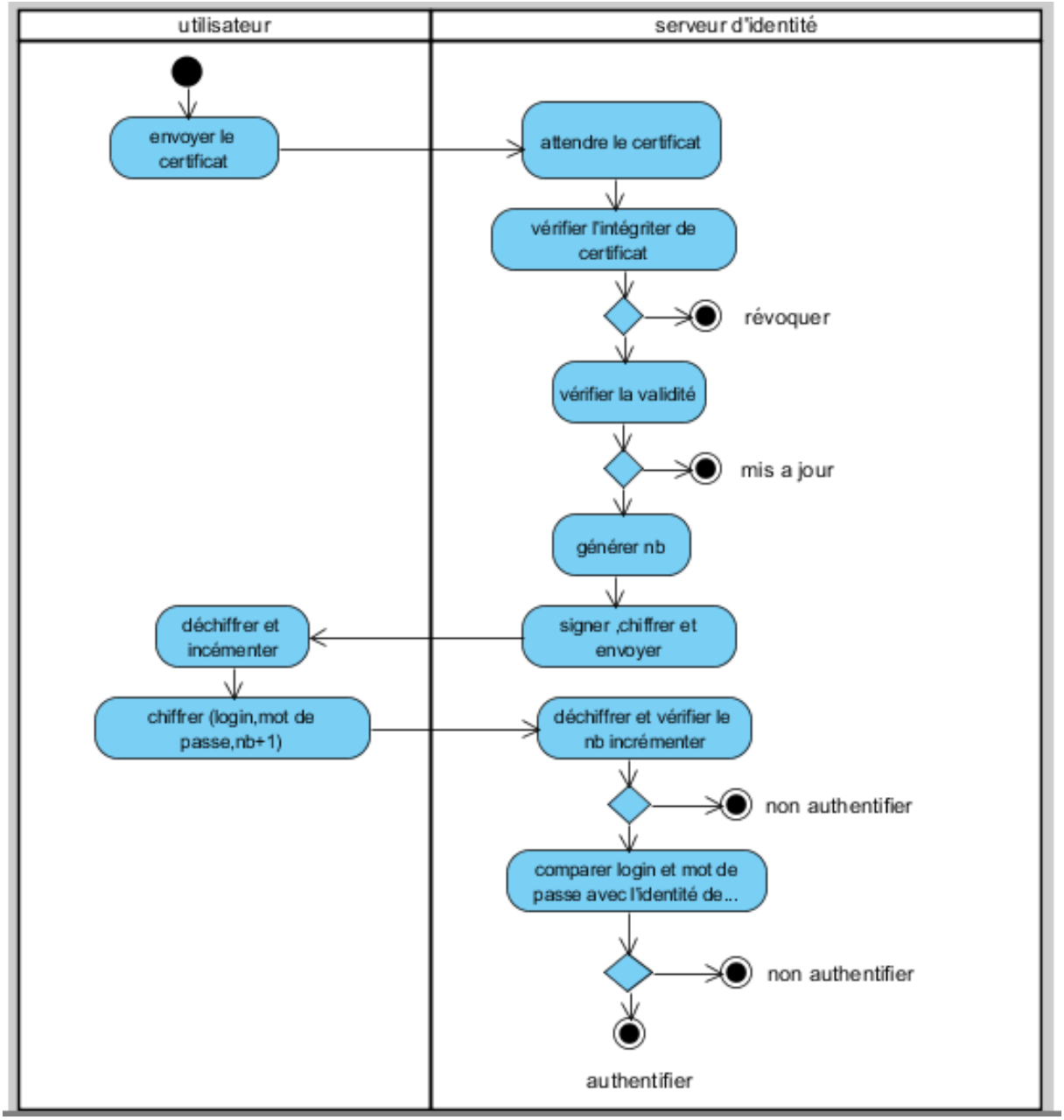

FIGURE 3.8 – Diagramme d'activité : "vérification du certificat".

3.13 Résistance aux attaques

3.13.1 Attaque men in the middle

L'attaque de l'homme du milieu (HDM) ou man in the middle attack (MITM), est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis[7].
Dans notre cas trois scénario se présente.

L'attaquant se fait passer pour le fournisseur des services auprès du serveur d'identité

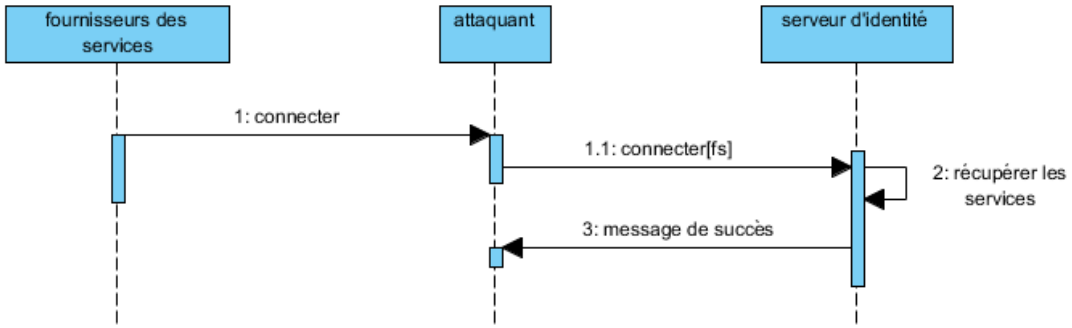

FIGURE 3.9 – Attaque men in the middle "fournisseur de services".

Dans l'implémentation du système proposé, on a utilisé les certificats pour faire face à cette attaque.

— Le fournisseur de services envoi son certificat pour s'authentifier auprès du serveur d'identité ;

— L'attaquant intercepte le certificat et modifie la clé publique de certificat ;

— Le serveur d'identité vérifie le certificat, et que celui-ci n'est pas intégré ;le serveur le prend comme étant une attaque.

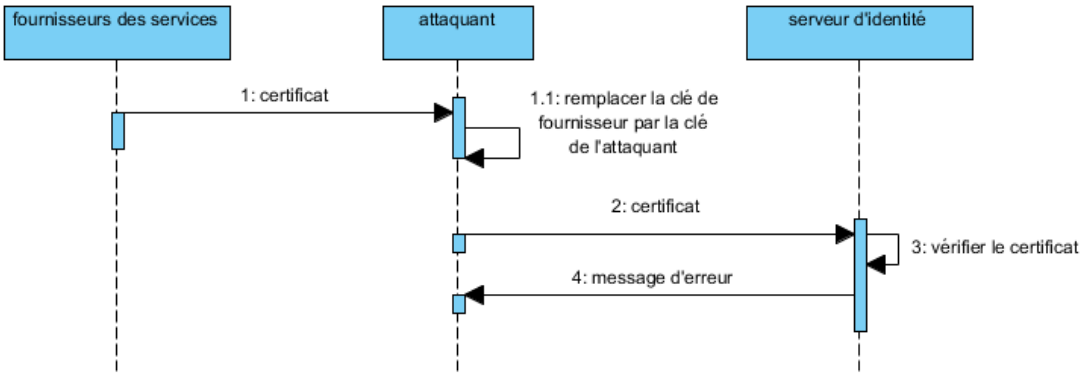

FIGURE 3.10 – Solution d'attaque men in the middle "fournisseur de services".

L'attaquant fait passer pour l'utilisateur auprès de serveur d'identité

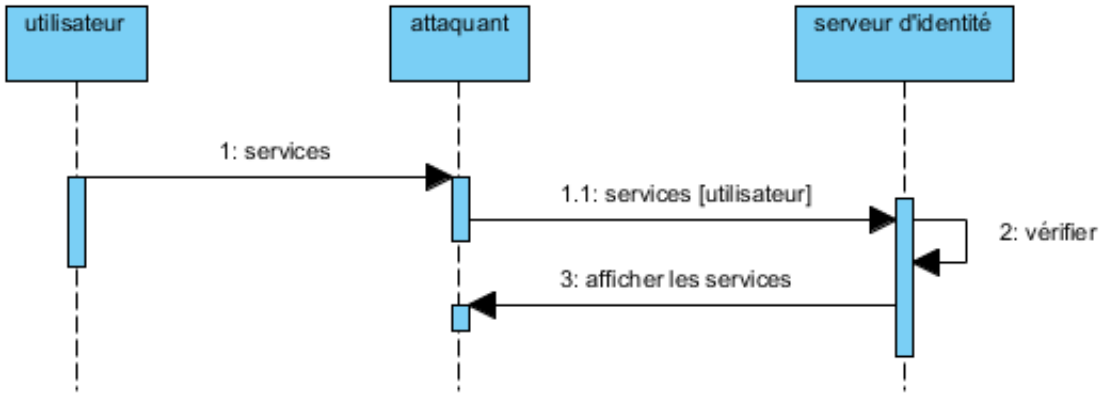

FIGURE 3.11 – Attaque men in the middle "utilisateur".

Le système implémenté empêche l'attaquant d'usurper l'identité de l'utilisateur et cela se présente comme suit :

— L'utilisateur envoi son certificat et les services choisis ;

— L'attaquant remplace la clé publique de certificat de l'utilisateur par sa clé publique ;

— Le serveur d'identité vérifie le certificat et que celui-ci n'est pas intègre.

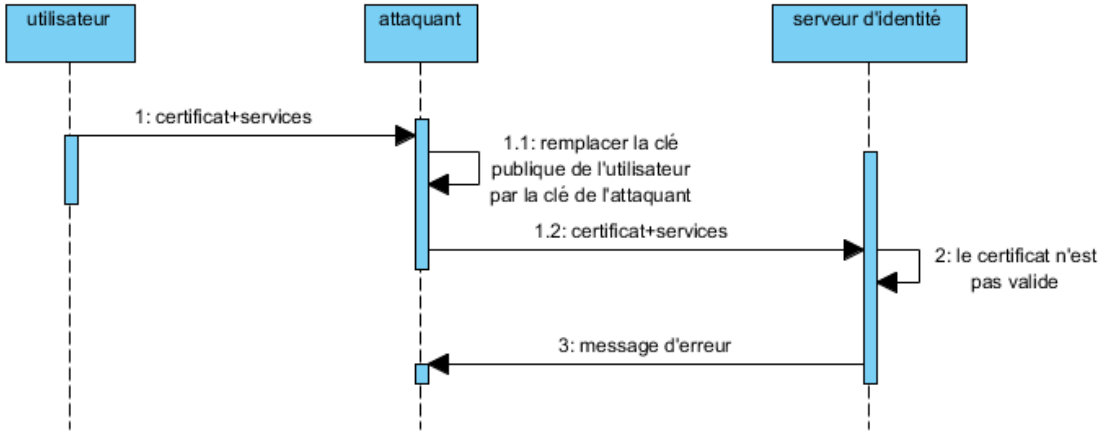

FIGURE 3.12 – Solution d'attaque men in the middle "utilisateur".

L'attaquant fait passer pour serveur d'identité

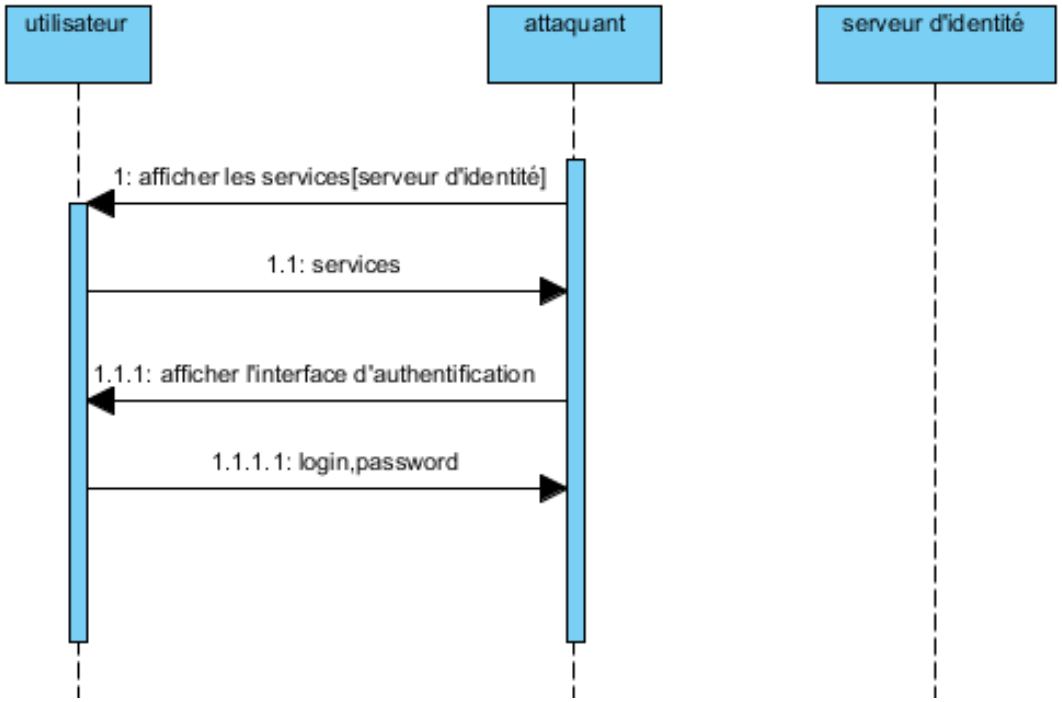

FIGURE 3.13 – Attaque men in the middle"serveur d'identité".

Le système proposé empêche l'attaquant de se faire passé pour le serveur d'identité et cela comme suit :

— A la réception du certificat et les services par l'attaquant qui se fait passé pour le serveur d'identité,l'attaquant génère un nombre aléatoire le signe avec sa clé privée et le chiffre avec la clé publique de l'utilisateur.

— L'utilisateur déchiffre le message avec sa clé privée et récupère la la clé publique a partir de certificat des vrais serveur d'identité.l'utilisateur n'arrive pas a vérifier la signature,le système se bloque.

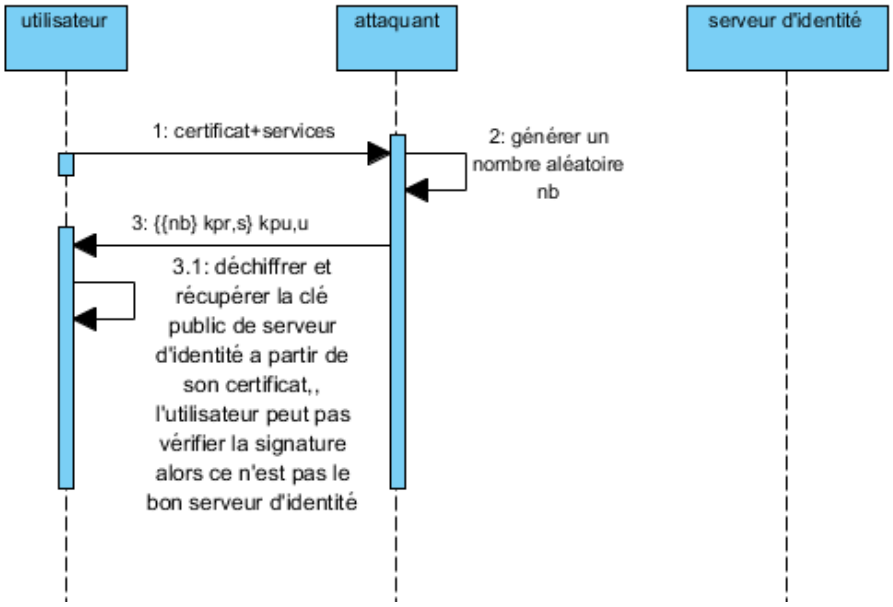

FIGURE 3.14 – Attaque men in the middle "serveur d'identité".

3.13.2 Attaque par rejeu

Les attaques par « rejeu » (en anglais « replay attaque ») sont des attaques de type « Man in the middle » consistant à intercepter des paquets de données et à les rejouer, c'est-à-dire les retransmettre tels quel (sans aucun déchiffrement) au serveur destinataire[7].

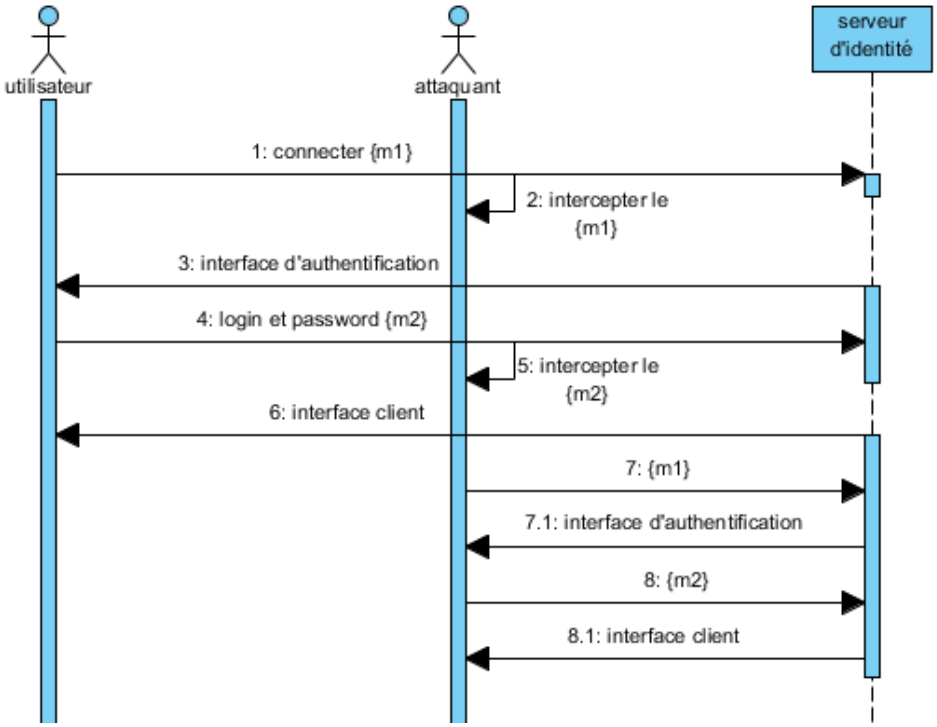

FIGURE 3.15 – Attaque par rejeu.

Solution d'attaque

Le système implémenter propose une solution pour ce faire face a l'attaque par rejeu cela est définie par la figure.

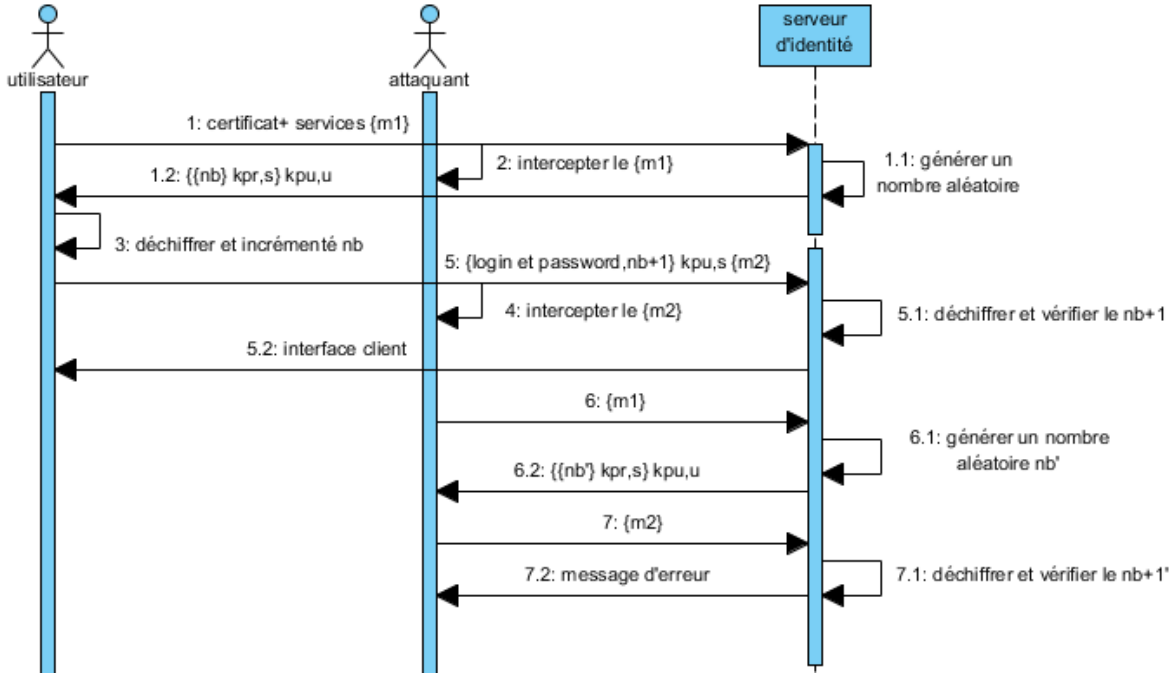

FIGURE 3.16 – Attaque par rejeu.

— L'attaquant intercepte les message envoyer par l'utilisateur au serveur d'identité ;
— Il envoi le 1er message intercepte ;
— Le serveur d'identité génère un nouveau nombre aléatoire et le signe avec sa clé privée et le chiffre la clé publique de l'utilisateur ;
— L'attaquant envoi le 2eme massage ;
— Le serveur d'identité vérifier le nombre,et que ce lui ci n'est pas le même qui a généré.

3.14 Conclusion

A travers ce chapitre, nous avons défini le système distribué utilisé, ainsi que l'outil choisi pour faciliter la communication des messages.

nous avons proposé une modélisation de la solution avec UML ; en utilisant ses différents diagrammes.

la solution sera implémenté dans le chapitre suivant.

Chapitre 4

Implémentation

4.1 Introduction

Après avoir achevé l'étape de conception de l'application, on va entamer dans ce chapitre la partie réalisation et implémentation dans laquelle nous allons décrire les différents outils et classes utilisés ainsi que quelques captures du logiciel et s'assurer que le système est prêt pour être exploité par les utilisateurs.

4.2 Outils de développement

Pour la réalisation de ce travail, nous avons eu recours aux outils suivants :

- Java comme langage de programmation ;
- Eclipse comme IDE ;
- NetBeans comme IDE ;
- Socket comme communication inter processus (Client/Serveur) ;

— **Java :** Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems en 1995. Il s'agit de la technologie sous-jacente qui permet l'exécution de programmes dernier cri, notamment des utilitaires, des jeux et des applications professionnelles. Le langage Java est utilisé sur plus de 850 millions d'ordinateurs de bureau et un milliard de périphériques dans le monde, dont des périphériques mobiles et des systèmes de diffusion télévisuelle[20].

— **Eclipse :** Eclipse permet le développement d'applications Java principalement, mais également d'autres langages grâce à l'utilisation de plugins.
Eclipse est une plateforme de développement écrite en Java, fruit du travail d'un consortium de grandes entreprises (IBM, Borland, Rational Rose, HP...). Il en resulte un IDE performant et open Source qui a su trouver sa place comme l'un des IDs Java les plus populaires. Elle intègre pour cela la prise en charge des outils comme Ant, SVN, JUnit...
Au niveau ergonomie, Eclipse n'a rien à envier à ses concurrents. Toutes les fonctionnalités indispensables sont là : création de projet, de template, refactoring, debuggage ... et remarquablement faciles à prendre en main. Mais la grande force de cet IDE réside dans l'ouverture

de son noyau qui permet l'ajout de très nombreux plugins. Il est par exemple possible d'intégrer des éditeurs XML, HTML, JSP, etc. ou encore de déployer ses applications vers le quasi totalité des serveurs du marché[20].

— **NetBeans :** C'est un environnement de développement intégré (EDI) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS[20].

— **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, attends des demandes de connexion qu'un processus, appelé client, lui envoie. Une fois l'étape d'établissement de connexion effectuée le fonctionnement redeviens symétrique[20].

4.3 Présentation de l'application proposé

Dans le cadre de l'option Java, nous avons souhaité développer un système d'authentification unique, Le logiciel est composé de plusieurs applications distinctes : l'application Client et l'application Serveur d'identité et l'application du fournisseur.

L'application Client consistera en une petite fenêtre dans laquelle l'utilisateur peut s'authentifier par la suite accéder a son profil.

L'application Serveur aura, quant à elle, le rôle de vérifier l'authentification des clients et des fournisseurs, de gérer les connexions (par mot de passe).

De plus, elle implémentera un mini-site permettant l'administration, la gestion des utilisateurs et des fournisseurs.

Enfin l'application du fournisseur qui lui permettra de s'authentifier auprès du serveur et de se re-authentifier après une durée de temps spécifique.

Le schéma suivant montre le principe de fonctionnement de notre application en mettant en valeur les principales classes et relation entre les différents composants.

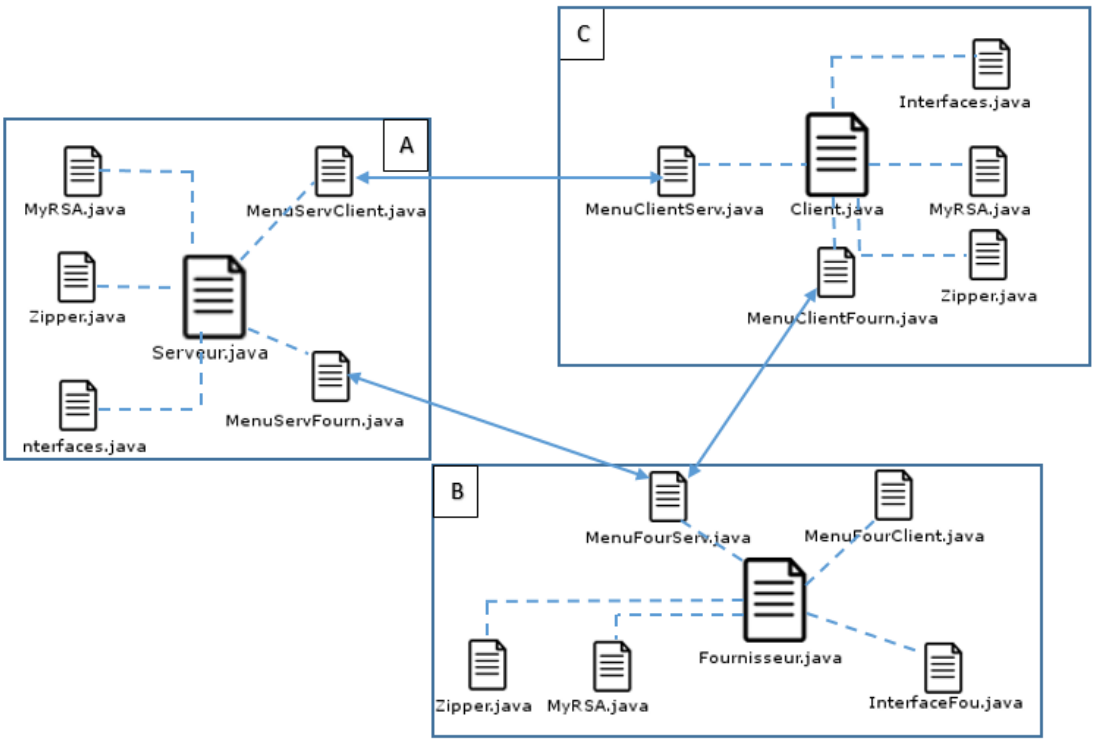

FIGURE 4.1 – Présentation des classes d'implémentation de système.

4.3.1 Description des parties

Partie A

- Représente le serveur d'identité qui a un socket de type SocketServer d'un port 2016 ;
- Possède deux classes principales MenuServeurFourn et MenuServeurClient ;
- Permet la vérification des certificats, gère la révocation et la mise à jour des certificats ;
- Gérer la base de données.

Partie B

- Représente le Fournisseur de services qui a deux types de sockets Socket et SocketServer d'un port 2017 ;
- Possède deux classes principales MenuFourServ et MenuFourClient ;
- Propose les différents services qui possèdent ;
- Possède une paire de clés publique/privé et un Certificat signé par le serveur ;
- Son authentification dure 12 heures.

Partie C

- Représente le Client qui à un socket d'un numéro de port qui varie entre 0 et 1023
- Possède deux classes principales MenuClientServ et MenuClientFourn ;
- Exécute les différents services en ligne ;
- Possède une paire de clés publique/privé et un certificat signé par le serveur d'identité.

4.3.2 Les différentes classes

La classe Socket

- La classe Socket représente en Java les sockets utilisés côtés client et coté serveur, elle permet de fournir un numéro de port concaténer à une adresse IP pour quelle soit unique.
- Elle permet au client de communiquer avec le Serveur et le Fournisseur de services.

La classe MenuServeurClient et MenuServeurFourn

- Ces deux classes spécifiques au serveur d'identité permettent de créer des BufferReader et PrintWriter avec le client et le fournisseur a fin d'établir une communication pour pouvoir partager des flux de données.

La classe MenuClientFour et MenuClientServ

- Ces deux classes spécifiques au serveur d'identité permettent de créer des BufferReader et PrintWriter avec le serveur et le fournisseur a fin d'établir une communication pour pouvoir partager des flux de données.

La classe MenuFournClient et MenuFourServeur

- Ces deux classes spécifiques au serveur d'identité permettent de créer des BufferReader et PrintWriter avec le client et le serveur a fin d'établir une communication pour pouvoir partager des flux de données.

La classe MyRSA

- Cette classe possède plusieurs méthodes qui lui permettent de générer une paire de clés, chiffrement et le déchiffrement.

La classe Zipper

- Cette classe permet de compresser et décompresser les fichiers avec un mot de passe.

4.3.3 Méthodes informatives

public InetAddress getInetAddress (), public int getPort ()

Ces méthodes renvoient l'adresse Internet et le port distants auquel le socket est connecté.

public InetAddress getLocalAddress (), public int getLocalPort ()

Ces méthodes renvoient l'adresse Internet et le port locaux que le socket utilise.

generatedKeyPair ()

Cette méthode permet la génération d'une paire de clés.

crypt(), decryptInString(),decryptInBytes()

Ces méthodes permettent le chiffrement et le déchiffrement.

new Thread(new MenuServerClient(socket),t1.start()

Les threads sont des fils d'exécution, elles permettent d'établir plusieurs communications en parallèle.

Zipper.pack(password),Zipper.unpack(fichier,password)

Ces méthodes permettent de compresser et décompresser des fichiers avec un mot de passe.

4.4 La mis en œuvre

4.4.1 Interface du serveur d'identité

Cette interface permet au serveur de lui afficher les clients en ligne, la liste de révocation de certificats et les différents échanges lors d'une tentative de connexion.

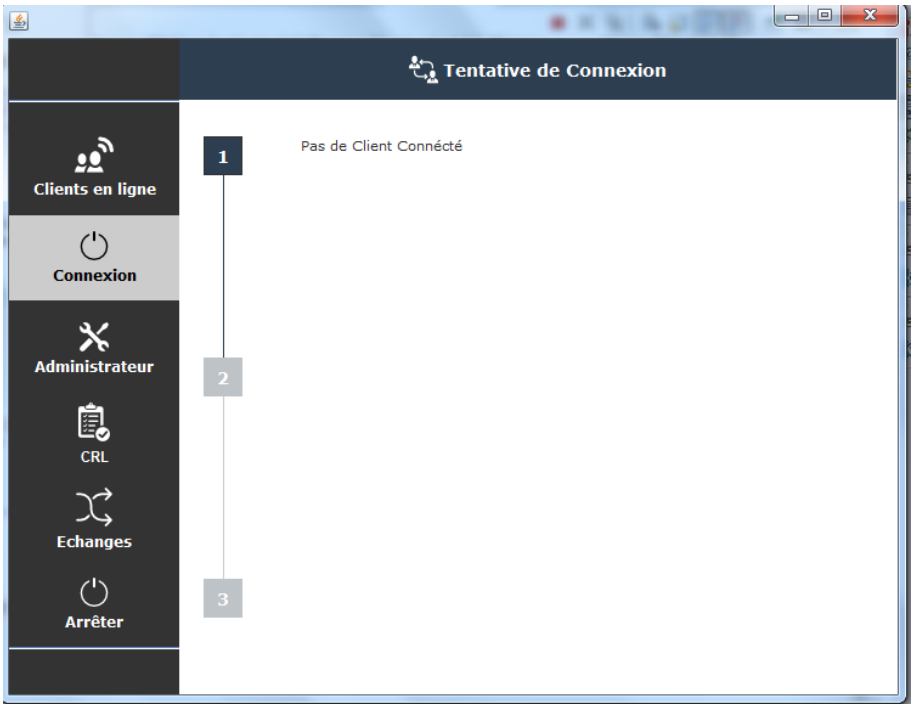

FIGURE 4.2 – Interface du serveur d'identité

4.4.2 Interface des certificats révoqués

Cette interface permet d'afficher la liste de révocation de certificats (Non intègre ou Compromis).

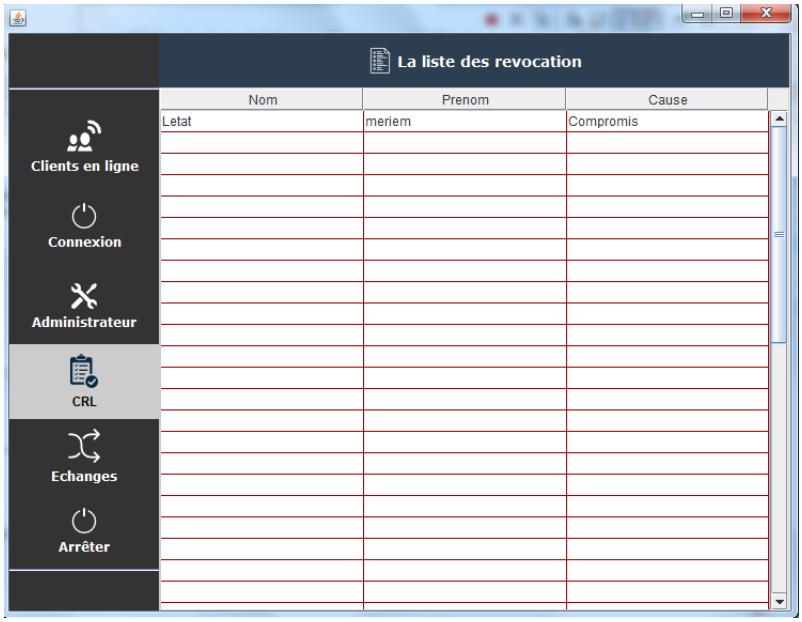

FIGURE 4.3 – Interface des certificats révoqués

4.4.3 Interface des utilisateurs en ligne

cetta interface permet d'afficher les utilisateur en ligne avec leurs adresses IP respectives.

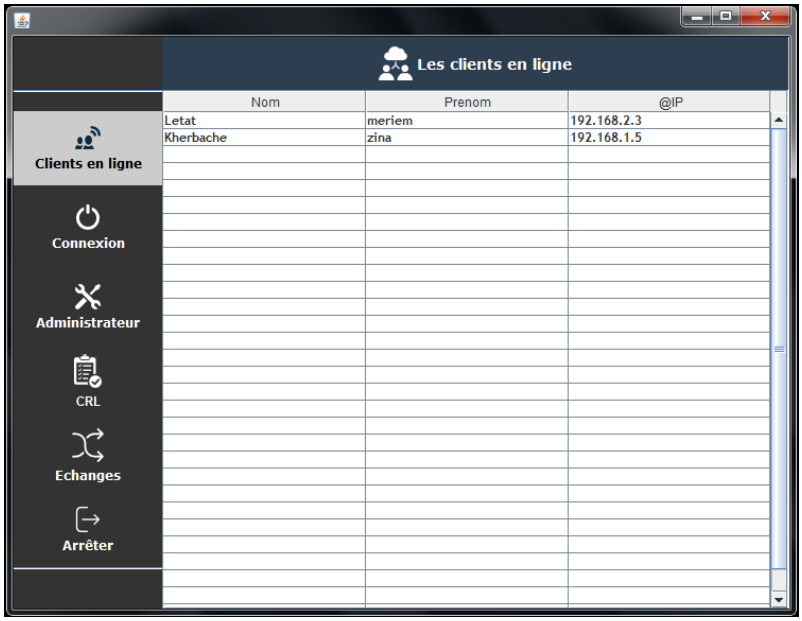

FIGURE 4.4 – Interface des utilisateurs en ligne

4.4.4 Interface d'accueil des fournisseurs des services

Cette interface permet au fournisseur d'afficher son certificat et de voir les différents échanges entre le fournisseur de service et les clients et le fournisseur et le serveur d'identité.

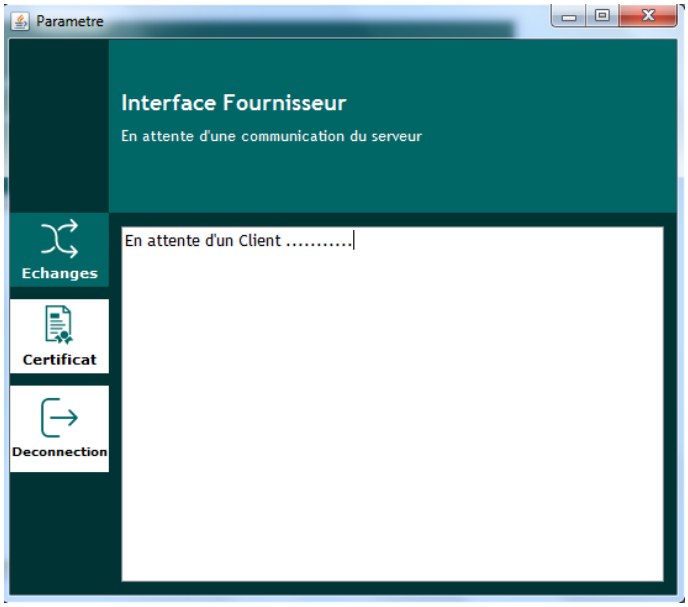

FIGURE 4.5 – Interface d'accueil des fournisseurs des services

4.4.5 Interface des échanges de Serveur d'identité

Cette Interface permet d'afficher tous les échanges qui transitent entre serveur d'identité et les clients ainsi que le serveur d'identité et les fournisseurs de services.

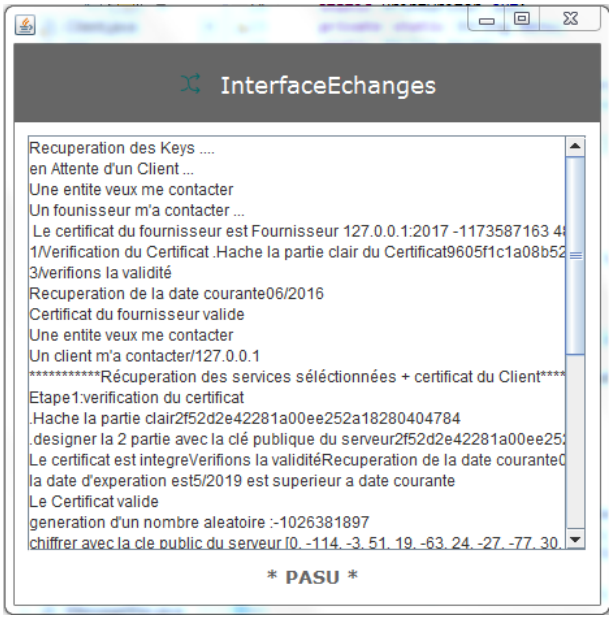

FIGURE 4.6 – Interface des échange de serveur d'identité

4.4.6 Interface d'importation du certificat de l'utilisateur

Cette interface permet à l'utilisateur d'importer les fichiers nécessaires à partir d'un USB à savoir le certificat.crt et private.key pour pouvoir s'authentifier (simulation d'une carte RFID).

FIGURE 4.7 – Interface d'importation du certificat de l'utilisateur

4.4.7 Interface de certificat

cette interface permet de présenté le certificat de l'utilisateur et de fournisseur de services.

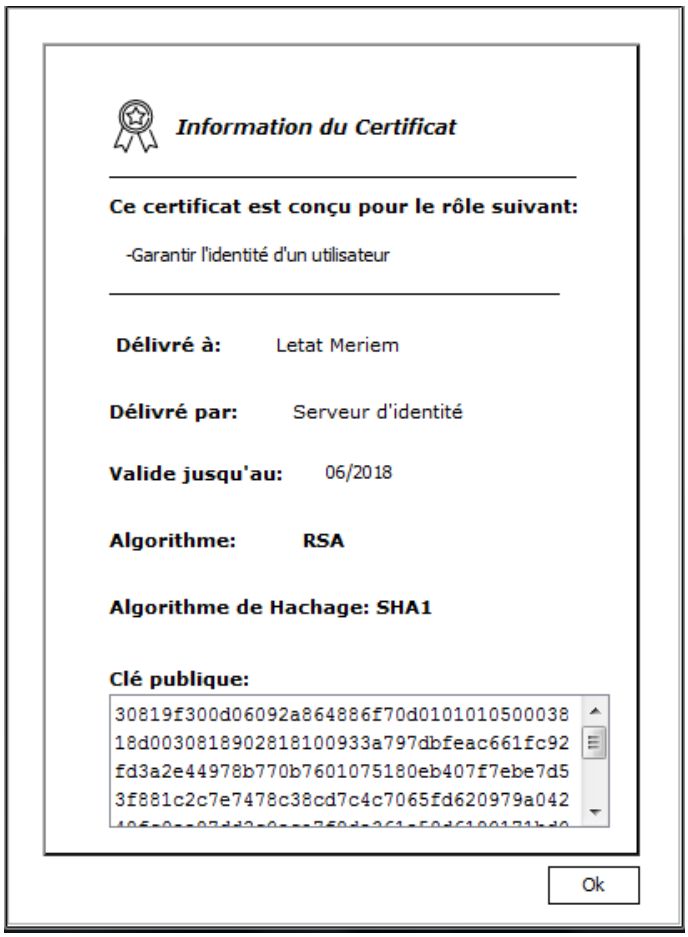

FIGURE 4.8 – certificat

4.4.8 Interface de choix des services en ligne

Cette interface est la première étape d'authentification d'un client car il faudrait choisir au moins un service en ligne pour pouvoir accéder à l'interface de connexion.

FIGURE 4.9 – interface de choix des services en ligne

4.4.9 Interface de vérification de certificat

Cette Interface permet d'afficher les différentes étapes de vérification d'un certificat fourni par un utilisateur.

Cas1 : intègre et valide

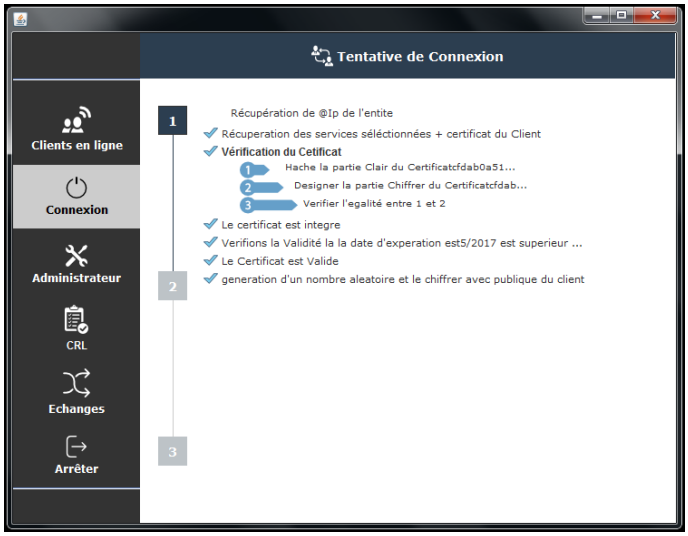

FIGURE 4.10 – Interface d'un certificat intègre et valide

Cas2 : non intègre

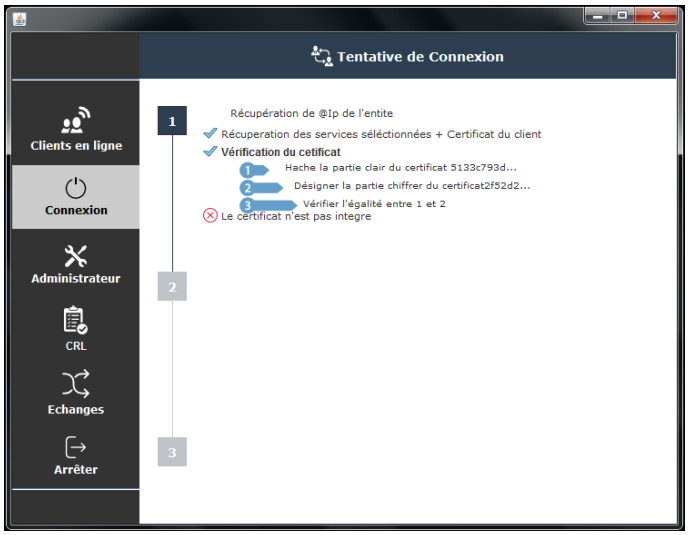

FIGURE 4.11 – Interface d'un certificat non intègre

Cas3 : intègre et invalide

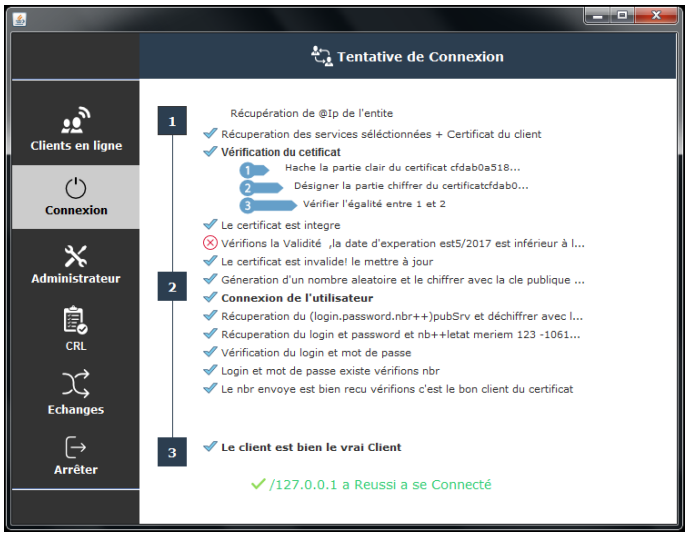

FIGURE 4.12 – Interface d'un certificat invalide

4.4.10 Interface connexion de l'utilisateur

Cette interface est la deuxième étape d'authentification après avoir envoyé le certificat, le client doit saisir son login et mot de passe a fin d'accéder son compte.

FIGURE 4.13 – Interface connexion de l'utilisateur

Cas1 : authentification réussie

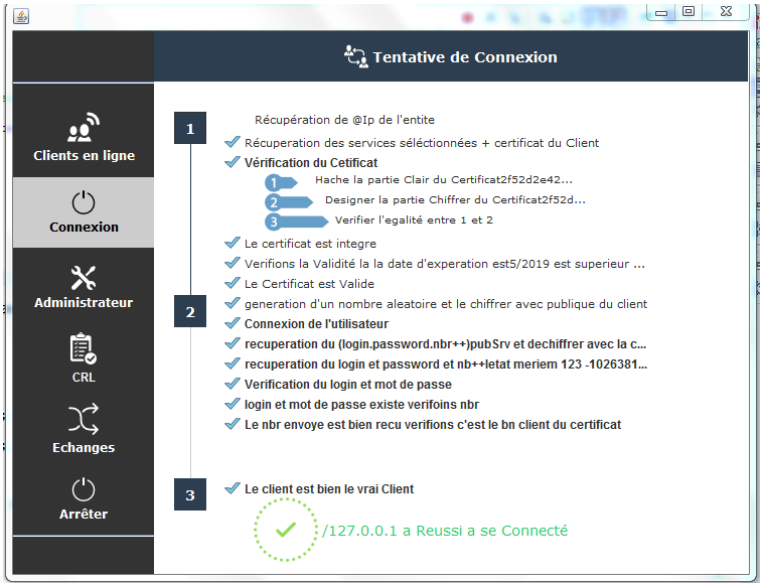

FIGURE 4.14 – authentification réussie

Cas2 : login et mot de passe incorrects

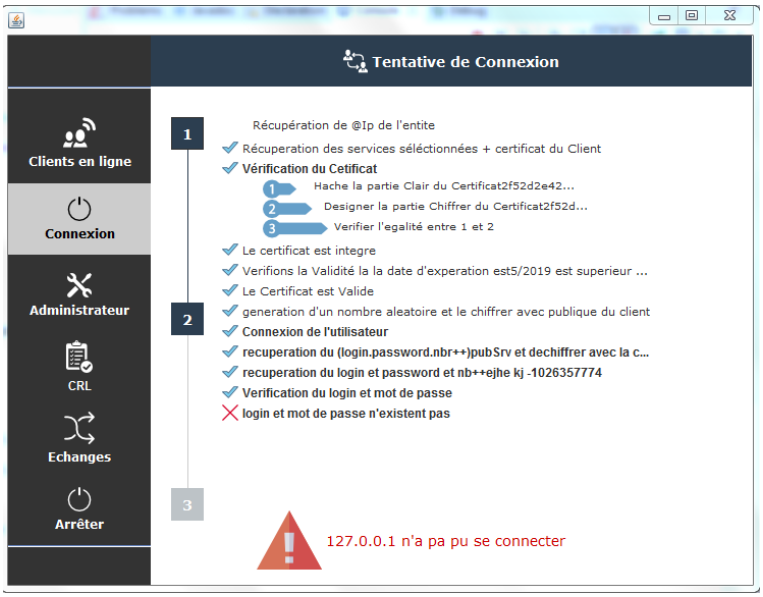

FIGURE 4.15 – authentification échoué

Cas3 : le nombre envoyé incorrect

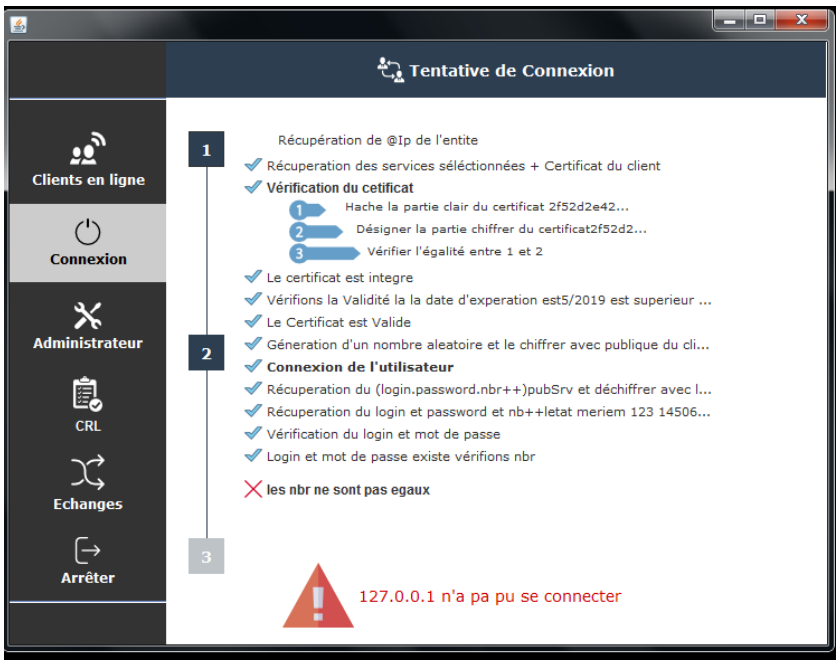

FIGURE 4.16 – authentification échoué

4.5 Conclusion

A travers ce chapitre, nous avons présenté la réalisation de l'application en justifiant nos choix technologiques, en représentant quelques interfaces graphiques que nous avons jugé les plus importantes et en décrivant brièvement comment nous avons planifié notre projet en terme de programmation.

Conclusion Générale et Perspectives

De nos jours, la sécurité informatique est quasi-indispensable pour le bon fonctionnement d'un réseau, aucune entreprise ne peut prétendre vouloir mettre en place une infrastructure réseau, quelque soit sa taille, sans envisager une politique de sécurité.
Nous avons tout au long de notre travail étudié les mécanismes de sécurité et les différentes techniques d'authentification, Suivi d'une étude détaillé de quelques protocoles d'authentification existants.

Le principe de notre travail est la mis en place un système sécurisé avec authentification unique tout en restant compatible avec les différentes technologies existantes, pour cela, nous nous sommes intéressés a la cryptographie hybride tout en utilisant les certificats et a une communication interprocessus Client/serveur en utilisant les sockets.

Notre proposition peut permettre aux utilisateurs de gagner en simplification d'accès à leur application. Il n'y a plus qu'une seule authentification par session de travail, la navigation entre applications ne nécessite plus de réauthentification. De plus si un portail d'accès multi-applications est mis en place, ils gagnent en ergonomie et en facilitation de basculement d'une application à une autre. Et surtout l'utilisateur ne possède qu'un seul mot de passe.

Pour terminer, nous tenons à souligner que nous n'avons nullement pas la prétention d'avoir présenté un travail parfait, car aucun travail scientifique ne peut l'être, ainsi nous laissons le soin à tous ceux qui nous lirons et qui sont du domaine de nous parvenir leurs remarques et suggestions pour l'enrichir et l'améliorer.

Comme perspectives de notre projet nous proposons :
- Décentraliser le système proposer afin d'avoir un système plus robuste et tolérant aux pannes ;
- Utilisation des méthodes de cryptage plus améliorer tel que les courbes elliptiques (ECC) pour avoir un système plus sécurisé. - utilisation de l'authentification biométrique ou avec reconnaissance faciale.

Bibliographie

[1] A.BIBA, F.QRABHERR,"Modélisation avec UML",InterEdition,2001.

[2] B.MARC, "A Survey of Public-key Infrastructures", Département d'informatique,Université McGill, Montréal, 1997.

[3] C.KAUFMAN,R RERLMAN,M .SPECINER ,"Network Security Private Communication In Public Word",Prentice Hall,Second Edition ,2003.

[4] C. SAILLARD ,"802.1X : Solution d'authentification sécurisée pour le futur réseau sans fil "Centre Réseau Communication, Université Louis Pasteur, Strasbourg.

[5] C.SIHAM,K.DAOUYA,"Conception et Réalisation d'un système D'authentification Distribué",UNIVERSIT2 DE EJAIA,FACULTE DES SCIENCE ET SCIENCE DE L'INGENIEUR,OPTION SYSTEME DISTRIBUES ET PARALLELES 2006-2007.

[6] E.BOUILLON, "Kerberos et la Sécurité ", Bruyéres-le-Chatel, France,2004.

[7] F.GALLO,"Méthodologie UML",Paris, Éditions eyrolles edition,200-2001.

[8] G. HARRY , "Failles De Sécurité Des Applications Web ",Business Media,2012.

[9] J. BOURIANES, C. LECLERC "Sécurité Des Echanges De Données Clients – Serveurs"école supérieur e génie informatique,2010 .

[10] J. KOHL , B. CLIFFORD NEUMAN, "The Kerberos Network Authentication Service (Version 5)", Internet Request for Comments RFC 1510, septembre 1993.

[11] K.PASIMAN'U PATIENCE " Etude pour la mise en place d'un systéme de paiment electronique de paimenet dans une institution financiére" Informatique de gestion Cycle de licence année 2011/2012.

[12] L.B L O C H , C. WO L F H U G E L, "Sécurité Informatique,Principles Et Methods",Edition 3,2004.

[13] L. SHKLAR,R. ROSEN ," Web Application Architecture : Principles, Protocols And Practices ",Springer Science et Business Media,2013.

[14] M.ABAKAR, "Étude et Mise en Œuvre d'une Architecture pour L'authentification et la Gestion de Document Numérique Certifie." thèse doctorale en Génie informatique,22 novembre 2012.

[15] M. BENZAID, B.MIHOUBI, N. NOUALI-TABOUDJEMAT," Mise en œuvre d'un système d'authentification avancée", Mémoire réalisé au Cerist(Centre de Recherche sur l'Information scientifique et Technique], N55/98, 1998.

[16] M. CHRISTERSON, P. JONSSON, G. OVERQARD,"Object-Oriented software engineering ! : A cas d'utilisation driven approach !", Addison-Wesley, 1992.

[17] M.MAKNEM, B.KHABOUCHI , Y.HAMMANI "Dévelopement d'un outil de supervision d'un systéme exploitation (tunisie télécom)" ,memoire de fin d'étude, Université Virtuelle de tunis 2010/2011.

[18] N. BAUDRU,"Sécurité Des Systèmes Informatiques",Bruyéres-le-Chatel, France, 2009.

[19] O. SALAÜN,"Introduction Aux Architectures Web De Single Sign-On ",Certicom Research, 15 Octobre 2003.

[20] P. SICARD "Documentation technique sur l'utilisation des Sockets ",Eyrolles, 2010.

[21] S.CHELOUAH, D.KABLI,"Conception et Réalisation d'un Système d'Authentification distribué ",memoire de fin d'étude, UNIVERSITE DE BEJAIA, FACULTER DES SCIENCE ET SCIENCE DE L'INGENIEUR, DEPARTEMENT INFORMATIQUE,OPTION :Système distribué et parallèles, DUNOD,2006/2007.

[22] S. VINSOT, "Les 7 Méthodes D'authentification Les Plus Utilisées ",Version 1.0,2010.

[23] V. NKIET ,E. DESBEAUX,Q.KEVIN ,"Le Cryptage PGP ",université PARIS-SUD11 2005-2007.

[24] W.MEJRI, M.SLIM ARAFA "Conception et realisation d'une application de gestion des comptes mail et internet " Licence appliqués en sciences et techniques de l'information et de communications, Université virtuelle de Tunis, 2012.

[25] Y. CHALLAL , "ICP/PKI : Infrastructures à Clés Publiques ",Université de Technologie de Compiègne Heudiasyc UMR CNRS 6599 France,2007.

[26] "Cryptographie à Clé Publique et Signature Numérique Principes de Fonctionnement",Étude technique réalisée par CGI Septembre 2002 .

[27] http ://www.securiteinfo.com/crypto/hash.shtml(consulter le 07/02).

[28] http ://home.ecn.ab.ca/ jsavard/crypto/mi0605.htm (consulter le 07/02).

[29] http ://www.rsasecurity.com/rsalabs/faq/(consulter le 19/03).

[30] http ://www.ietf.org/rfc(consulter le 20/03).

[31] http ://web.mit.edu/kerberos/www/(consulter le 28/03).

[32] http ://www.openssl.org(consulter le 17/04).

[33] https ://hal.archives-ouverture.fr/hal-00561730 jean-luc archimbaud Certificats(electronique) :pourquoi ?Comment ?2 (consulter le 29/03).