UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences

Département d'informatique

MEMOIRE DE MASTER

En Informatique

Option : Sécurité de système d'information

Mise en place d'une solution de sécurité pour l'intégration de la signature électronique dans CNTSID

Réalisé par :
BEHALIL Bouthaina
BELAZIZ Amel

Encadré par :
Mme. BEY Fella (Saad Dahleb)
M. MEZIANE Yasine (CNTSID)

Soutenu le 06 juillet 2022, Devant le jury composé de :

Mme. N. Boustia:	Université Blida 1	- Présidente
Mme. F. Bey :	Université Blida 1	- Examinateur
Mr. Sahnoun	Université Blida 1	- Rapporteur

Promotion : 2021/2022

Remerciements

Avant tout nous remercions dieu le tout puissant qui nous a donné la force, la patience et le courage pour qu'on puisse accomplir ce modeste travail.

En préambule à ce mémoire, il nous est agréable de citer et adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leurs aides et qui ont contribué à l'élaboration et au bon déroulement de ce travail :

A notre promotrice madame Bey Fella.

A notre encadreur monsieur Meziane Yasine.

Nous exprimons nos reconnaissances à tous personnes qui a contribué de près ou de loin à l'achèvement de ce travail, nos parents, nos enseignants, nos amis.

Nous remercions également les membres de jury d'avoir accepté juger ce modeste travail.

Dédicace

Je dédie ce travail à tous ceux que j'aime.

À MON CHER PÈRE

Qui est le meilleur père dans ce monde, qui n'a jamais cessé de me fournir son assistance et son amour infini, il m'a appris le vrai sens de la vie, du respect et de générosité. Grâce à son encouragement, ses sacrifices qu'il a consenti pour mon succès, sa confiance et son soutien moral et matériel et en exprimant mes gratitudes, mon profond amour et ma passion.

À MA CHÈRE MÈRE

Qui a veillé sur moi et a tout souffert sans me faire souffrir. Pour votre amour,votre patience et votre générosité, pour tous les efforts que vous avez consentis en ma faveur. J'espère avoir été digne de votre affection et de votre confiance.

À MES CHÈRES SOEURS "HADIL ET NOUR EL HOUDA" ET MON CHER FRÈRE "YASEER ABD EL HADI"

Merci pour vos encouragements . Je vous souhaite une vie pleine de succès, de joie et que Dieu nous garde unis.

À toute ma famille.
Merci.

- Bouthaina

Dédicace

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite.

À MES CHERS PARENTS

qui m'ont éclairé le chemin de la vie par leur grand soutien et leurs encouragements, par leurs dévouements exemplaires et les énormes sacrifices qu'ils m'ont consentis durant mes études et qui ont toujours aimé me voire réussir.

À MES CHÈRES SOEURS "KHADIDJA ET MALIKA" ET MON CHER FRÈRE "ISLEM"

Merci pour vos encouragements . Je vous souhaite une vie pleine de succès, de joie et que Dieu nous garde unis.

À toute ma famille.
Merci.

- Amel

Résumé

La digitalisation des processus administratifs intéresse de nombreuses entreprises. Les petites et moyennes entreprises ont eu recours à cette transformation. la signature électronique est un mécanisme permettant d'assurer l'intégrité des documents électroniques. Autrement dit un moyen de sécurité destiné à garantir l'identité du signataire ainsi que l'intégrité et la confidentialité du document tout en repose sur la cryptographie asymétrique et les certificats numériques.DegiSign est une solution pour l'intégration de la signature électronique dans le Centre National des Transmissions et du système d'information des Douanes.

Mots clés : signature électronique,intégrité,document électronique,sécurité,cryptographie asymétrique,certificats numérique.

Abstract

The digitalization of administrative processes is of interest to many companies. Small and medium-sized companies have resorted to this transformation. The digital signature is a mechanism to ensure the integrity of electronic documents. In other words, it is a means of security designed to guarantee the identity of the signatory as well as the integrity and confidentiality of the document while relying on asymmetrical cryptography and digital certificates. DegiSign is a solution for integrating the electronic signature into Center of Transmissions and the Information System of Customs.

Keywords : digital signature,integrity,electronic documents,security,asymmetric cryptography,digital certificates.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

API Interface de Programmation Applicative. 49

CNTSID Centre National des Transmissions et du Système d'Information des Douanes. 15

DES Data Encryption Standard. 21

IDE Integrated Development Environment. 48

JCA Cryptography Architecture Java. 49

JCE Java Cryptography Extension. 49

MD-SHA Message Digest-Secure Hash Algorithm. 26

MD4 Message Digest 4. 26

MD5 Message Digest 5. 26

MVC Model-View-Controller. 49

NIST National Institute of Standards and Technology. 27

NSA National Security Agency. 26

PGP Pretty Good Privacy. 24

PKCS Public-Key Cryptography Standards. 24

PKI Public Key Infrastructure. 59

RIPEMD RIPE Message Digest. 26

RSA Ronald Rivest, Adi Shamir, Leonard Adleman. 23

SHA Secure Hash Algorithm. 27

SHA-0 Secure Hash Algorithm 0. 27

SHA-1 Secure Hash Algorithm 1. 27

SHA-2 Secure Hash Algorithm 2. 27

UML Unified Modeling Language. 36

UP Unified Process. 16

XAMPP Cross-Platform (X), Apache (A), MySQL (M), PHP (P) et Perl(P). 47

Introduction Générale

La modernisation de l'administration Algérienne a pour effet d'améliorer les services.La numérisation a joué un rôle très important dans la modernisation,ce qui a engendré des impacts sur la gestion des administrations.L'administration électronique touche en fait aux systèmes d'informations, ç'est à dire aux modalités d'échanges et de conservation d'informations de toute nature entre plusieurs entités grâce à l'utilisation des technologies de l'information dans tous les domaines de l'administration.

Depuis plusieurs années, la digitalisation des processus administratifs intéresse de nombreuses entreprises. Les petites et moyennes entreprises ont eu recours à cette transformation afin d'améliorer les trois critères : la qualité, la rapidité et la maîtrise des coûts.

Le Centre National des Transmissions et du Système d'Information des Douanes (CNTSID) envisage de tirer parti des avantages de la numérisation pour gagner du temps, améliorer les services et constituer un facteur de réduction des coûts. Dans toute l'organisation, la dématérialisation des tâches est le véritable moyen d'atteindre l'efficacité, la transparence et la confiance.

La dématérialisation de nombreuses démarches administratives est en phase de mise en œuvre, ce qui se traduit par la numérisation des actions de la structure et entre autres, intégration de la signature numérique dans l'échange de documents administratifs. Grâce à une telle initiative, il ne s'agit plus seulement de vérifier l'identité d'un individu ou de s'assurer que seules les personnes autorisées ont accès aux informations. Il faut désormais également s'assurer que d'une part, les données n'ont pas été falsifiées par un tiers et d'autre part, que l'organisation ou la personne impliquée dans l'échange ne peut pas nier son rôle dans l'échange.

Le travail qu'a été demandé consiste à proposer et réaliser un système de signature électronique des documents internes de CNTSID pour objectif de :

- Garantir l'intégrité d'un document, ç'est à dire de s'assurer que le document n'a pas été altéré entre sa signature et sa consultation
- Vérifier l'authentification, ce qui garantit l'identité du signataire.

La solution développée consiste à signer des documents à l'aide de techniques cryptographiques et à garantir l'identité du signataire grâce à l'utilisation de certificats électroniques. Nous utilisons la norme PKCS :la première PKCS1 pour le cryptage RSA et la deuxième PKCS12 pour le transfert d'informations personnelles d'identité. Nous utilisons également SHA-256 pour le hachage, l'algorithme RSA pour le chiffrement et les fournisseurs de services cryptographiques en langage Java.

Le mémoire est organisé en quatre chapitres :

Dans le premier chapitre, nous présenterons les objectifs de la sécurité. Puis, nous présenterons quelques généralités sur la cryptologie. Nous présenterons, ensuite la signature électronique, ainsi que le hachage et le certificat électronique .

Dans le deuxième chapitre, nous aborderons l'analyse et spécification des besoins de CNTSID. nous commençons par l'identification des besoins. Nous détaillerons, ensuite, la descreption de système réalisé. Puis nous présenterons les spécifications fonctionneles , non fonctionnels et techniques.

Dans le troisième chapitre, nous présenterons les concepts du langage de modélisation unifié UML et ses diagrammes, ainsi que les différentes étapes de la méthode de développement, inspirée des étapes du processus Unified Process (UP), que nous avons choisi pour la conduite de notre projet. Nous expliquerons, ensuite les fonctionnalités de notre application et nous détaillerons les diagrammes de cas d'utilisation, les diagrammes de séquence,le diagramme de classe,les diagrammes d'activités.

Dans le quatrième chapitre, nous présenterons la réalisation de notre projet. Nous commencerons par la présentation de l'environnement logiciel utilisé pour la réalisation du système qui est composée essentiellement du langage de programmation JAVA, l'environnement de développement NetBeans et le système de gestion de base de données Mysql. Et enfin, nous présenterons quelques interfaces de l'application.

Nous terminons ce mémoire avec une conclusion générale et quelque perspectives.

Chapitre 1

Généralité

1.1 Introduction

La signature électronique est une technique simple et efficace dans le monde numérique d'aujourd'hui grâce aux les techniques de cryptographie moderne qui permettent de sécuriser efficacement les systèmes informatiques, les communications et l'information. Retenons que la cryptographie permet la conception de la signature électronique et lui attribue l'authenticité, l'intégrité et la confidentialité via ses outils de chiffrement. La connaissance de ces concepts est donc indispensable pour le développement de ce système de signature.Dans ce chapitre nous allons définir les principes essentiels dans la cryptographie.

1.2 Les objectifs de la sécurité

Assurer la sécurité d'un système informatique revient à atteindre un ensemble d'objectifs afin de garantir la protection des informations contre toute divulgation, altération ou destruction.[31]

a- La confidentialité : seuls les utilisateurs autorisés ont l'accés à l'information.

b-L'intégrité : : Assurer que les informations n'ont pas été modifiées(ou altérées) par des entités non autorisées ou inconnues. Généralement,l'intégrité est appliquée sur des données en transmission.[31]

c-L'authenticité : identifier la personne ou l'organisme qui a effectué l'action.

d-La disponibilité : les ressources sont disponibles à tout moment pour les personnes autorisées.

e-La non-répudiation : ne permet pas au propriétaire d'une information de réfuter la possession de cette information(chaque individu est responsable de ses actions).

1.3 La cryptologie

La cryptologie est un mot composé qui vient du grec : cryptos signifiant secret et logy signifiant science. En effet, c'est la science du secret et ne peut être vraiment considérée ainsi que pendant une courte période. C'est une science mathématique à deux branches : la cryptographie, l'écriture secrète, et la cryptanalyse, l'analyse de cette dernière. La dénomination générale de cryptologie pour désigner la partie de la sécurité des systèmes d'information qui consiste à assurer ou autrement compromettre les principaux objectifs de la sécurité(notamment la confidentialité, l'intégrité des données, l'authentification et la non-répudiation).[5]

1.3.1 Cryptanalyse

La cryptanalyse est l'utilisation de méthodes mathématiques pour reconstruire sans ambiguïté des messages cryptés. Par conséquent, tout cryptosystème doit être résistant aux méthodes de cryptanalyse. Lorsqu'une méthode cryptanalytique peut décrypter un message crypté à l'aide d'un cryptosystème, on dit que l'algorithme de cryptage a été cassé. Il existe généralement quatre méthodes de cryptanalyse.[22]

a-Une attaque sur texte chiffré seulement : consiste à trouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés.[22]

b-Une attaque sur texte clair connu : consiste à trouver la clé de déchiffrement à partir d'un ou plusieurs chiffrés, connaissant le clair correspondant.[22]

c-Les attaques sur des textes clair choisi : consistent à trouver des clés de déchiffrement à partir d'un ou plusieurs textes chiffrés, les générant potentiellement à partir du texte en clair par un attaquant.[22]

d- Les attaques sur des textes chiffrés choisis : consistent à trouver des clés de déchiffrement à partir d'un ou plusieurs textes chiffrés, les générant potentiellement à partir du texte en clair par un attaquant.[22]

1.3.2 Cryptographie

La cryptographie est l'étude des méthodes de communication sécurisée entre deux parties.Typiquement, il y a deux parties qui veulent s'envoyer des messages, en évitant la possibilité qu'un tiers comprenne ces messages devrait ils utilisent un ensemble des techniques permettant de chiffrer les messages, afin de devenir incompréhensibles sans opérations spécifiques.il est intrinsèquement basé sur l'arithmétique. Dans le cas du texte, il faut convertir les lettres qui composent le message en une série de chiffres (en bits).[24]

1.3.3 Les objectifs de la cryptographie

La cryptographie est utilisée pour masquer les messages de certains utilisateurs. Cette utilisation est d'autant plus intéressante aujourd'hui que les communications sur Internet circulent dans des infrastructures où la fiabilité et la confidentialité ne peuvent être garanties. Désormais, la cryptographie est utilisée non seulement pour protéger la confidentialité des données, mais aussi pour garantir l'intégrité et l'authenticité des données.[22]

1.3.4 La cryptographie classique

La cryptographie classique concerne la période antérieure aux ordinateurs. Ce sont des systèmes basés sur les lettres et les caractères d'une langue naturelle (allemand, anglais, français, etc.). Les principaux outils utilisés remplacent les caractères par d'autres et les transposent dans des ordres différents. Les meilleurs systèmes (de cette classe d'algorithmes) répètent plusieurs fois ces deux opérations de base.[6]

La cryptographie classique est l'étude des méthodes permettant la transmission des données de manière confidentielle. Pour protéger un message, une transformation dépendant d'un paramètre, appelée clé, lui est appliquée pour le rendre inintelligible (chiffré). Le déchiffrement est l'opération inverse de la reconstruction du texte en clair à partir du texte chiffré. La plupart des méthodes de chiffrement reposent essentiellement sur deux moyens la transposition et la substitution .[1]

1.3.4.1 Chiffrement par substitution

Le chiffrement par substitution consiste à remplacer une ou plusieurs lettres dans un message par une ou plusieurs autres lettres. On distingue généralement plusieurs types de cryptosystèmes par substitution :[6]

a-La substitution mono-alphabétique : consiste à remplacer une lettre dans le message par une autre lettre de l'alphabet. [22]

b-Les substitutions poly-alphabétique : incluent l'utilisation des séquences alphanumériques simples répétées régulièrement. [22]

c-La substitution homophonique : permet de faire correspondre chaque lettre d'un message en clair avec un éventuel ensemble d'autres caractères. [22]

d-La substitution polygrammes : consiste à remplacer un jeu de caractères d'un message par un autre jeu de caractères.[22]

Chiffrement de César : c'est un système simple par substitution mono-alphabétique consistant à décaler les lettres de l'alphabet.Le chiffrement est limité par le nombre de lettres de l'alphabet.Il est facile de retrouver le message d'origine en essayant tous les décalages.[18]

Par exemple, en déplaçant le message "MASTER SSI" de trois places, on obtient "PDVWHU VVL". Lorsque l'addition de valeurs donne plus de lettres que la lettre Z, il suffit de commencer par A.

1.3.4.2 Chiffrement par transposition

Une des premières techniques cryptographiques est le chiffrement par transposition,qui consiste à mélanger les symboles ou les groupes de symboles d'un message clair suivant des règles prédéfinies pour créer de la diffusion. Ces règles sont déterminées par la clé de chiffrement. Une suite de transpositions forme une permutation. Un des premiers exemples connus d'un tel chiffrement est la scytale spartiate. [10]

Le chiffrement assyrienne : utilisée au Vème siècle avant J-C par les grecs. La scytale consiste en un bâton, autour duquel est enroulée une lanière de cuir. L'expéditeur écrit son message sur la lanière, puis une fois terminé la déroule et l'envoie. Le récepteur enroule à son tour la lanière reçue sur un bâton de même diamètre, ce qui lui permet ainsi de retrouver le texte original.[13]

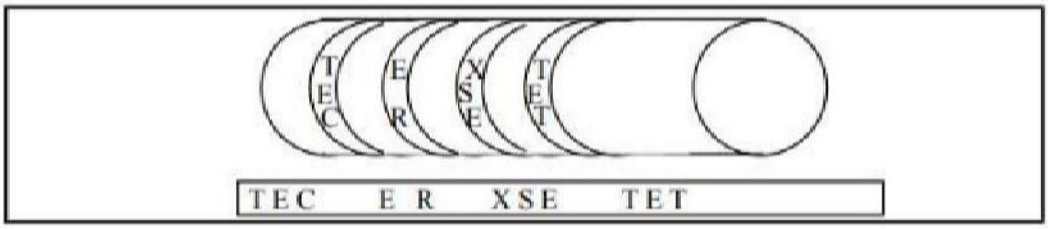

FIGURE 1.1 – chiffrement assyrienne.[3]

1.3.5 La cryptographie moderne

La cryptographie moderne est divisée en deux sous-familles principales. La plus ancienne est la cryptographie symétrique. L'idée principale est que l'expéditeur et le destinataire d'un message partagent un secret, appelé clé secrète. En 1976, Diffie et Hellman ont introduit la cryptographie asymétrique, également connue sous le nom de cryptographie à clé publique.[12]

1.3.5.1 Le chiffrement symétrique

Le chiffrement symétrique consiste à utiliser la même clé pour le chiffrement et le déchiffrement.

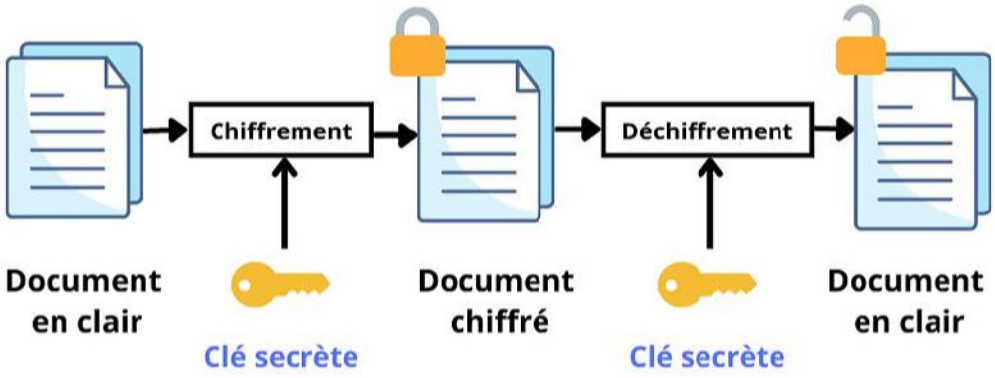

FIGURE 1.2 – chiffrement symétrique

Ce type de chiffrement consiste à utiliser une clé privée pour appliquer des opérations aux données à chiffrer afin de les rendre incompréhensibles. Ainsi le moindre algorithme peut rendre le système quasiment inviolable (la sécurité absolue n'existe pas). Le principal inconvénient des cryptosystèmes à clés provient de l'échange de clés.

En effet, le chiffrement symétrique repose sur l'échange de secrets et pose le problème de la distribution des clés. En revanche, les utilisateurs souhaitant communiquer avec plusieurs personnes tout en assurant différents niveaux de confidentialité doivent utiliser autant de clés privées que d'interlocuteurs.[22]

LE chiffrement DES : Data Encryption Standard (DES),est l'archétype du chiffrement par blocs, un algorithme qui prend une chaîne de bits de texte en clair de longueur fixe et la transforme par une série d'opérations compliquées en une autre chaîne de bits de texte chiffré de même longueur.[25]

Le DES utilise également une clé pour personnaliser la transformation, de sorte que le décryptage ne peut être effectué que par ceux qui connaissent la clé particulière utilisée pour le cryptage. Dans le cas de DES, la taille du bloc est de 64 bits , mais seuls 56 bits sont utilisés et les 8 bits restants peuvent être utilisés pour la parité, puis rejetés dans l'algorithme. Par conséquent,

la longueur de clé effective de DES est de 56 bits. La structure globale de l'algorithme est illustrée à la Figure. [25]

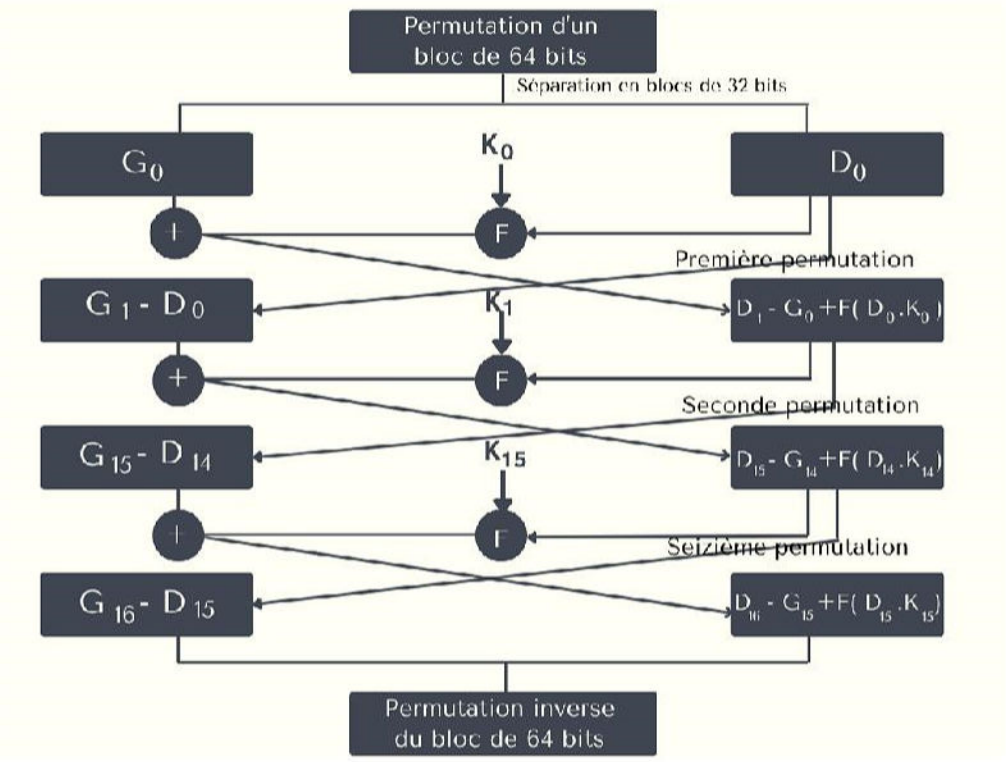

FIGURE 1.3 – chiffrement DES [23]

1.3.5.2 Chiffrement asymétrique

Dans un cryptosystème asymétrique, les clés existent par paires, une clé publique pour le chiffrement et une autre clé pour le déchiffrement. L'utilisateur choisit une clé aléatoire qu'il est le seul à connaître.Lorsqu'un utilisateur souhaite envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer à l'aide de la clé publique du destinataire, et le destinataire pourra déchiffrer le message à l'aide de sa clé privée. Ainsi, le cryptage asymétrique permet aux personnes d'échanger des messages cryptés sans posséder de secret commun.La structure globale est illustrée à la Figure.[22]

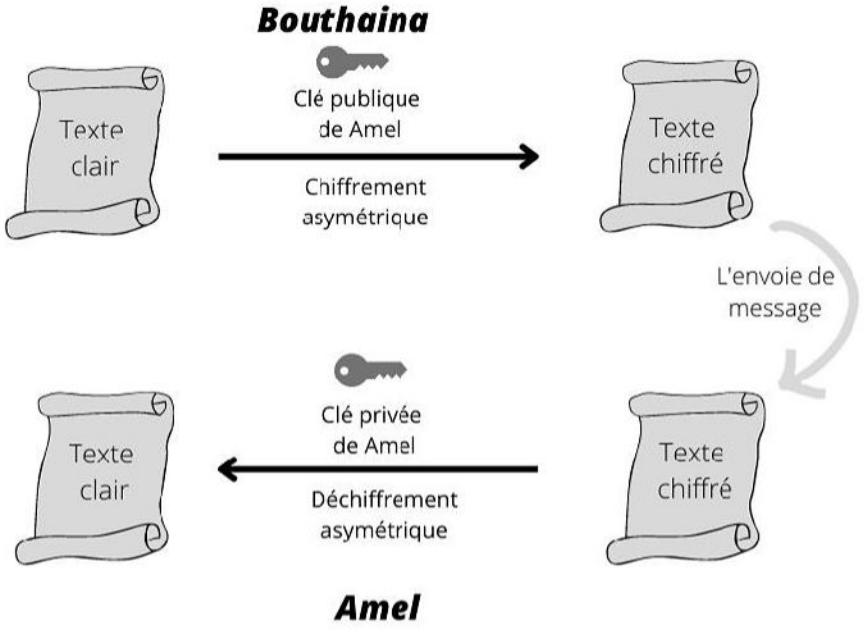

FIGURE 1.4 – chiffrement asymétrique

Chiffrement RSA : Ronald Rivest, Adi Shamir, Leonard Adleman (RSA) est un algorithme de chiffrement à clé publique. Il a été nommé d'après le nom de ses inventeurs : Ron Rivest, Adi Shamir et Leonard Adleman.La sécurité est assurée par le fait qu'il est très difficile de décomposer de très grands nombres en facteurs premiers.[4]
On choisi deux nombres premiers p et q. On calcule leur produit, que l'on appellera modulus de RSA : n = p * q. On choisi ensuite une clé de chiffrement aléatoire e de telle manière que e et phi(n) = (p - 1)(q - 1) soient premiers entre eux. On utilise ensuite l'algorithme d'Euclide pour calculer la clé de déchiffrement d telle que : d est équivalent à e^{-1} (mod (p - 1)(q - 1)). On peut constater que d et n sont aussi premiers entre eux. Le nombre e sera la clé publique de RSA, et d sera la clé privée. Les nombres p et q ne seront plus utilisés mais ne doivent en aucun cas être révélés. Soit m le message à chiffrer. Ce message m devra être plus petit que n. Le chiffrement se fait en utilisant l'équation suivante : $c = m^e \bmod n$ Le déchiffrement s'effectue à l'aide de la clé privée $m = c^d \bmod n$.[4]

1.3.5.3 Chiffrement hybride

C'est un mélange de deux types de cryptographie symétrique et asymétrique. Le message est chiffré avec une méthode symétrique à l'aide d'une clé secrète, cette clé est ensuite cryptée de manière asymétrique séquentiellement puis envoyée au destinataire. Le destinataire commence par déchiffrer la clé puis utilise cette clé pour décoder le message.[20]

Chiffrement PGP : Pretty Good Privacy (PGP), créé par Phil Zimmermann en 1991. Elle est bien conçue pour fournir des services d'authentification, d'intégrité et de confidentialité qui peuvent être utilisés pour sécuriser les courriers électroniques et les applications de stockage de fichiers.[17]

Zimmermann a sélectionné certains des meilleurs algorithmes cryptographiques disponibles et les a intégrés dans une application générale indépendante de la plate-forme. Ainsi, selon PGP, il s'agit d'un cryptosystème hybride. Le processus de génération de messages PGP est une combinaison en série de hachage,de signature du hachage, de compression de données, de cryptographie à clé symétrique et enfin, cryptographie à clé publique, appliquées sur le courriel, l'une après l'autre. [17]

1.3.6 Les standards de cryptographie à clé publique "PKCS"

La norme de cryptographie à clé publique est une spécification rédigée par RSA Laboratories en collaboration avec des développeurs de systèmes de sécurité du monde entier. Publiés pour la première fois en 1991 par un petit groupe d'utilisateurs précoces de la technologie à clé publique, les fichiers PKCS ont été largement référencés et mis en œuvre. Les contributions à la famille Public-Key Cryptography Standards (PKCS) ont été utilisées dans plusieurs formats et sont devenues des normes de facto, telles que les documents ANSI X9, PKIX, SET, S/MIME et SSL.[30]

PKCS #1	Cryptographie RSA
PKCS #3	Accord sur clé Diffie-Hellman
PKCS #5	Cryptographie par mot de passe
PKCS #6	Syntaxte des extensions de certificat
PKCS #7	Syntaxe des messages de cryptographie
PKCS #8	Syntaxte d'information sur les clés privées
PKCS #9	Types d'attributs choisis
PKCS #10	Syntaxe de demande de certification
PKCS #11	Interface de jeton cryptographique
PKCS #12	Syntaxe d'échange d'informations personnelles
PKCS #13	Cryptographie à courbe elliptique
PKCS #15	Formattage des informations sur les jetons de cryptographie

TABLE 1.1 – Les spécifications PKCS.[30]

1.4 La signature électronique

Les signatures électroniques sont un autre mécanisme qui fournit des fonctions d'authentification et d'intégrité. Il est surtout utilisé pour le courrier électronique. Pour générer une signature électronique, vous devez d'abord utiliser une fonction de hachage sur le texte, et le résultat est une séquence de bits de taille fixe, beaucoup plus petite que la taille du texte d'origine. Cette séquence de bits est également appelée condensé ou hachage, car la fonction de hachage est telle qu'il y a de fortes chances que le résultat de la fonction soit différent si un bit du texte original était modifié.[2]

MD5 (Message Digest) et SHA (Secure Hash Algorithm) sont des algorithmes de hashage les plus connus. Pour effectuer une signature électronique, l'outil de messagerie calcule d'abord le hash du message avant de l'envoyer.Il crypte ensuite cette empreinte digitale à l'aide d'un algorithme asymétrique et de la clé privée de l'utilisateur. Ce résultat s'appelle une signature électronique. Cette signature est ajoutée au message avant l'envoi, ce qui en fait un message signé.[2]

Destinataire déchiffre cette empreinte chiffrée à l'aide de la clé publique de l'expéditeur. Il recalcule ensuite la fonction de hachage du message reçu et compare le résultat avec le hachage déchiffré. Si les deux sont égaux, le message n'a pas été modifié en transit et l'expéditeur est authentifié.[2]

En effet, les 2 empreintes digitales seront différentes si le message est modifié en transit. De plus, le fait de pouvoir déchiffrer une empreinte chiffrée avec sa clé publique prouve que l'empreinte doit avoir été chiffrée avec la clé privée de cette personne, clé que seul possède l'émetteur. Cela authentifie donc l'émetteur.[2]

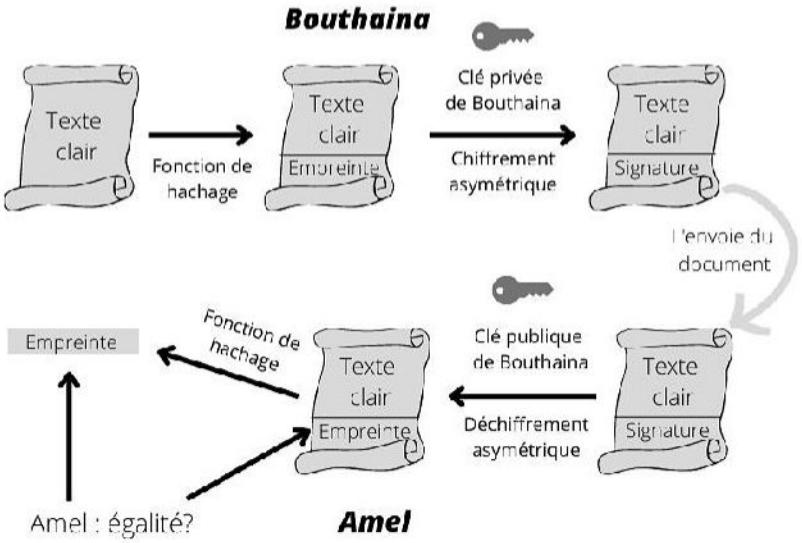

FIGURE 1.5 – Signature électronique

1.5 Le hachage

Les fonctions de hachage servent à calculer à partir d'une donnée de taille arbitraire fournie en entrée une empreinte de taille fixe. Cette taille varie en général entre 128 et 512 bits. Cette empreinte, appelée aussi condensé ou simplement haché doit dépendre de tous les bits du message et est utilisée pour représenter le message de façon compacte.[7]

Une fonction de hachage est un algorithme entièrement public qui n'implique de valeurs secrètes à aucun moment du calcul. Cependant, les fonctions de hachage appartiennent à la famille des algorithmes symétriques car leur construction est très similaire à celle des chiffrements par blocs. Une fonction de hachage devrait idéalement se comporter comme une fonction aléatoire. Dans le même temps, de nombreuses propriétés doivent être respectées. En particulier, il doit être difficile de trouver des fonctions de collision ou d'inversion.[7]

1.5.1 Les fonctions de hachage

Les origines de cette famille remontent à la conception de la fonction Message Digest 4 (MD4), introduite par Rivest en 1990. L'attaque de la version réduite a été rapidement trouvée par den Boer et Bosselaers, et Rivest a proposé Message Digest 5 (MD5) en version améliorée. Bosselaers lui-même a proposé RIPE Message Digest (RIPEMD) en 1992 comme alternative à MD4,et National Security Agency (NSA) fit de même l'année suivante en introduisant la première génération des algorithmes SHS/SHA . Dans les deux derniers cas, les fonctions ont été modifiées peu de temps après, en 1996 et 1995 respectivement. D'autres algorithmes, tels que SHA-2, introduit en 2002, ont également été influencés par MD4. [15]

Il existe plusieurs fonction de hachage représenter dans les points suivants :

1.5.1.1 MD4

Cette fonction de hachage peut être considéré comme le début de la famille Message Digest-Secure Hash Algorithm (MD-SHA). En effet, les principes qui ont guidé la conception de sa fonction de compression sont largement repris par les fonctions suivantes. Cependant, cette fonctionnalité a été peu utilisée et a été très rapidement remplacée par MD5.[11]

1.5.1.2 MD5

Le MD5 a été conçu par Rivest en 1991 . Cette fonctionnalité a été largement utilisée et est toujours disponible dans de nombreux produits et protocoles. Ceci est dû à la modification de MD4 : 16 étapes ont été ajoutées à la fonction de compression, l'étape élémentaire a été légèrement modifiée.[11]

1.5.1.3 SHA-0

Secure Hash Algorithm 0 (SHA-0) est la première fonctionnalité de la famille Secure Hash Algorithm (SHA), conçue par la NSA en 1993 et normalisée par National Institute of Standards and Technology (NIST) avant d'être rapidement supprimée et remplacée par SHA-1. Les choix faits pour sa conception n'ont pas été publiquement expliqués, mais les principes généraux reprennent largement ceux de MD4 et MD5. SHA-0 permet de calculer des empreintes de 160 bits, ce qui représente un gain substantiel quant à la complexité des attaques génériques par rapport aux fonctions précédentes. Dès 1995, le NIST a publié la fonctionnalité SHA-1, qui est détaillée ci-dessous, qui est une amélioration mineure par rapport à SHA-0. [11]

1.5.1.4 SHA-1

Secure Hash Algorithm 1 (SHA-1) a été publié en 1995 par le NIST pour pallier les faiblesses de SHA-0 . Nous allons maintenant décrire les détails de sa fonction de compression. Ce choix est motivé par deux raisons :[11]

a– Est une cible représentative de diverses techniques de cryptanalyse mises en œuvre contre la famille MD-SHA.[11]

b– Malgré les vulnérabilités connues conduisant à son remplacement progressif par Secure Hash Algorithm 2 (SHA-2), il reste probablement la fonctionnalité la plus utilisée en pratique aujourd'hui.[11]

1.5.1.5 SHA-2

En 2002 le NIST a normalisé une nouvelle famille de fonctions de hachage, SHA-2 . Quatre versions de cet algorithme permettent de calculer des empreintes de 224, 256, 384 et 512 bits.[11]

Comme ses prédécesseurs, SHA-2 utilise l'algorithme d'extension de domaine de Merkle-Damgård et la construction de Davies-Meyer. La taille des variables de chaînage est de 256 bits pour les deux premières versions et de 512 bits pour les deux dernières. Les blocs de message correspondants ont pour tailles respectives 512 et 1024 bits. De manière générale, SHA-224 et SHA-256 utilisent des registres de 32 bits, et SHA-384 et SHA-512 utilisent des registres de 64 bits. La structure générale des opérations effectuées est similaire entre ces quatre fonctions.[11]

1.6 Certificat électronique

Un mécanisme supplémentaire doit être créé pour vérifier la validité des clés publiques : les certificats électroniques. Un certificat personnel électronique est l'équivalent électronique d'une carte d'identité ou d'un passeport. Le passeport contient des informations sur son titulaire (nom, prénom, adresse, etc.), signature manuscrite, date d'expiration, cachet et présentation

reconnaissables (forme, couleur, papier) Ce passeport n'est pas un faux, qu'il a été délivré par une autorité bien connue. [2]

Un certificat électronique est un petit fichier contenant des informations similaires. Le format standard actuellement est le format X509v3. Les certificats contiennent deux parties :partie contenant les informations et l'autre partie contenant la signature de l'autorité de certification.Toutes ces informations sont signées par l'autorité de certification, ce qui signifie qu'une fonction de hachage crée une empreinte de ces informations, qui est ensuite compressée et chiffrée à l'aide de la clé privée du demandeur, la clé publique ayant été préalablement Largement distribué pour permettre aux utilisateurs de vérifier les signatures à l'aide de la clé publique de l'autorité de certification. [2]

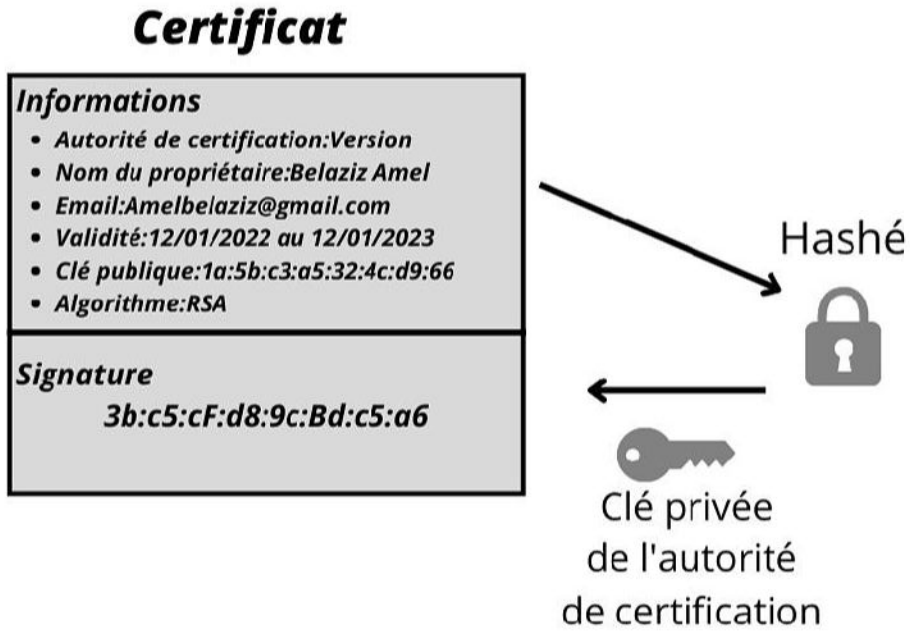

FIGURE 1.6 – signature électronique du certificat

1.7 Conclusion

La signature électronique repose sur deux familles d'algorithmes, qui sont utilisées de manière complémentaire aux algorithmes de chiffrement asymétriques et aux fonctions de hachages. La signature électronique va faire appel à ces deux familles d'algorithmes, afin de pouvoir garantir l'authenticité et l'intégrité d'un document électronique.

Dans ce chapitre, nous avons présenté la cryptographie et ses concepts de base,la signature électronique, certificat électronique et les fonctions de hachage.Dans le chapitre suivant on entame l'analyse et la spécification des besoins.

Chapitre 2

Analyse et spécification des besoins

2.1 Introduction

Un travail ne se fait pas en vain, il doit accomplir des tâches et satisfaire des besoins. Il est donc trivial de penser que la définition des exigences assure une part non négligeable de la réussite du projet.

Dans ce chapitre, on entame l'analyse et la spécification des besoins afin de déterminer et préciser les différentes fonctionnalités attendues de notre application. Nous identifions et décrivons d'abord les exigences, puis définissons les besoins fonctionnels, non fonctionnels et techniques de l'application.

2.2 La description d'état d'accueil

Le centre National des Transmissions et du Système d'Information des Douanes est créé auprés de la direction générale des douanes placer sous l'autorité du directeur général des douanes et dirigé par un directeur de centre, d'après le journal officiel de la republique Algérienne N°13 le 26 février 2017,le centre a pour missions de :

- Collecter les besoins des services des douanes en matière de technologies de l'information et de la communication, de confectionner les cahiers des charges techniques et fonctionnels y afférents et de suivre l'exécution des programmes et des contrats d'acquisition.
- Èstablir des interfaces avec les systèmes d'informations des autres intervenants de la chaîne logistique du commerce international.
- Ètudier les conditions d'implantation des stations des transmissions et du système d'information et de leur fonctionnement continu sur l'ensemble des services des douanes.
- Définir et de préciser le régime de travail en matière d'exploitation et d'utilisation des technologies de l'information et de la communication et, de veiller à son application.
- Veiller à la sécurité des technologies de l'information et de la communication en douane.

Le centre est organisé en quatre sous-directions illustrées dans la figure suivante :

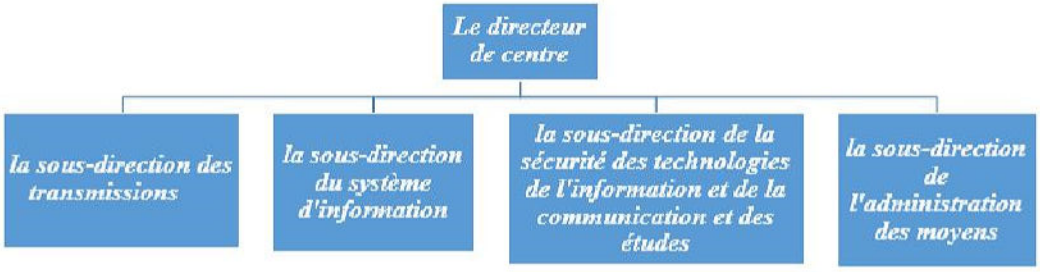

FIGURE 2.1 – L'organigramme de CNTSID

La sous-direction des transmissions, est chargée, notamment : de gérer et d'exploiter les réseaux des télecommunications et de veiller à l'application et au respect des règles de l'exploitation, conformément aux prescriptions en vigueur.

La sous-direction du système d'information, est chargée, notamment : de veiller à l'harmonisation des logiciels et équipements des technologies de l'information et de la communication avec ceux des institutions de l'etat dans l'optique de leur interopérabilité et mutualisation, de codifier, de développer et d'administrer les banques de données du système d'information des douanes , de développer et de maintenir les logiciels d'automatisation de l'activité de l'administration des douanes aussi bien de métier que de soutien de développer les sites intranet et internet des douanes et de veiller à leur mise à jour continue et automatique.

La sous-direction de la sécurité des technologies de l'information et de la communication et des études, est chargée, notamment d'étudier, d'élaborer et de veiller à l'exécution des procédures de sécurité arrêtées en matière de technologies de l'information et de la communication, notamment celles relatives au système d'information et aux centres radioélectriques.

La sous-direction de l'administration des moyens, est chargée, notamment : de gérer les moyens du centre et de gérer le personnel et de proposer les mesures qui en permettent la stabilité et la motivation

2.3 L'identification des besoins

Le CNTSID ont un effectif important, ce qui nécessite et utilise beaucoup de documents administratifs papier. Le traitement quotidien des documents papier demande beaucoup d'efforts. En calculant le temps total nécessaire à l'impression, la signature et la numérisation de chaque document est important, Afin de les délivrer à leurs destinataires.

Les responsables du CNTSID ont souhaité passer à la dématérialisation, qui pourrait faire gagner du temps grâce aux signatures numériques. Cette solution technologique prend encore plus d'importance dans le monde des documents dématérialisés, permettant de faire évoluer les procédures administratives, d'approvisionnement ou de gestion.

La réalisation de la plateforme de signature numérique nécessite la mise en place d'une solution fiable pour que les identités électroniques et les identifiants de signature soient vérifiés par un tiers de confiance « une autorité de certification » ou bien toute une infrastructure de confiance.

2.4 La description du travaille à réaliser

Nous allons développer une application qui nous permet la description d'un système de signature électronique des documents administratifs internes du CNTSID.En garantissant l'identité de signataire,ainsi que l'intégrité et la confidentialité avec des mécanismes de sécurité(la cryptographie asymétrique et les fonctions de hachage).

Les objectifs de notre travail sont la mise en oeuvre d'un outil qui permet :

- La disposition d'un moyen pour assurer :

  L'intégrité : le document n'a pas été altérer entre l'envoi et la réception.

  L'authenticité : identifier la personne ou l'organisme qui a effectué l'action (certificat électronique)

  La non-répudiation : la personne qui a réalisé l'action ne peut le nier.

  La confidentialité : la cryptographie.

- La création d'une interface qui repend aux critères ergonomique, qui est simple à utiliser.
- Les Différentes taches gérées par l'utilisateur :
  - la signature d'un document(word/pdf).
  - La vérification d'un document signé.
  - la consultation de son certificat électronique.
- Les Différentes taches gérées par l'administrateur :

— L'authentification.

— La création d'un nouveau certificat électronique à un nouveau utilisateur.

— L'extraction de clé publique d'un nouvel utilisateur.

2.5 La spécification des besoins fonctionnels

Les besoins fonctionnels représentent ce que le système doit faire en réponse à une requête. La solution à réaliser doit fournir un ensemble de fonctions qui doivent être liées à un ensemble de besoins de l'utilisateur. Ceux-ci définissent les services que les utilisateurs attendent de cette solution.

La solution doit satisfaire les principaux besoins fonctionnels qui se présentent dans les points suivants :

L'authentification : l'application assure l'authentification de l'administrateur.

La gestion des certificats : L'application doit être capable d'offrir à l'administrateur la possibilité d'ajouter un nouveau certificat et l'extraction de clés publiques.

La signature des documents : L'application doit être capable d'offrir à l'utilisateur la possibilité de signer un document sélectionner.

La vérification de signature : L'application doit être capable d'offrir à l'utilisateur la possibilité de vérifier la signature.

Le precessus métier de la signature électronique des documents se résume en deux parties essetielles :

a-Le signataire : à partir d'un document à signer :

- On calcule la valeur de hachage cryptographique du document.
- On chiffre cette valeur avec sa clé privée,ce qui implique la signature.
- On transmet le document signé et l'identité de signataire.

b-Le recepteur :

- Reçoit le document signé.
- Applique le hachage cryptographique sur le document qui recoit et trouve l'empreinte H1.
- Récupérer la clé publique du signataire à partir de son identité.
- Utilise la clé publique pour déchiffrer la signature et trouve l'empreinte H2.
- Compare H1 et H2.
- La signature est valide si les deux empreintes sont égales.

2.6 La spécification des besoins non fonctionnels

Les besoins non fonctionnels sont une restriction ou une contrainte qui pèse sur un service du système, telles que les contraintes environnementales et de mise en œuvre ainsi que les exigences de performance, les dépendances du projet, la maintenabilité, l'évolutivité et la fiabilité.

L'application doit répond à tous ces critères :

- La solution doit être performante, fiable et facile à utiliser.
- Les taches doivent être indépendantes pour ne pas se bloquer à une phase spécifique.
- La solution doit être sécurisée et doit accepter les améliorations.
- La solution est conviviale.

2.7 La spécification des besoins techniques

Les besoins techniques listent toutes les contraintes et choix de conception du système, les outils et matériaux choisis, et la prise en charge des contraintes d'intégration sont souvent des pré-requis pour une architecture générique.

Les choix techniques adoptés pour le projet sont comme suit :

- La modélisation du système avec UML.
- Le langage JAVA et l'IDE Netbeans pour le développement de la solution.
- Les API :

  JCA (Java Cryptography Architecture) qui définit l'architecture générale du framework et les fonctionnalités cryptographiques de base (fonctions de hachage, signature numérique, clés, certificats...)

  JCE (Java Cryptography Extension) qui fournit des fonctionnalités cryptographiques de haut niveau (chiffrement/déchiffrement avec algorithmes symétriques/asymétriques, authentification de messages (HMAC),...).
- Le fournisseur Bouncy Castel.
- L'algorithme SHA-256 pour le hachage.
- L'algorithme RSA pour le cryptage asymétrique.
- PKCS (les standards de cryptographie à clé publique).

2.8 Conclusion

Pour mener à bien la partie de spécification des besoins dans ce projet, nous avons défini les différents besoins attendus de notre solution sous l'objectif principale du centre (sécurisé les technologies de l'information et de la communication en douane), en identifiant les fonctionnalités du système pour les deux acteurs l'administrateur et l'utilisateur . Dans le chapitre suivant, nous présenterons la conception de notre application.

Chapitre 3

La conception

3.1 Introduction

Un projet est un ensemble de tâches avec un début et une fin pour répondre à des besoins préalablement prédéfinis. Le choix du cycle de vie du logiciel et la modélisation ou la conception sont des parties importantes de la bonne réalisation du projet.

Pour la réalisation de notre projet on a choisi la méthode agile UNIFIED PROCESS (UP et RUP) par ce qu'elle présente des caractéristiques dont trois sont préconisés par UML. Ainsi qu'elle procède de manière itérative et incrémentale ce qui permet de découvrir les erreurs et les incompréhensions plus tôt. Cette méthode permet aussi de gérer les changements à chaque étape ou phase de développement.

3.2 Le choix de cycle de vie

Le « cycle de vie d'un logiciel » désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition.

3.2.1 Le diagramme de Gantt

Le diagramme de GANTT est tracé lorsque l'ordonnancement du projet est défini. Il permet de représenter graphiquement l'occupation du temps par les activités du projet. Et de ce diagramme découle le profil de consommation des ressources allouées aux activités.[19]

Le diagramme suivant va représenter les taches principales à réaliser dans notre projet.

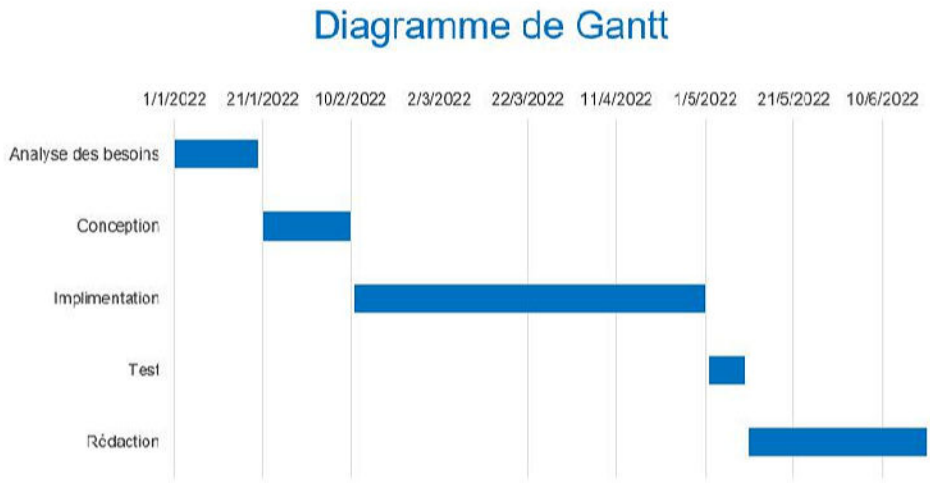

FIGURE 3.1 – Diagramme de Gantt

3.3 Unified Modelling Language "UML"

Unified Modeling Language (UML) est un langage de modélisation pour décrire des logiciels informatiques. Modéliser et décrire des logiciels à l'aide de langages de modélisation UML permet d'obtenir des modèles UML. Modèle UML Décrit uniquement les modèles conceptuels, mais pas les contraintes Entité sur le modèle. En pratique, des contraintes supplémentaires sont parfois nécessaires pour décrire le modèle UML. Ces contraintes sont généralement décrites en langage naturel, et Ne désambiguïsez pas.[21]

3.4 La conception

3.4.1 Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la fonctionnalité du système du point de vue de l'utilisateur.Le modèle de cas d'utilisation capture les exigences d'un système et les fonctionnalités futures que doit l'implementer, les cas d'utilisation sont un moyen de communication avec les utilisateurs et d'autres parties prenantes ce que le système est destiné à fairem en montrant l'interaction entre le système et les entités externes au système. [14]

3.4.1.1 L'identification des acteurs

Notre solution repose sur 2 types d'acteurs.le rôle de chacun est représenté ci-dessus :

Rôles de l'administrateur
Le cas d'utilisation démarre lorsque l'administrateur souhaite : 1. Ajouter un nouveau certificat . 2. Extraire la clé publique d'un nouvel utilisateur.

TABLE 3.1 – Les principaux Rôles de l'administrateur.

Rôles de l'utilisateur
Le cas d'utilisation démarre lorsque l'utilisateur souhaite : 1. Signer un document . 2. Vérifer la signature d'un document signé. 3. Consulter son certificat.

TABLE 3.2 – Les principaux Rôles de l'utilisateur.

3.4.1.2 Le diagramme de cas d'utilisation de l'administrateur

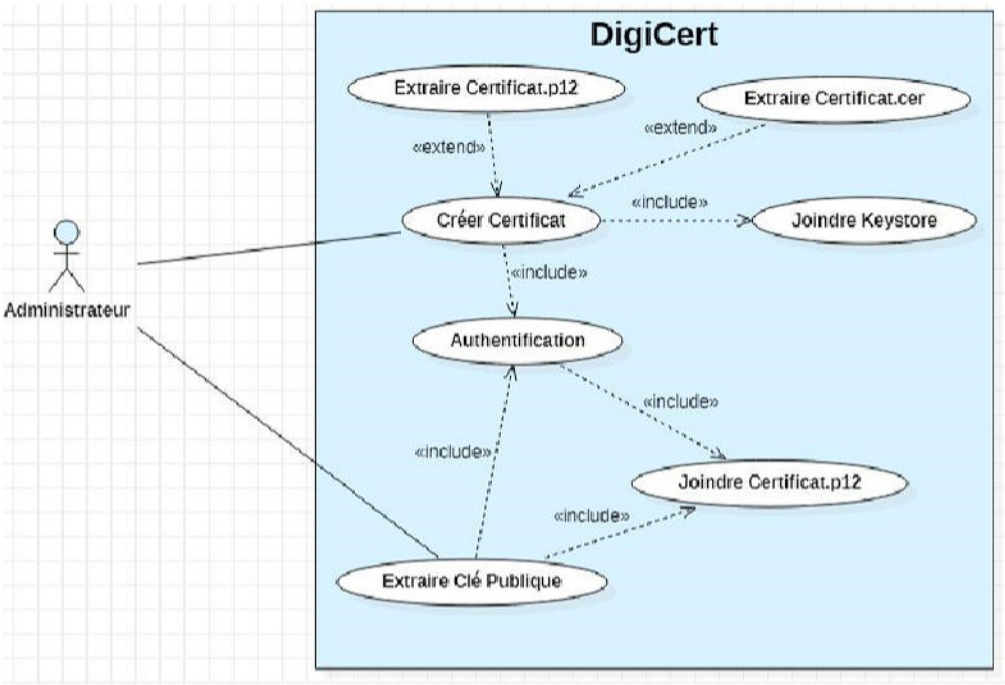

FIGURE 3.2 – Diagramme de cas d'utilisation de l'administrateur.

La description des cas d'utilisations de l'administrateur :

— **Authentification :**
Émet : l'accés a l'accueil de l'application DigiSign .
Reçoit : certificat.p12 et son mot de passe .

— **Ajouter un nouveau certificat :**
Émet :le certificat.p12 et certificat.crt.
Reçoit : information de l'utilisateur .

— **Extraire clé publique de nouveau utilisateur :**
Émet : extraction de clé publique.
Reçoit : le certificat.p12 et certificat.crt .

3.4.1.3 Le diagramme de cas d'utilisation de l'utilisateur

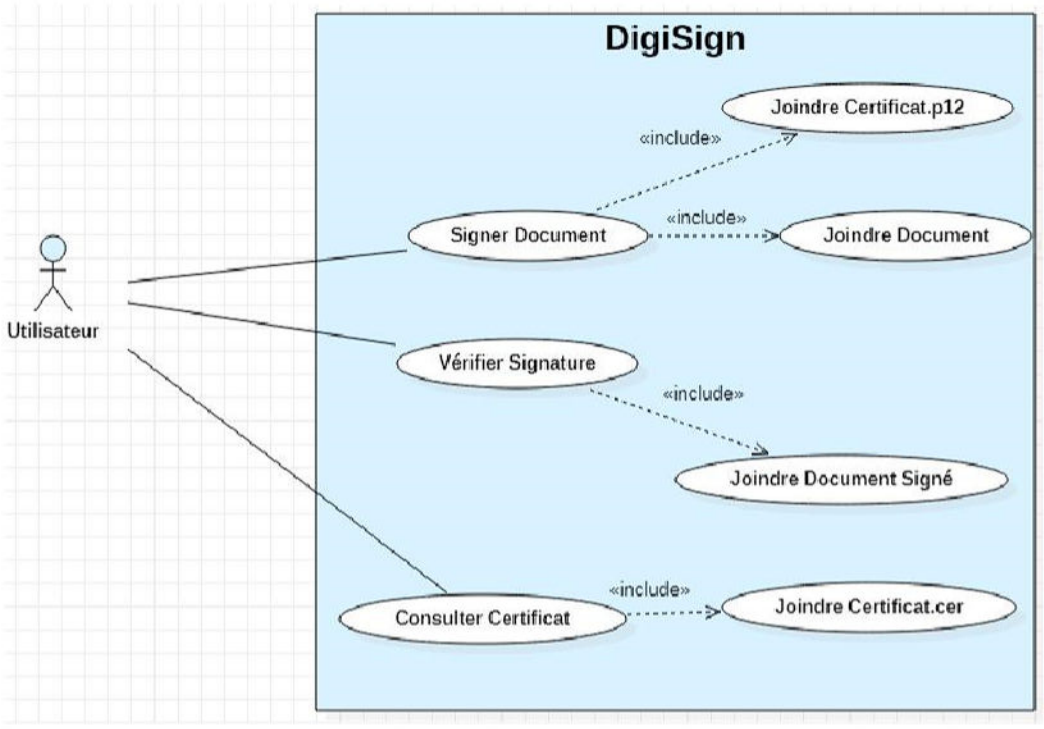

FIGURE 3.3 – Diagramme de cas d'utilisation de l'utilisateur.

La description des cas d'utilisations de l'utilisateur :

— **Signer un document :**
Émet : document signé .
Reçoit : document à signer et cetificat.p12 .

— **Vérifier la signature d'un document signé :**
Émet :verification de signature.
Reçoit :document signé et la clé publique .

— **Consulter certificat :**
Émet : consultation de certificat.
Reçoit : le certificat.crt .

3.4.2 Le diagramme de séquence

Le diagramme de séquence est un diagramme d'interaction entre les objets, qui met l'accent sur le classement des messages par ordre chronologique durant l'exécution du système. Il est utilisé pour représenter certains aspects dynamiques d'un système : dans le contexte d'une opération, d'un système, d'un sous-système, d'un cas d'utilisation (un scénario d'un cas d'utilisation) selon un point de vue temporel. En effet dans cette phase, et après identification des cas d'utilisation, nous représentons à l'aide des diagrammes de séquences les différents cas d'utilisation.[14]

3.4.2.1 Le diagramme de séquence de l'extraction de clé publique

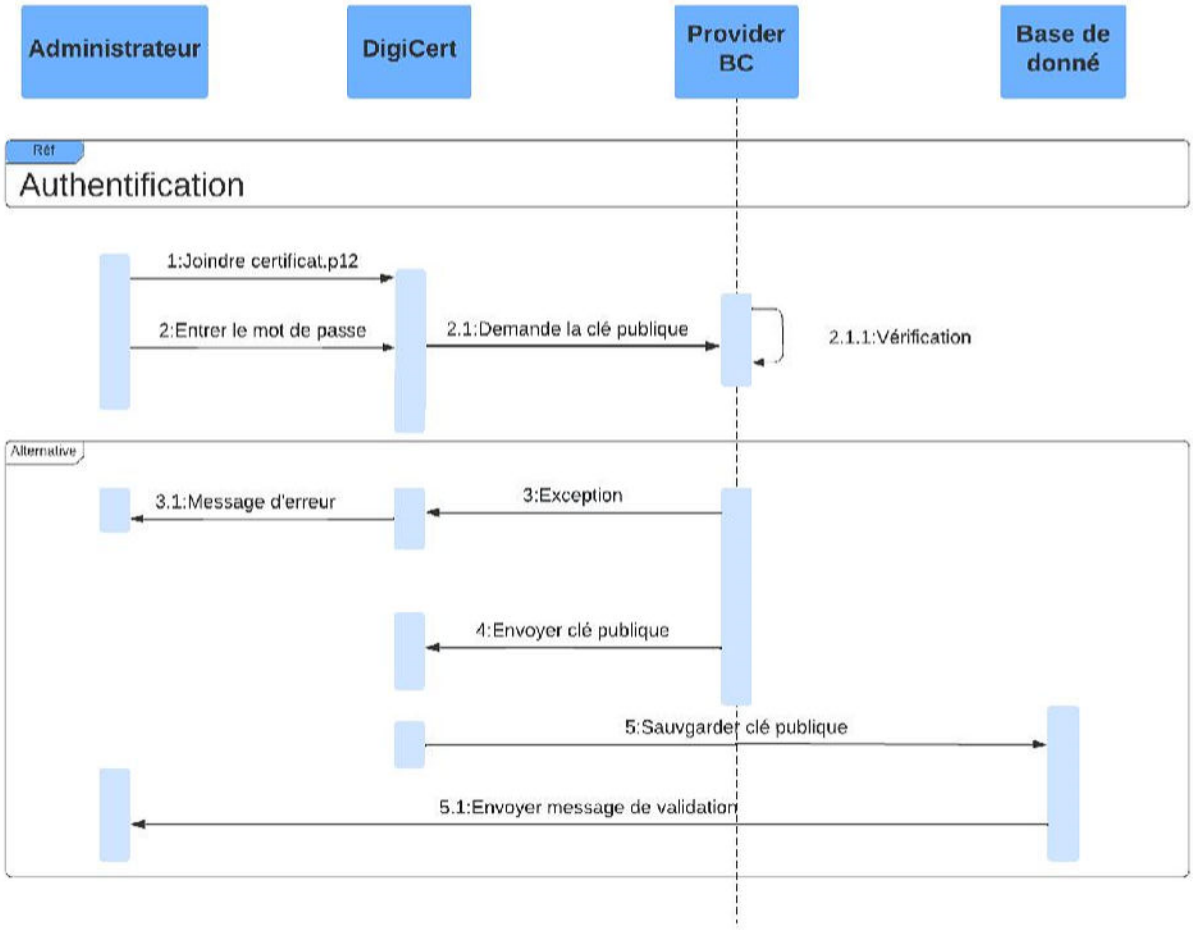

FIGURE 3.4 – Diagramme de séquence de l'extraction de clé publique.

Pré condition :

L'administrateur doit authentifier à DigiCert.

Scénario normal :

1 : L'administrateur doit joindre certificat.

2 : Envoyer le mot de passe et demande la clé publique,Le fournisseur Bouncy Castle vérifier le mot de passe.

3 :Le fournisseur Bouncy Castle envoie la clé publique .

4 : Sauvgarder la clé publique dans la base de données.

5 : Le système affiche un message de validation.

Post condition :

L'extraction de clé publique est effectué.

3.4.2.2 Le diagramme de séquence de la signature

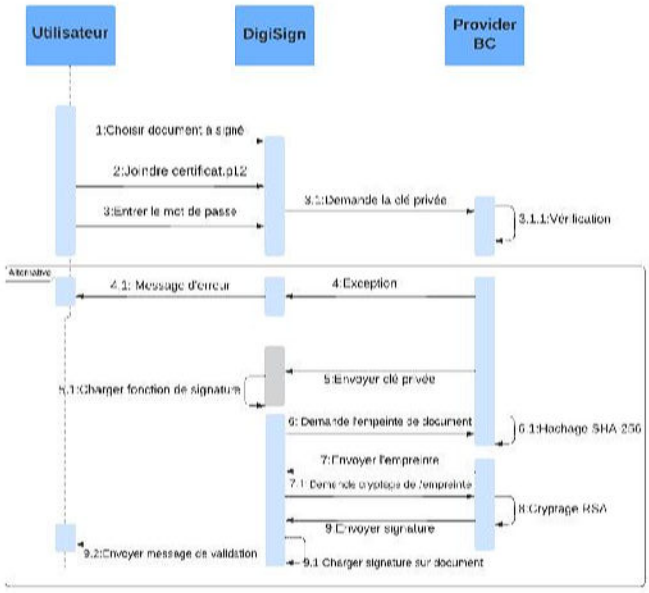

FIGURE 3.5 – Diagramme de séquence de la signature.

Pré condition :

L'utilisateur doit avoir un certificat.

Scénario normal :

1 : L'utilisateur joindre le document à signer.

2 : Joindre son certificat.

3 : Entrer son mot de passe de certificat et demander la clé privée.

4 :Le fournisseur Bouncy Castle vérifier le mot de passe.

5 : Envoyer la clé privée et charger la fonction de signature.

6 : Le système demande l'empreinte de document le fournisseur faire le hachage avec SHA-256 .

7 : Le fournisseur envoyer l'empreinte et le système demande le cryptage de l'empreinte.

8 :Le fournisseur crypter avec l'algorithme de RSA.

9 :Envoyer la signature, charger la signature sur le document et envoyer message de validation.

Post condition :

La signature de document est effectué.

3.4.2.3 Le diagramme de séquence de la vérification de la signature

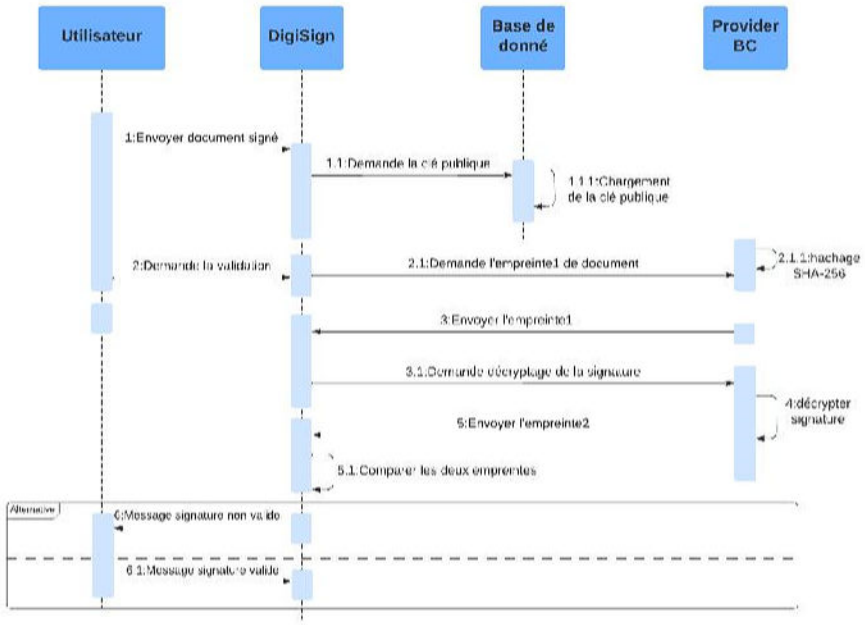

FIGURE 3.6 – Diagramme de séquence de la vérification de la signature.

Pré condition :

L'utilisateur doit avoir un document signé.

Scénario normal :

1 : L'utilisateur joindre le document signer, demande la clé publique à la base de données qui la charger.

2 : Demande la validation, demande l'empreinte 1 de document à le fournisseur Bouncy Castele qui fait le hachage avec SHA-256.

3 : Entrer l'empreinte 1 et le système demander le décryptage de la signature.

4 :Le fournisseur Bouncy Castle décrypter la signature.

5 : Envoyer l'empreinte 2 et le système comparer les deux empreinte.

6 :envoyer le message de signature valide.

Post condition :

La signature de document est effectué.

3.4.3 Le diagramme d'activité

Les diagrammes d'activité sont un cas particulier des diagrammes d'état qui doivent être utilisés dans des situations où la plupart des événements représentent l'achèvement d'actions générées en interne. Le comportement est donc dominé par le traitement interne. En revanche, les diagrammes d'états doivent être utilisés dans les situations où des événements principalement asynchrones se produisent.[8]

3.4.3.1 Le diagramme d'activité de la signature

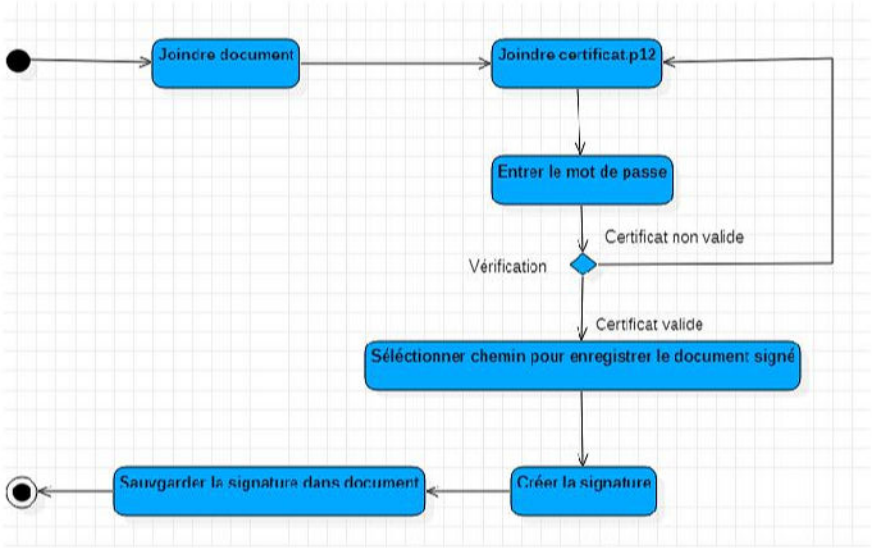

FIGURE 3.7 – Diagramme d'activité de la signature.

3.4.3.2 Le diagramme d'activité de la vérification de la signature

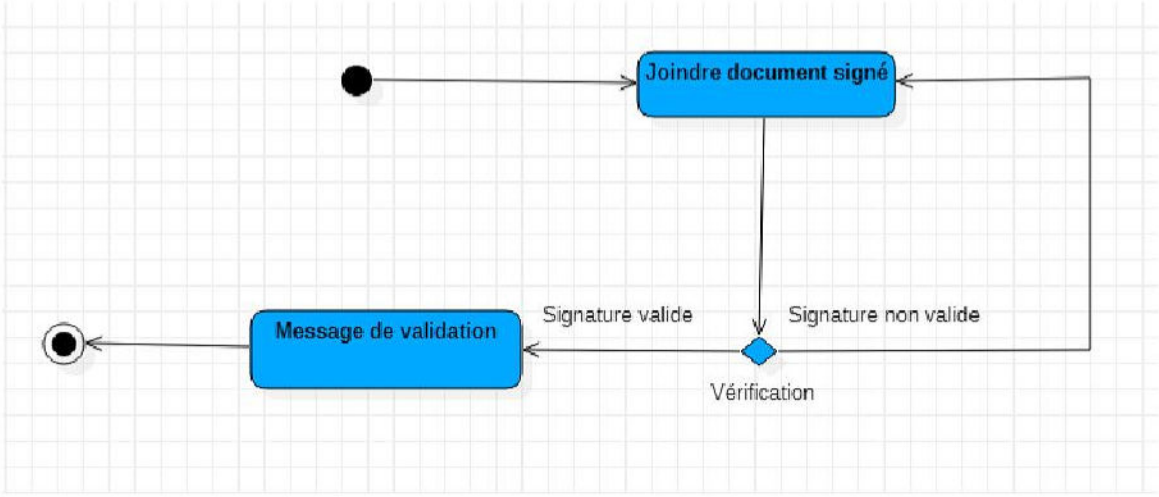

FIGURE 3.8 – Diagramme d'activité de la vérification.

3.4.3.3 Le diagramme d'activité de la création de nouveau certificat

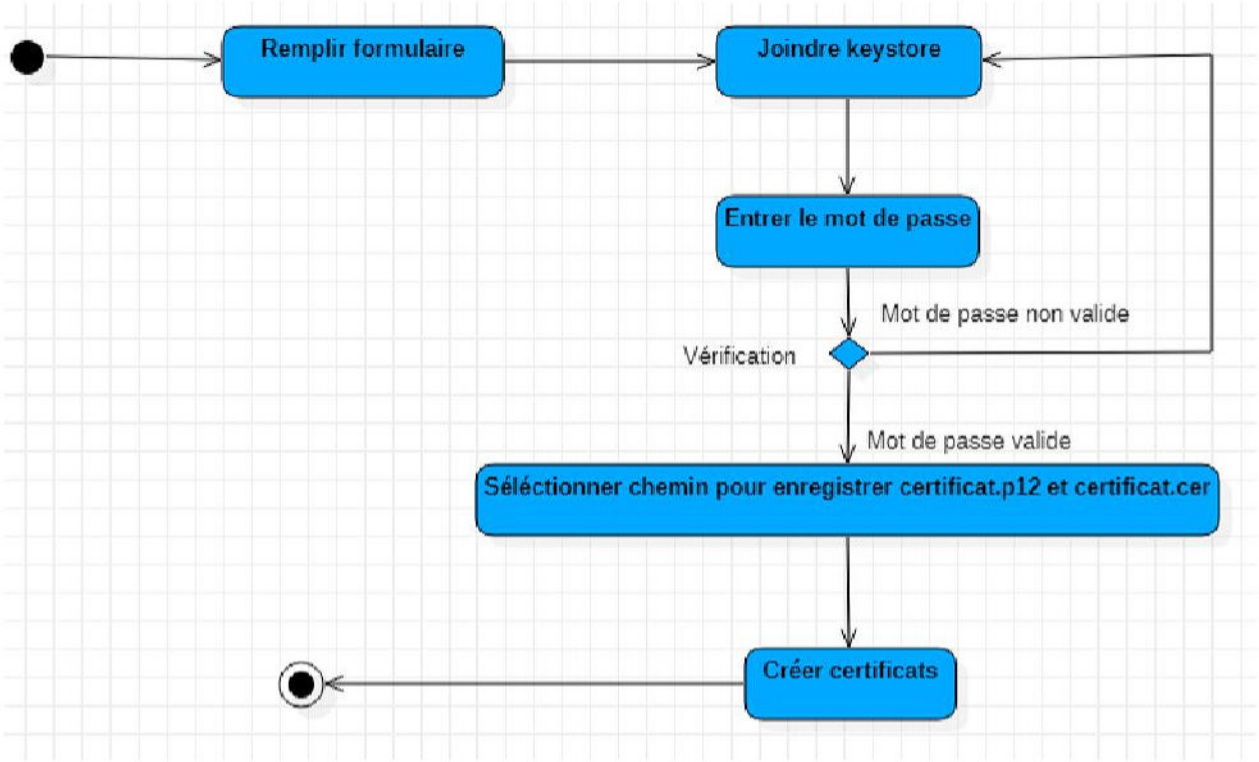

FIGURE 3.9 – Diagramme d'activité de la création de nouveau certificat.

3.4.4 Le diagramme de classe

les diagrammes de classe correspondent à une vue statique structurelle du système .ils représentent un ensemble de classes,d'interfaces, de collaborations et leurs relations.Ce sont les diagrammes les plus fréquents dans une modélisation par objets.[26]

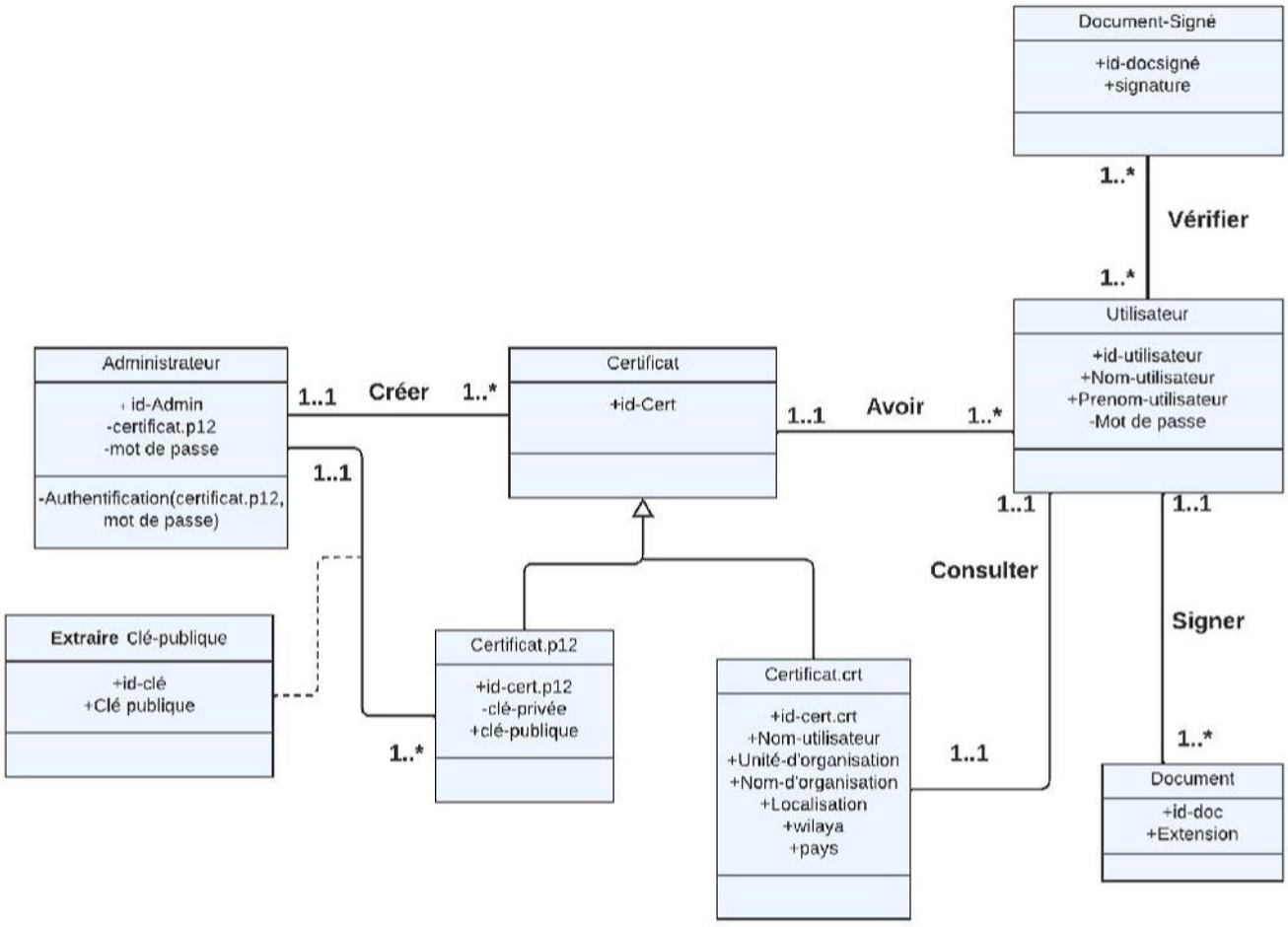

FIGURE 3.10 – Diagramme de classe.

3.4.5 Conclusion

UML permet de construire plusieurs modèles d'un système : certains montrent le système du point de vue de l'utilisateur, d'autres montrent sa structure interne, et d'autres encore fournissent une vue globale ou détaillée. Ces modèles se complètent et peuvent être assemblés tout au long du cycle de vie du développement du système.

Dans ce chapitre, nous avons modélisé DigiSign, un système de signature électronique de documents (word/pdf),afin d'aborder l'implémentation dans le chapitre suivant.

Chapitre 4

L'implémentation

4.1 Introduction

En informatique l'implémentation désigne la mise en œuvre, ou la réalisation, donc L'objectif de ce chapitre est de présenter les techniques, les langages et les outils utilisés pour la mise en œuvre de notre application pour répondre aux besoins définis précédemment. Nous présenterons les différents composants du système ainsi que quelques interfaces illustrant les différentes options offertes.

Le choix du langage s'est porté sur **Java**,qui étant orienté objet à la base et conçu pour être de type sécurisé et facile à utiliser.Il contient une architecture de sécurité destinée à protéger les ressources locales de code chargé à distance.

4.2 Les outils de développement

Cette partie a été concrétisée par la présentation des différents outils utilisés pour la réalisation de notre projet.

4.2.1 XAMPP

Cross-Platform (X), Apache (A), MySQL (M), PHP (P) et Perl(P) (XAMPP). C'est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) facile à installer offrant une bonne souplesse d'utilisation permettent l'exploitation d'un serveur Apache, de l'SGBD MySQL et l'interpréteur PHP. XAMPP est également multiplate-forme, ce qui signifie qu'il fonctionne aussi bien sur Linux, Mac et Windows. [29] Son interface est illustré à la figure suivante.

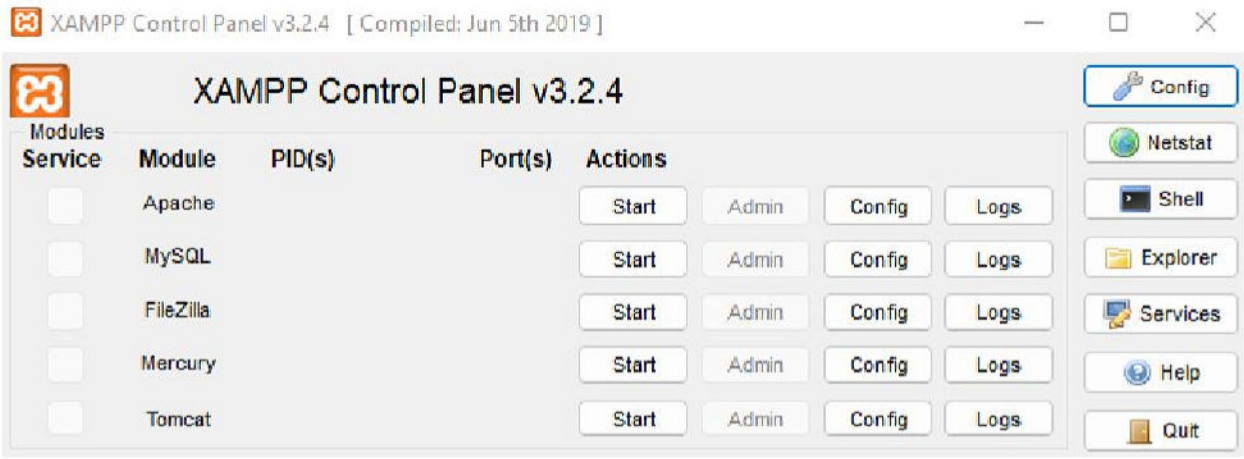

FIGURE 4.1 – XAMPP

4.2.2 L'environnement de développement "NetBeans"

NetBeans est un Integrated Development Environment (IDE), environnement libre et facile à utiliser. Il prend en charge plusieurs langages de programmation à savoir Java, C++, php et bien d'autres. Il fournit plusieurs outils tels qu'un éditeur de texte doté d'une pré-compilation avancé, gestionnaire de projet ainsi que des outils de débogage et de test de programmes, c'est un outil qui facilite grandement la phase de développement et de test.[16]

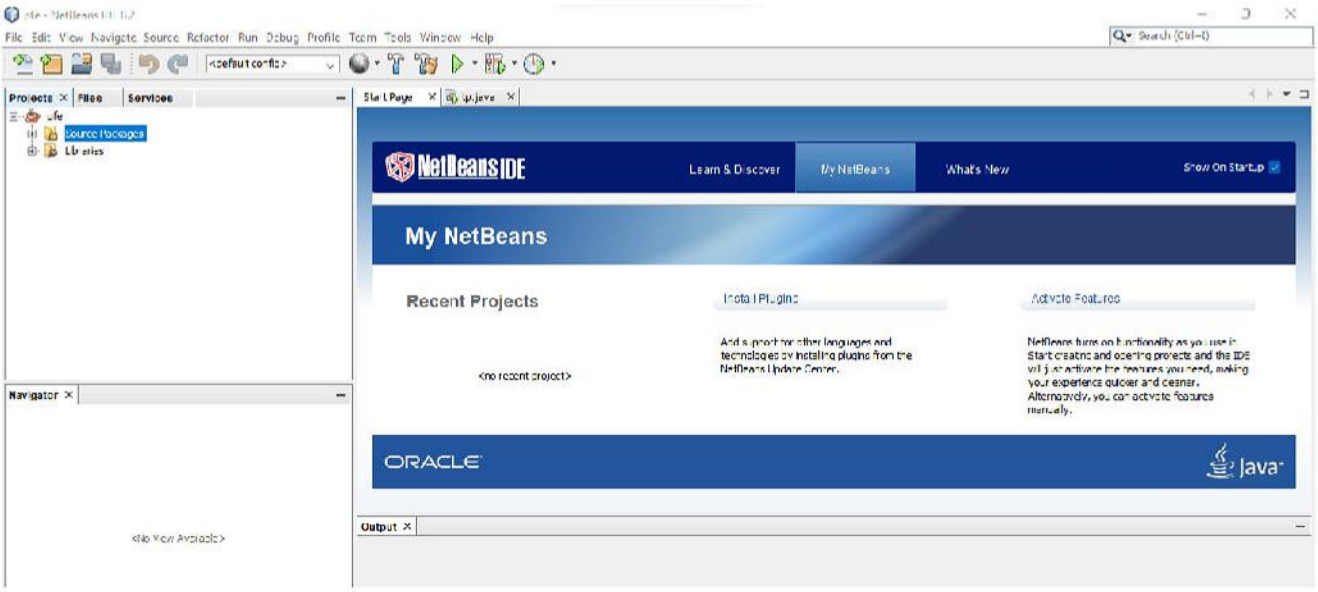

FIGURE 4.2 – L'IDE NetBeans

4.2.3 Les interfaces de programmation d'applications "API"

Interface de Programmation Applicative (API) de Cryptographie Java permet de chiffrer et de déchiffrer des données en Java, de gérer des clés, de signer et d'authentifier des messages, de calculer des hachages cryptographiques. La technologie Java possède un certain nombre d'API qui fournissent des classes utiles pour écrire des applications sécurisées. Ces API offrent un riche ensemble de fonctionnalités, avec la prise en charge d'un large éventail de tâches :[9]

Cryptography Architecture Java (JCA) permet de fournir des capacités cryptographiques aux programmes Java. JCA comprend la prise en charge des résumés de messages, des signatures numériques, de la gestion des paires de clés, de l'authentification et des certificats. La conception de ce structure suit le modèle Model-View-Controller (MVC), qui sépare les concepts de la mise en œuvre. [9]

Java Cryptography Extension (JCE) est une extension de JCA. JCE fournit un structure et des implémentations pour le cryptage, la génération et l'accord de clés, et les algorithmes de code d'authentification de message (MAC). La prise en charge du chiffrement comprend les chiffrages symétriques, asymétriques, par blocs et par flux. [9]

4.2.4 Le magasin de clés "KeyStore"

Java KeyStore est une base de données pouvant contenir des clés. Le Java KeyStore est représenté par la classe KeyStore (java.security.KeyStore) . La classe KeyStore permet de stocker, gérer et récupérer les éléments contenus dans un dépôt de clés.La classe KeyStore permet d'accéder et de modifier les deux types d'éléments que peut contenir un dépôt : des clés et des certificats.[27]

4.2.5 L'outil de gestion des clés (Keytool)

Java Keytool c'est un outil de ligne de commande qui peut fonctionner avec les fichiers Java KeyStore. Keytool peut générer des paires de clés vers le fichier KeyStore, exporter des certificats et importer des certificats vers KeyStore et certaines autres fonctions. Keytool est livré avec une installation Java. [27]

4.2.6 Le fournisseur Bouncy Castele

Dans le langage de programmation Java, JCA fournit l'architecture de base pour l'utilisation d'algorithmes cryptographiques, elle spécifie les interfaces avec lesquelles les algorithmes cryptographiques peuvent être utilisés dans un programme Java. Les implémentations réelles sont ensuite fournies par ce que l'on appelle des fournisseurs de cryptographie. La légion du paquet Bouncy Castle pour Java implémente la plupart des interfaces définies dans la JCA en particulier, elle implémente la classe Java Cryptography Provider et s'intègre donc facilement à la JCA. Le fournisseur Bouncy Castle offre plusieurs routines de cryptage et de décryptage.[28]

4.3 Description de l'application développée

Le système développé est composé de deux espaces : espace administrateur et utilisateur. Dans ce qui suit nous allons donner un scénario d'utilisation des différentes fonctionnalités de notre système.

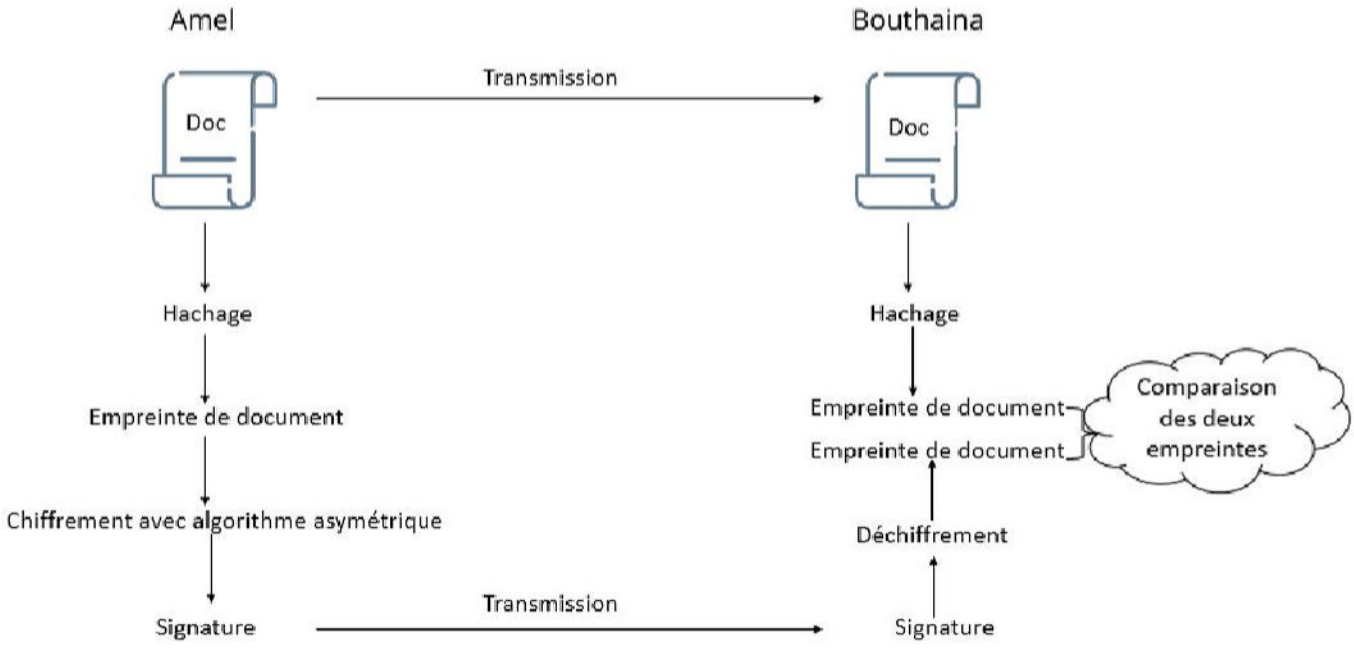

FIGURE 4.3 – Signature électronique.

4.3.1 L'espace administrateur "DigiCert"

L'administrateur c'est le responsable de toutes les actions concernant les acteurs du système . DigiCert représente l'espace administrateur de notre application afin d'offrir une utilisation simple et conviviale .

4.3.1.1 L'authentification

Cette interface permet à l'administrateur de s'authentifier à l'application en utilisant son certificat numérique.

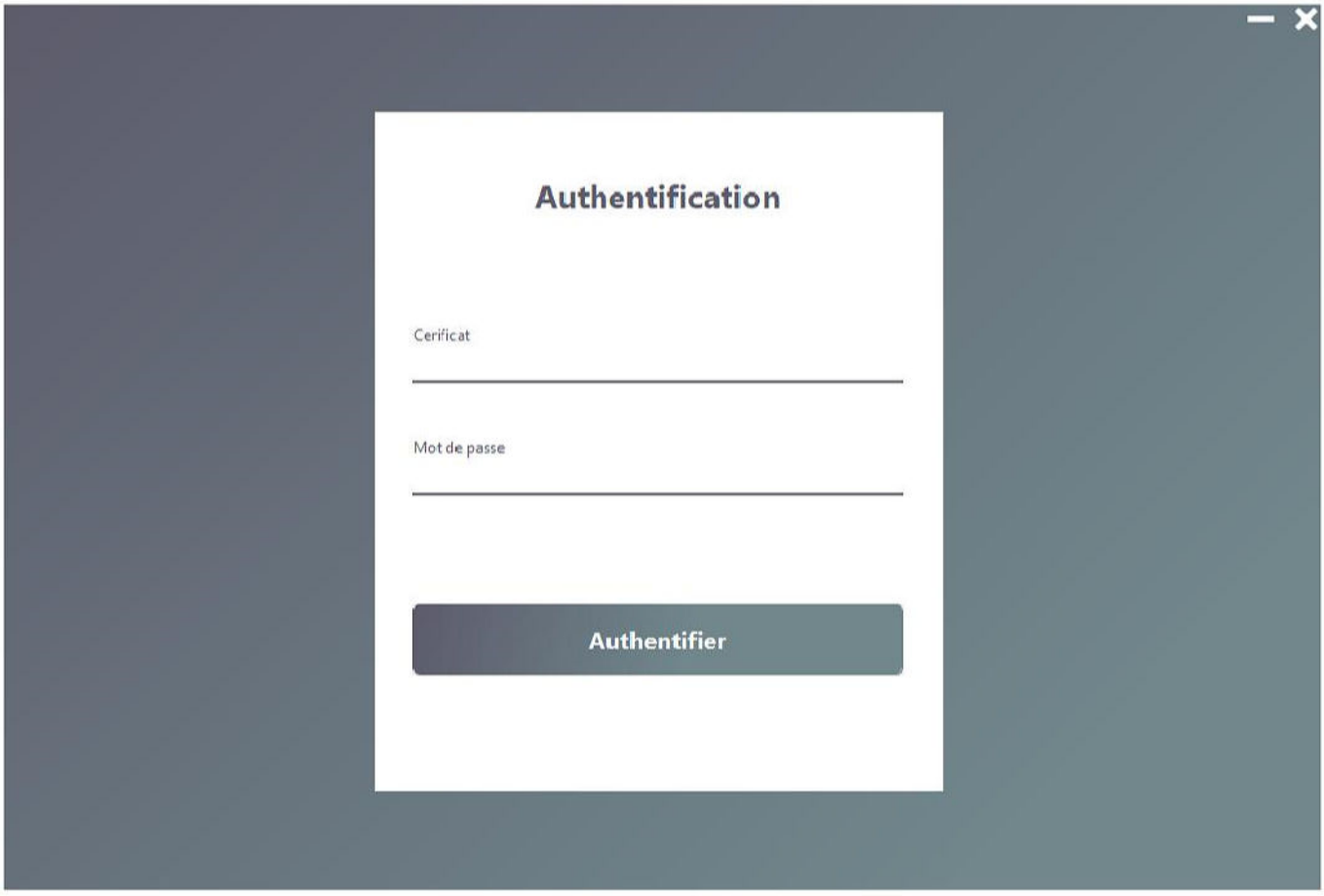

FIGURE 4.4 – L'authentification

4.3.1.2 La création de nouveau certificat

Chaque nouvel utilisateur doit avoir un certificat numérique pour signer ses documents. La figure ci dessous représente le formulaire de création de nouveau certificat.

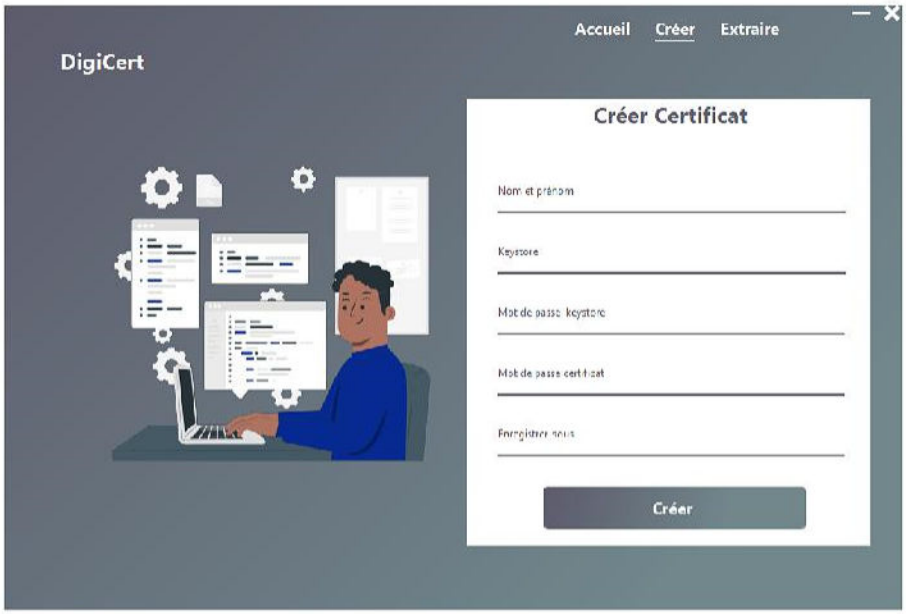

FIGURE 4.5 – La création de nouveau certificat

4.3.1.3 L'extraction de clé publique

Dans cette interface, l'administrateur peut facilement extraire la clé publique afin de l'utiliser pour la vérification de l'authenticité de signataire.Les clés publiques sont extrait dans la base de données sous forme de fichiers associer avec les noms des utilisateurs.

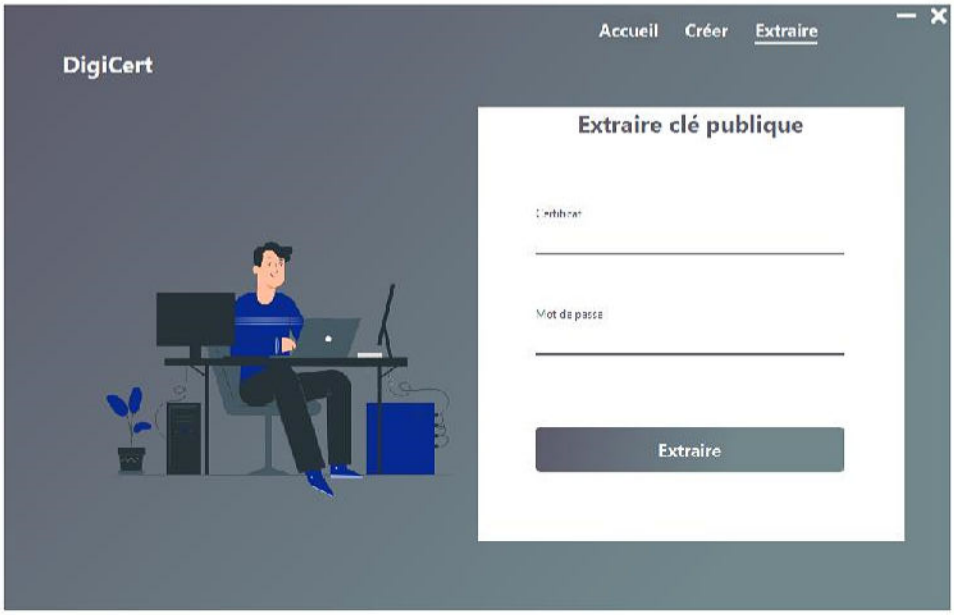

FIGURE 4.6 – L'extraction de clé publique

4.3.2 L'espas utilisateursce utilisateur "DigiSign"

A travers DigiSign les acteurs peuvent utiliser le système chacun selon ses droits. Nous donnons une description pour chaque fenêtre ce qui concerne les différentes interfaces que constituent notre application.

4.3.2.1 La signature d'un document

La figure suivante montre le formulaire rempli par l'utilisateur(signataire) pour signer un document(Word/PDF).

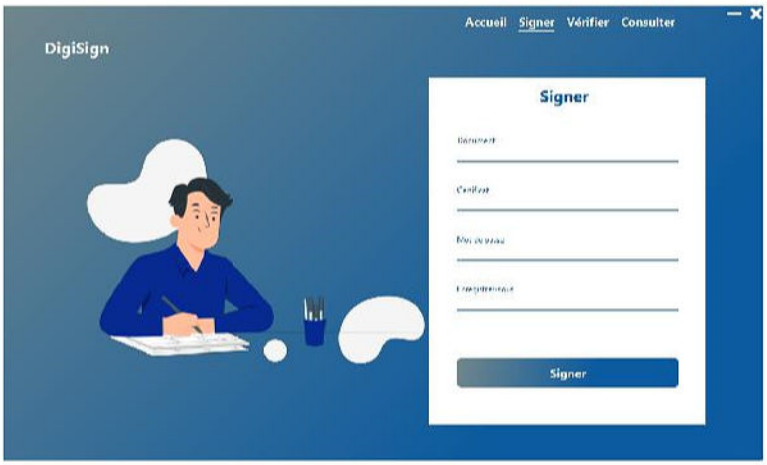

FIGURE 4.7 – La signature d'un document

On a utilisé l'algorithme SHA- pour calculer l'empreinte de document.

```
MessageDigest md = MessageDigest.getInstance("SHA-256", "BC");
byte[] empreinte = Hex.encode(md.digest(content2.getBytes()));
```

FIGURE 4.8 – L'algorithme de hachage

Concernant le cryptage on a utilisé l'algorithme RSA.

```
Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
```

FIGURE 4.9 – L'algorithme de cryptage

a- Le document PDF signé :

La figure suivante représante un essai sur un document PDF.

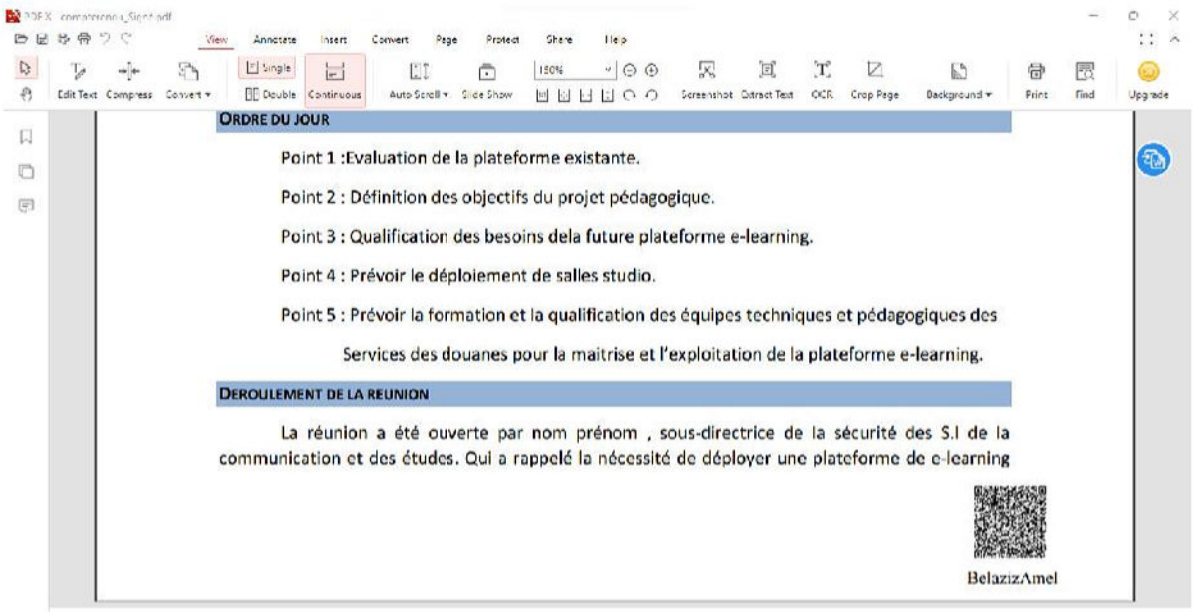

FIGURE 4.10 – Document pdf signé

FIGURE 4.11 – QR-code de la signature

On a transformer la signature à un qr-code ,en utilisant la fonction suivante.

```
public static void generateQRcode(String data, String path,
      String charset, Map map, int h, int w)
      throws WriterException, IOException {

    BitMatrix matrix = new MultiFormatWriter().encode
    (new String(data.getBytes(charset), charset),
          BarcodeFormat.QR_CODE, w, h);
     MatrixToImageWriter.writeToFile(matrix,
     path.substring(path.lastIndexOf('.') + 1),new File(path));

          }
```

FIGURE 4.12 – Génération de QR-code

b- Le document Word signé :

La figure suivante représante un essai sur un document Word.

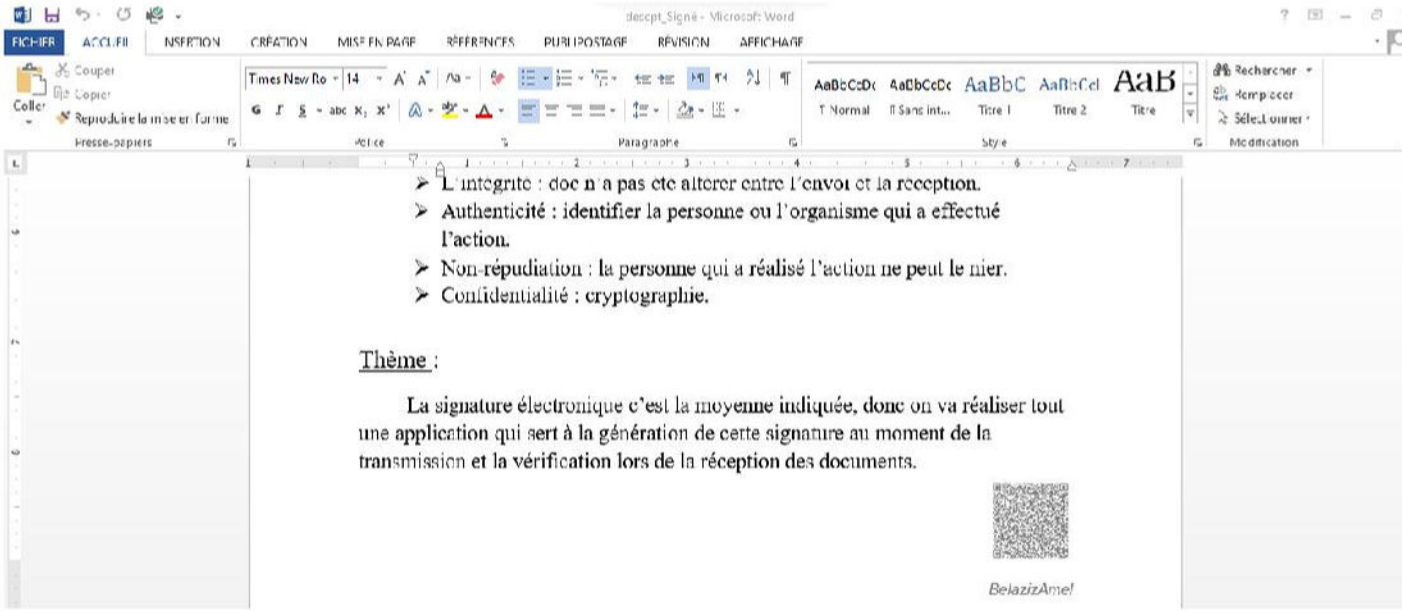

FIGURE 4.13 – Document Word signé

4.3.2.2 La vérification de la signature

La figure suivante montre comment un utilisateur vérifie un document signé.

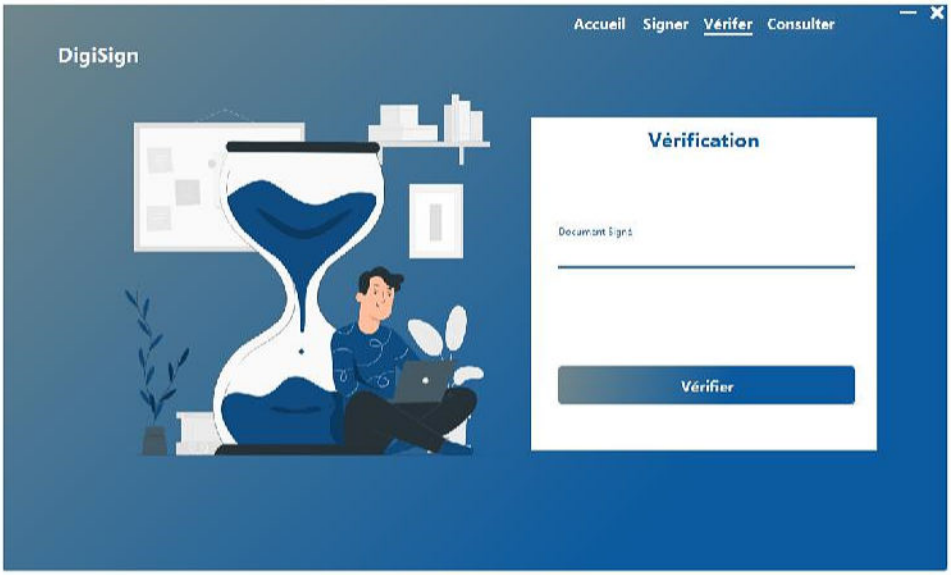

FIGURE 4.14 – La vérification de la signature

La vérification se fait par le hachage de document et le décryptage de cette empreinte afin de comparer les deux résultats.

```
Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
cipher.init(Cipher.DECRYPT_MODE,keyFactory.generatePublic(publicKeySpec));
byte[] decryptedMessage = cipher.doFinal(encryptedBytes);
```

FIGURE 4.15 – L'instance de l'algorithme utilisé

4.3.2.3 La consultation de certificat

L'utilisateur a le droit de consulter son certificat(le nom de l'autorité, les dates de validité du certificat..). La figure suivant représente l'interface de consultation de certificat.

FIGURE 4.16 – La consultation de certificat

4.4 Conclusion

La phase d'implémentation a été concrétisée par la réalisation de l'application, en respectant la modélisation qu'on a présentée dans le chapitre précédent pour la conception de notre projet .

Dans la première partie de ce dernier chapitre nous avons présenté les différents outils et technologies qui ont été utilisés pour réaliser cette solution.Dans la deuxième partie on a essayé de faire une explication de notre plateforme et donner une présentation générale sur les deux espaces (administrateur/utilisateur).

Conclusion et perspectives

Une signature numérique est un mécanisme permettant de garantir l'intégrité des documents électroniques et d'authentifier l'auteur, similaire aux signatures manuscrites pour les documents papier. La signature numérique permettra aux établissements qui l'adoptérent de réduire les frais et de gagner en temps et en flexibilité. La mise en œuvre de ce projet de master vise à répondre aux besoins de développement des systèmes de signature électronique au niveau du la douane.

Afin de fixer les idées sur le domaine de la signature numérique et sa terminologie, ce mémoire commence par une étude théorique approfondie, et l'analyse et la conception du système résultant sont détaillées dans des chapitres séparés, ces détails qui nous permettent d'implémenter solidement un système de signature. le système réalisé a été expliqué dans la dernière partie de ce mémoire.

Dans ce projet, nous implémentons un système de signature numérique pour des documents dans différents formats (word/pdf) basé sur Public Key Infrastructure (PKI). Ce dernier est basé sur des forces cryptographiques considérables, nous avons créé cet outil en langage java, nous avons des fournisseurs de service cryptographique, et on s'est appuyé sur les standards PKCS.

Notre travail est considéré comme la première version du projet "signature électronique" du Centre National des Transmissions et du Système d'Information des Douanes et constitue en même temps un maillon très important du système de numérisation du centre. Dans le même temps,le système de signature numérique réalisé est générique, puisqu'il peut être utilisé par toute entreprise qui le souhaite et qui a les moyens de l'appliquer.

Le travail que nous avons réalisé peut être amélioré et enrichi. Nous espérons apporter plus de fonctionnalités pour le rendre performant et plus fonctionnel. C'est pourquoi nous envisageons d'améliorer dans cette perspective l'application à même de pouvoir crypter et signer tous types de donnés (images, vidéos... etc),convertir l'application à une application cloud qui peut vérifier la signature des document reçu dans la boite mail, autre perspective l'implémentation de cette solution à une application mobile.

Bibliographie

[1] Amounas, S. and Elmoukhtar, A. (2017). Cryptographie quantique.

[2] Archimbaud, J.-L. (2003). Les principes techniques des certificats électroniques. *Les Cahiers du numérique*, 4(3) :101–110.

[3] Asymétrique, C. and Principe, A. R. (2021). Unité d'enseignement fondamentale : Uef1.

[4] Baigneres, T. (2002). Attaquea texte chiffré choisi contre les protocoles basés sur pkcs.

[5] Bekkouche, T. (2018). *Développement et implémentation des techniques de cryptage des données basées sur les transformées discrètes.* PhD thesis.

[6] Bouallagui, S. (2010). *Techniques d'optimisation déterministe et stochastique pour la résolution de problèmes difficiles en cryptologie.* PhD thesis, INSA de Rouen.

[7] Boura, C. (2012). *Analyse de fonctions de hachage cryptographiques.* PhD thesis, Université Pierre et Marie Curie-Paris VI.

[8] Breu, R., Hinkel, U., Hofmann, C., Klein, C., Paech, B., Rumpe, B., and Thurner, V. (1997). Towards a formalization of the unified modeling language. In *European Conference on Object-Oriented Programming*, pages 344–366. Springer.

[9] Cholakov, N. and Milev, D. (2005). The evolution of the java security model. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'2005).* Citeseer.

[10] Dumont, R. (2009). Cryptographie et sécurité informatique. *Eyrolles*, 2010.

[11] Fuhr, T. (2011). *Conception, preuves et analyse de fonctions de hachage cryptographiques.* PhD thesis, Télécom ParisTech.

[12] Gelin, A. (2017). *Calcul de groupes de classes d'un corps de nombres et applications à la cryptologie.* PhD thesis, Paris 6.

[13] Giraud, M., Lafourcade, P., and Ordinateur, G. I. S. (2019). Mission cryptographie. *Ressi*, 19 :15.

[14] Guérin, E. and Moussouni, F. (2001). Transcriptome hépatique : modélisation conceptuelle par une approche uml. *Les Cahiers du numerique*, 2(2) :177–196.

[15] Karpman, P. (2016). Analyse de primitives symétriques. *Soutenance le*, 18(11).

[16] Kouloughli, I., Castagna, P., and Sari, Z. (2016). Développement d'un système multi-agents (sma) pour l'optimisation du temps de déstockage dans un système automatisé de stockage/déstockage. *Electrotehnica, Electronica, Automatica*, 64(3).

[17] Kumar, D., Kashyap, D., Mishra, K., and Misra, A. (2010). Security vs cost : An issue of multi-objective optimization for choosing pgp algorithms. In *2010 International conference on computer and communication technology (ICCCT)*, pages 532–535. IEEE.

[18] Lefebvre, C. (2021). *Application du chiffrement homomorphe à la protection de la vie privée*. PhD thesis.

[19] Milon, O. (1999). *Gestion de projet avec contraintes de ressources*. École Polytechnique de Montréal.

[20] Nitaj, A. (2013). La cryptographie et la confiance numérique.

[21] Nleng, C. H. (2014). *Modélisation et vérification du flux d'information pour les systèmes orientés objets*. PhD thesis, École Polytechnique de Montréal.

[22] Pillou, J.-F. and Bay, J.-P. (2016). *Tout sur la sécurité informatique-4e édition*. Dunod.

[23] Pub, F. (1999). Data encryption standard (des). *FIPS PUB*, pages 46–3.

[24] Rubinstein-Salzedo, S. (2018). *Cryptography*. Springer.

[25] Tianfu, W. and Babu, K. R. (2012). Design of a hybrid cryptographic algorithm. *International Journal of Computer Science & Communication Networks*, 2(2) :277–283.

[26] Truong, N. T. (2006). *Utilisation de B pour la vérification de spécifications UML et le développement formel orienté objet*. PhD thesis, Université Nancy II.

[27] Vrabie, T. and Bonta, A. (2021). Cryptographie java.

[28] Waage, T. and Wiese, L. (2014). Benchmarking encrypted data storage in hbase and cassandra with ycsb. In *International Symposium on Foundations and Practice of Security*, pages 311–325. Springer.

[29] Walia, E. S. and Gill, E. S. K. (2014). A framework for web based student record management system using php. *International Journal of Computer Science and Mobile Computing*, 3(8) :24–33.

[30] Wang, Y. (2012). Public key cryptography standards : Pkcs. *arXiv preprint arXiv :1207.5446.*

[31] Zoubeyr, F. (2019). Securite informatique.