الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de de Fin d'études Master

Filière : Informatique

Option : Ingénierie des Medias

Thème :

Mise en œuvre d'un système de contrôle d'accès aux données XML

Encadré Par :

Mr. BERREHOUMA NABIL

Présenté par :

BOUGHERARA Alaeddine

GHARDAOUI Akram Abd Elmouaz

Juin 2017

Résumé

Avec l'expansion des technologies de l'information et de la communication et la démocratisation des diapositives électroniques permettant un accès rapide et direct à l'information via internet. La sécurisation de l'information est devenue un enjeu majeur. XML est aujourd'hui le standard de-facto pour décrire, échanger et disséminer tout type d'informations entre différents acteurs et pour des objectifs très variés.

Par conséquence, garantir l'intégrité, la confidentialité et la propriété intellectuelle des données XML sont devenu une question de grande importance. Plusieurs travaux sont intéressés aux différentes facettes de ce problème dans la littérature. Notre objectif dans ce projet et de faire un état de l'art sur le contrôle d'accès des données XML et étudier leurs forces d'expressivité. Et de proposer un modèle qui essaie de recueillir les avantages des modèles étudiés et éviter leurs lacunes.

Mots clés— XML , sécurisation de l'information , intégrité , confidentialité , contrôle d'accès.

Remerciements

En préambule à ce mémoire, j'adresse ces quelques mots pour remercier notre grand **Dieu** tout puissant pour exprimer ma reconnaissance envers sa grande générosité. Dieu m'a donné la volonté, la patience, la santé et la confiance durant toutes mes années d'études.

Je remercie **mes parents** d'être si patients, si généreux et tellement merveilleux, ils ont toujours été une source de motivation d'encouragements et de beaucoup de bonheur.

Je tiens 'a remercier sincèrement Monsieur **Nabil BERREHOUMA**, qui, en tant que mon encadreur, s'est toujours montré a l'écoute tout au long de la réalisation de ce mémoire, ainsi que pour son aide et le temps qu'il a bien voulu me consacrer.

jc souhaitc aussi adresser mes remerciements les plus sincères aux personnes qui m'ont apporte leur aide et qui ont contribué à l'élaboration de ce mémoire.

En effet, je voudrai remercier mon université, ma famille, mes enseignants et tous ceux qui ont participé de pr'es ou de loin à la réalisation de mon mémoire.

Je remercie également mes camarades et mes amis du département pour leurs conseils et leurs idées.

Enfin, j'adresse mes plus sincères remerciements 'a tous mes proches et amis, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Sommaire

Liste des Figures

Liste des Tables

Liste des abbreviations

JO	Journal Officiel
CNIL	Commission Nationale de l'Informatique et des Libertés
TIC	Technologies d'Information et Communication
ITSEC	Information Technology Security Evaluation Criteria
Ks	Clé Secrète
Kp	Clé Publique
RSA	Rivest Shamir Adleman
ECC	Elliptic Curve Cryptography
PKI	Public Key Infrastructure
VPN	Virtual Private Network
IPsec	Internet Protocol Security
DAC	Discretionary Access Control
MAC	Mandatory Access Control
RBAC	Role-based Access Control
ACL	Acess Control List
CL	Capability List
OTAN	Organisation du Traité de l'Atlantique Nord
XML	eXtensible Markup Language

DTD	Document Type Definition
XSD	XML Schema Definition
XPath	XML Path Language
XQuery	XML Query
W3C	World Wide Web Consortium
SAX	Simple API for XML
DOM	Document Object Model
SWS-RBAC	Role Based Access Control for Single Web services
CWS-RBAC	Role Based Access Control for Composite Web services
XACML	eXtensible Access Control Markup Language

Introduction

Avec l'expansion des technologies de l'information et de la communication et la démocratisation des diapositives électroniques permettant un accès rapide et direct à l'information via internet. La sécurisation de l'information est devenue un enjeu majeur. XML est aujourd'hui le standard de-facto pour décrire, échanger et disséminer tout type d'informations entre différents acteurs et pour des objectifs très variés. Par conséquence, garantir l'intégrité, la confidentialité et la propriété intellectuelle des données XML sont devenu une question de grande importance. Plusieurs travaux sont intéressés aux différentes facettes de ce problème dans la littérature. Le contrôle d'accès aux données XML est l'une de ces facettes qui doit être traitée avec précaution vue son impact direct sur l'ensemble des critères sécuritaire de n'importe quel système. En effet, le contrôle d'accès d'une manière générale c'est la définition qui accède à quelles ressources et dans quelle manière. Le problème qui se pose actuellement c'est la nature des données XML qui est organisé d'une manière hiérarchique. Contrairement aux systèmes classiques (systèmes d'exploitation, SGBD, réseaux informatique) les ressources peuvent être imbriqué, chevauché ou même éparpillé ce qui rend la définition des règles de contrôle d'accès un défi majeur.

Notre objectif dans ce mémoire est scindé en deux parties. La première consiste à faire un état de l'art sur les travaux réalisé dans le cadre du contrôle d'accès aux données XML afin de bien maitriser la problématique et mettre en évidence les points faibles et point forts des principaux contributions. Et dans la deuxième partie de proposer notre propre modèle de contrôle d'accès qui s'inspire des travaux existant et se veut être extensible afin d'intégrer à

chaque fis des nouvelles capacités. Pour atteindre les objectifs fixés, nous avons organisé notre mémoire en cinq chapitres.

Dans le premier chapitre, nous avons essayé de motiver notre travail par l'étude de l'aspect juridique de la sécurité informatique. Bien que cet aspect ne rentre pas dans le cœur de notre travail, mais nous jaugeons qu'il est indispensable d'avoir une idée sur l'évolution des règlementations qui régissent le domaine de sécurité informatique et plus particulièrement le domaine de contrôle d'accès afin de fournir des solutions adaptée et qui peuvent être utilisée plus tard par les différents établissements de notre pays. Un intérêt particulier est consacré dans le même chapitre aux aspects techniques de la sécurité informatique. La compréhension et la maitrise de ces aspects est absolument nécessaires pour la suite du mémoire.

Dans le deuxième chapitre, nous commençons à prendre en main l'étude du contrôle d'accès dans son globalité. Des définitions et des classifications des différents modèles sont exposé d'une manière approfondie, l'objectif est de faire une comparaison pour dégager les principaux éléments à retenir lors de la mise en œuvre d'un système de ce genre.

Durant le troisième chapitre, nous faisons une anatomie sur la nature des ressources XML. Nous présentons tout d'abord l'intérêt d'adopter le langage XML comme un support de présentation des données, puis la syntaxe et la validation des données XML sont expliquées profondément. XML est un standard et des centaines d'outils sont fournis pour l'exploration, l'interrogation, la transformation, l'encryptage, etc. sont disponibles. Nous essayons alors de présenter quelques outils qui vont nous servir lors de la réalisation de notre modèle.

Le quatrième chapitre est consacré à l'étude de principales contributions dans la littérature pour répondre à la problématique du contrôle d'accès aux données XML. Nous avons commencé par la définition des critères qui doivent être pressentes et on a terminé par l'établissement d'un tableau comparatif des différentes solutions. Nous arrivons enfin au cinquième chapitre qui est le plus important dans ce mémoire où nous présentons notre contribution s'agissant de la conception d'un modèle de contrôle d'accès basé sur le modèle RBAC. Nous expliquons la philosophie de notre modèle à travers

implémentation d'une application client/serveur qui emploi notre proposition pour contraigner l'accès aux ressources XML. Bien que notre solution est dans ses débuts, mais nous montrons sa force d'expressivité avec l'étude d'un cas et l'expérimentation de plusieurs tests.

Chapitre 1

Aspect Juridiques et Techniques de la sécurité informatique

1.1 Introduction

Durant ce chapitre, nous allons survoler l'ensemble des réglementations algériennes relatives à la protection des données personnelles, en adoptant une démarche qui suit la progression historique relative l'usage de l'outil informatique comme support pour la création , le maintien et l'échange des données personnelles. Nous allons mettre en évidence les relations entre les différents textes, ainsi que l'évolution logique constante dictée par l'évolution technologiques et l'introduction des nouvelles approches pour la sécurisation des données dans les domaines des TIC. Nous présentons aussi une introduction aux techniques de base de la sécurité informatique qui va nous servir tout au long ce mémoire.

1.2 Aspect Juridique

Notre pays n'est pas dans une ile isolée du reste du monde. La règlementation algérienne est naturellement liée aux autres réglementations apparus dans le reste du monde, et particulièrement dans les pays européen à leurs tète la France.Il est tout à fait rationnel alors de commencer par la présentation des

axes phares régissant la protection des données personnelles en France.

La loi de 06 janvier 1978, fût le premier texte en France dans cette direction. Ce texte définit le contexte relatif à l'informatique, aux fichiers et aux libertés.Ce texte a subi une modification majeure avec la loi du 6 août 2004 relative à la protection des personnes physiques à l'égard des traitements de données à caractère personnel.Dans ce qui suit, nous présenterons une description détaillée de réglementation en France puis en Algérie.

1.2.1 Loi relative à l'informatique, aux fichiers et aux libertés

Étant donné l'importance pour la protection les données électronique et les données personnelles en particulier. L'état de France était l'un des premiers pays qui ont cherché à protéger les données personnelles, pour cette raison, il a publié plusieurs lois dans cette direction. La loi du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés était la première loi, avec ses 72 articles la vie personnelle est devenue protégée contre tous les types de la cybercriminalité. Les principes de ce loi sont résumés dans le premier article qui dit que :

" L'informatique doit être au service de chaque citoyen, son développement doit s'opérer dans le cadre de la coopération internationale, elle ne doit porter atteinte ni à l'identité humaine, ni aux droits de l'homme, ni à la vie privée, ni aux libertés individuelles ou publiques. Pour garantir le respect des règles qu'elle édicte, la loi crée une institution de contrôle : la Commission Nationale de l'Informatique et des Libertés. Pour assurer la transparence des fichiers informatisés, la loi instaure un système de formalités préalables à la mise en œuvre des traitements automatisés" [1] tous les articles de ce loi nous concerne dans notre recherche mais Le grand mérite de la loi (dans son article 2) est de définir précisément ce qu'est une donnée personnelle ainsi que les traitements qui s'y appliquent.

1.2.2 Loi relative à la protection des personnes physiques

Au cours de l'année 2004 ,la loi du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés a été modifiée en apportant des éclaircissements . C'est la Loi n 2004-801 du 6 août 2004 relative à la protection des personnes physiques à l'égard des traitements de données à caractère personnel ,le principe général posé par la paragraphe 3 du nouvel article 32 est clair : en cas de collecte indirecte , il y a un devoir d'information du responsable de la collecte indirecte envers la personne concerné ,Ceci vise à protéger le propriétaire de donnée .

la loi explique en détail ce qu'il faut entendre par "traitement de données à caractère personnel" il s'agit de toute opération ou de tout ensemble d'opération portant sur de telles données, quel que soit le procédé utilisé, et notamment la collecte, l'enregistrement, l'organisation, la conservation, l'adaptation ou la modification, l'extraction, la consultation, l'utilisation, la communication par transmission, diffusion ou toute autre forme de mise à disposition, le rapprochement ou l'interconnexion, ainsi que le verrouillage, l'effacement ou la destruction.

La loi donne également une définition large de la notion de "données à caractère personnel", puisqu'il s'agit de toute information relative à une personne physique identifiée ou qui peut être identifiée, directement ou indirectement, par référence à un numéro d'identification, ou à un ou plusieurs éléments qui lui sont propres

1.3 Législation Algérienne

L'Algérie a commencé à prendre des décisions juridiques pour protéger les données personnelles a partir la début du 21 siècle par la loi n 2000-03 du 5 Joumada El Oula 1421 correspondant au 5 août 2000, modifiée et complétée, fixant les règles générales relatives à la poste et aux télécommunications qui balise principalement les modalités de protection des données transmises par voie par les voie du secteur postal (télécommunication téléphonique , télex , les colis postaux ...) , après l'évolution des TIC et pour les droits moraux et

patrimoniaux de l'auteur et protégé leurs œuvres par L'ordonnance n 03-05 du 19 Joumada El Oula 1424 correspondant au 19 juillet 2003 relative aux droits d'auteur et aux droits voisins . La fin des dix premiers années 2000 , L'informatique est devenu un moyen important pour stocker des informations personnelles. Dans le but de sécuriser les échanges des données personnelles , la loi n 09-04 du 14 Chaâbane 1430 correspondant au 5 août 2009 est adopté. il apporte des règles particulières relatives à la prévention et à la lutte contre les infractions liées aux technologies de l'information et de la communication. pour appliquer ces règles, un organe national de prévention et de lutte contre la criminalité liée aux technologies de l'information et de la communication [Art. 13 page 7 JO N 47] est cré .

récemment l'Algérie a commencé de moderniser le secteur de la justice qui est un secteur très sensible. pour cela elle est oblige de régler cette révolution par la promulgation de la loi n 15-03 du 11 Rabie Ethani 1436 , qui défini un système centralisé pour le traitement automatisé des données informatiques relatives a l'activité du ministère de la justice et des Établissements qui en relèvent ainsi que des juridictions de l'ordre judiciaire ordinaire, de l'ordre judiciaire administratif et du tribunal des conflits.[Art. 2 page 4 , JO N 06] , toujours dans la même direction et pour la sécurisation de de leur système , la Loi n 15-04 du 11 Rabie Ethani 1436 correspondant au 1er février 2015 fixant les règles générales relatives à la signature et à la certification électroniques.

1.3.1 Loi relative à la poste et aux télécommunications

cette loi a pour objet de fixes les règles générales relatives a la poste et aux télécommunication et pour de fournir des services de qualité et spécifiquement assurés des conditions objectives, transparentes et non discriminatoires dans un environnement concurrentiel tout garantissant l'intérêt général .

1.3.2 L'ordonnance relative aux droits d'auteur et aux droits voisins

cette ordonnance a pour objet de définir les droits d'auteur et les droits voisins, ainsi que les œuvres littéraires ou artistiques protégées et fixer les sanctions des préjudices subis par la violation de ces droits et Les dispositions de cette ordonnance garantissent la protection ces droits.[4]

La protection est accordée, quel que soit le genre, la forme et le mode d'expression, le mérite ou la destination de l'œuvre, dès la création de l'œuvre, que celle-ci soit ou non fixée sur un support permettant sa communication au public.

1.3.3 Loi relatives à la prévention et à la lutte contre les infractions liées aux TIC.

Loi N 09-04du14 Chaâbane 1430 correspondant au 5 août 2009 est apparue essentiellement pour la prévention et à la lutte contre les infractions liées aux technologies de l'information et de la communication [5].

1.3.4 Loi relative à la modernisation de la justice

La présente loi [6] vise a moderniser le fonctionnement de la justice algérienne a travers :

- La mise en place d'un système informatique centralisé du ministère de la justice, ce système est sécurisé par la certification électronique qu'elle bien détaillé en section 1.6.6.
- La communication des documents judiciaires et des actes de procédure par voie Électronique, l'utilisation de la visioconférence dans les procédures judiciaires.

1.3.5 Loi relative à la signature et à la certification électroniques

Il est indispensable pour instaurer toutes sorte d'une gouvernance électronique de disposer d'une infrastructure règlementaire et technique pour la gestion des signatures et certificats électroniques. Le texte 15-04 [7] est considéré aujourd'hui'hui comme la base réglementaire en Algérie dans cette orientation.

Dans cette loi nous trouvons essentiellement la définition de la nomenclature officielle relative à ce domaine où nous retenons l'essentiel dans le tableau suivant :

Nomenclature	Explication
Signature électronique	données sous forme électronique, jointes ou liées logiquement à d'autres données électroniques, servant de méthode d'authentification.
Signataire	personne physique qui détient des données de création de signature électronique, agissant pour son propre compte ou pour celui de la personne physique ou morale qu'elle représente .
Certificat électronique	document sous forme électronique attestant du lien entre les données de vérification de signature électronique et le signataire.
Clé cryptographique privée	chaîne de chiffres détenue exclusivement par le signataire et utilisée pour créer une signature électronique, cette clé est liée à une clé cryptographique publique.
Clé cryptographique publique	chaîne de chiffres mise à la disposition du public afin de lui permettre de vérifier la signature électronique, elle est insérée dans le certificat électronique.
Autorisation	désigne le régime d'exploitation de services de certification électronique et se matérialise par le document officiel délivré au prestataire de manière personnelle lui permettant de commencer la fourniture effective de ses services.
Tiers de confiance	personne morale qui délivre des certificats électroniques qualifiés ou éventuellement fournit d'autres services en matière de certification électronique au profit des intervenants dans la branche gouvernementale.
Prestataire de services de certification électronique	personne physique ou morale qui délivre des certificats électroniques qualifiés et fournissant éventuellement d'autres services en matière de certification électronique.
Politique de certification électronique	Ensemble des règles et procédures organisationnelles et techniques liées à la signature et à la certification électroniques.

Table 1.1: définition de la nomenclature officielle

Le texte de la loi 15-04 consacre la protection des données personnelles non pas seulement par l'application des différentes techniques cryptographique mais aussi par l'interdiction dans l'article 5 de toutes transfert des données utilisée dans le cycle de l'infrastructure technique hors de celui-ci que dans les cas prévus par la législation en vigueur.

1.4 Définition de la sécurité informatique

a travers l'étude les différentes législations qui on a fait dans la section précédente nous constatons que la sécurisation des données personnelles est une opération très importante pour protéger de toutes types de vulnérabilités .

la sécurité des données d'après Information Technology Security Evaluation Cri-teria (ITSEC) est basée sur trois propriétés : ***confidentialité*** - ***intégrité*** - ***disponibilité***.

Nous travaillons dans cette section à présenter les aspects de la sécurisation des données informatiques et leur utilité , la question qui se pose quelles sont ces techniques ?

1.5 Objectifs fondamentaux de la sécurité informatique

La sécurité informatique consiste à garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu La sécurité des systèmes d'information vise les objectifs suivants [2] :

- *Confidentialité* : S'assurer que personne ne peut lire le message, sauf le récepteur prévu.
- *Authentification*: Le processus de prouver son identité.
- *Intégrité*: Assurer au récepteur que le message reçu n'a pas été altéré de quelque façon que ce soit de l'original.

- *Non-répudiation*: Un mécanisme pour prouver que l'expéditeur a vraiment envoyé ce message.

1.6 Cryptologie

La cryptologie est une mécanisme permettant de camoufler des messages, de le rendre incompréhensible pour quiconque n'est pas autorisé. Elle fait partie d'une ensemble de théories et de techniques liées à la transmission de l'information [3] .

Cryptologie = *Cryptographie* + *Cryptanalyse.*

On distingue deux grands types d'algorithmes de chiffrement, les algorithmes a clef secrète(classique) et les algorithmes a clef publique (moderne).

1.6.1 Cryptographie Classique(Symétrique)

La cryptographie symétrique, est la plus ancienne historiquement. Elle est extrêmement répandue à cause de ses performances remarquables. Elle suppose qu'au moins deux personnes partagent la connaissance de la même clé secrète, ce qui leur confère donc un rôle symétrique. Elle s'appuie principalement sur les fonctions booléennes et les statistiques.

Exemple : A veut transmettre à B un message M

A:M claire +{Ks}==>M chiffrée ——> **B**: M chiffrée +{Ks}==> M claire

Figure 1.1: Chiffrement Symétrique

1.6.2 La cryptographie Moderne(Asymétrique)

La cryptographie asymétrique évite le partage d'un secret entre les deux interlocuteurs [8]. Dans un système de chiffrement a clef publique, chaque utilisateur dispose d'un couple de clefs, une clef publique qu'il met en général a disposition de tous dans un annuaire, et une clef secrète connue de lui seul. Pour envoyer un message confidentiel a Mohamed, Ali chiffre donc le message clair a l'aide de la clef publique de Mohamed. Ce dernier, a l'aide de la clef secrète correspondante, est seul en mesure de déchiffrer le message reçu.

Exemple :

1- Kp est la clé publique (le cadenas), que vous pouvez révéler à quiconque. Si A veut envoyer un message a B, il transmet Kp(message).

2- Ks est la clé secrète (la clé du cadenas), elle reste en possession de B.Il décodez le message en calculant Ks(Kp(message))=message.

3- La connaissance de Kp par un tiers ne compromet pas la sécurité de l'envoi des messages codés, puisqu'elle ne permet pas de retrouver Ks. Il est possible de donner librement P, qui mérite bien son nom de clé publique.

A:M claire +{Ks}==>M chiffrée ——> **B**: M chiffrée +{Kp+Ks}==> M claire

Figure 1.2: Chiffrement Asymétrique

1.6.3 Algorithmes de cryptage symétriques et asymétriques

il existe plusieurs algorithmes spéciaux de chaque technique de cryptage mentionné dans les sections 1.7.1 et 1.7.2 , la figure suivante montre les algorithmes de chaque une :

1.6.4 Algorithmes des cryptage symétrique

comme montré sur la figure il existe Deux catégories des algorithmes de cryptage symétrique Chiffrement par bloc (Block Cipher) et Chiffrement par flot (Stream Ciphers) et chaque catégorie composée de plusieurs techniques .

1.6.5 Algorithmes des cryptage asymétrique

Nous sommes plus intéressés dans cette mémoire par ce type de cryptage , pour cette raison nous allons décrire brièvement chaque technique notamment la technique RSA [9] .

- **Rivest Shamir Adleman (RSA) :** cet algorithme a été publier en 1978 par Rivest-Shamir-Adleman , elle est utilisée pour crypter et signer les données et L'algorithme RSA de base pour la confidentialité peut être explique comme suit :
  Texte chiffré = (texte clair)^e mod n
  Texte clair = (texte chiffré)^d mod n
  Clé privée = (d, n) Clé publique = (e, n)
  L'algorithme RSA de base pour l'authentification peut être explique comme suit.
  Texte chiffré = (texte clair)^d mod n
  Texte clair = (texte chiffré)^e mod n
  Clé privée = (d, n)Clé publique = (e, n)

- **Diffie-Hellman :**Développé par Dr. Whitfield Diffie et Dr. Martin Hell-man en1976 .Il n'est pas destiné au chiffrement ou au déchiffre-

ment , mais il permet à deux parties impliquées dans la communication de générer une clé secrète partagée pour l'échange confidentiel d'informations.

- **Elliptic Curve Cryptography (ECC):** a cryptographie à courbe elliptique (ECC) fournit une fonctionnalité similaire à RSA. (ECC) est mise œuvre dans de plus petits appareils comme les téléphones cellulaires.
  Il nécessite moins de puissance de calcul par rapport à RSA. Les systèmes de cryptage ECC sont basés sur l'idée d'utiliser des points sur une courbe pour définir la paire de clés publique / privée.

- **El Gamal:** est un algorithme utilisé pour transmettre des signatures numériques et des échanges de clés. La méthode est basée sur le calcul des logarithmes et les caractéristiques des nombres logarithmiques et des calculs. L'Algorithme de Signature Numérique (DSA) est basé sur El Gamal .

1.6.6 Certificats numériques

Le certificat est un ensemble d'informations qui identifie un utilisateur ou un serveur et contient des informations telles que le nom de l'organisation qui a émis le certificat, l'adresse e-mail , le pays de l'utilisateur et la clé publique de l'utilisateur[10]. Ces certificats peuvent servir à l'authentification des utilisateurs du système d'in- formation pour contrôler l'accès à certaines applications ou à certaines données, notamment des publications en ligne .

1.6.7 Fonction de hachage

Une fonction de hachage est aussi appelée fonction de hachage à sens unique ou "one-way hash function" en anglais.
Ce type de fonction est très utilisé en cryptographie, principalement dans le but de réduire la taille des données à traiter par la fonction de cryptage. En effet, la caractéristique principale d'une fonction de hachage est de produire

un haché des données, c'est-à-dire un condensé de ces données. Ce condensé est de taille fixe, dont la valeur diffère suivant la fonction utilisée : nous verrons plus loin les tailles habituelles et leur importance au niveau de la sécurité [11].

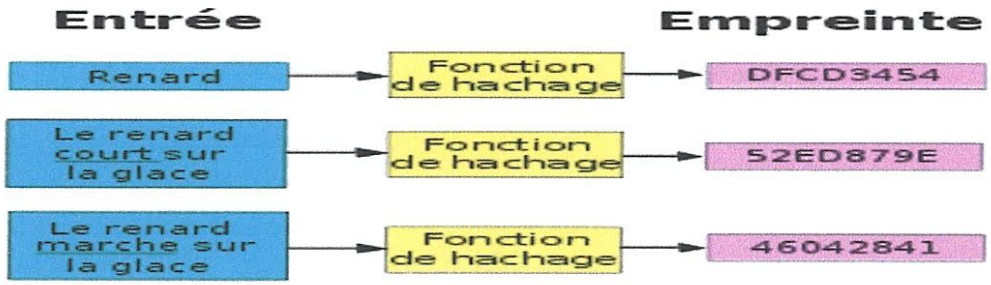


Figure 1.3: Fonction de Hachage

1.6.8 Signature numérique

La signature numérique est un procédé qui garantit l'authenticité d'un document,, ainsi que son intégrité donc le fait que le document n'a pas été modifié. Techniquement elle se présente comme une suite de chiffres dont la combinaison avec le document signé est suffisamment complexe pour qu'il soit impossible de falsifier l'une ou l'autre de façon indétectable.
La signature numérique est réalisée au moyen d'un certificat électronique, équivalent numérique de la carte d'identité.

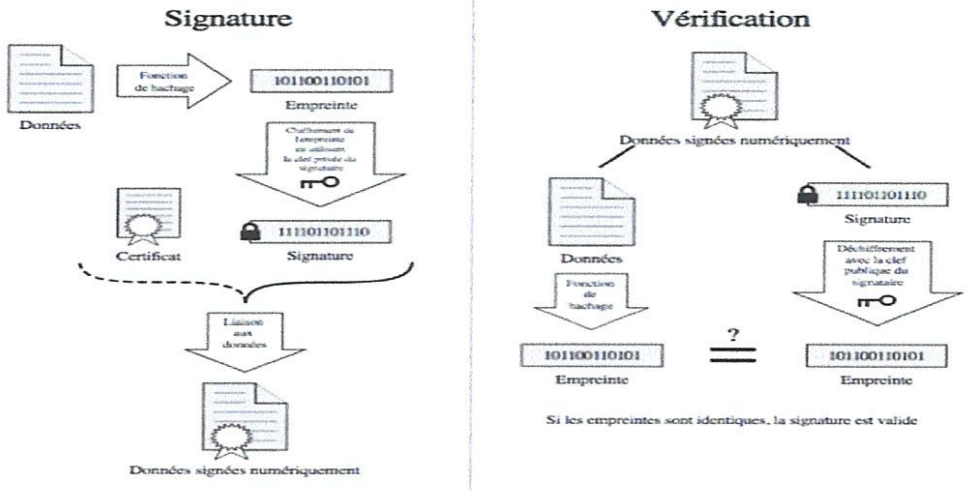


Figure 1.4: signature et vérification des données

1.6.9 PKI (*Public Key Infrastructure*)

PKI (*Public Key Infrastructure*) est un système de gestion des clés publiques qui permet de gérer des listes importantes de clés publiques et d'en assurer la fiabilité, pour des entités généralement dans un réseau. 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. [12].
Une infrastructure PKI fournit donc quatre services principaux:

- Fabrication de bi-clés.
- Certification de clé publique et publication de certificats.
- Révocation de certificats.
- Gestion la fonction de certification.

Donc PKI est une structure à la fois technique et administrative, avec 80 % d'organisationnelle et 20 % de technique. Le domaine des PKI est intéressant.

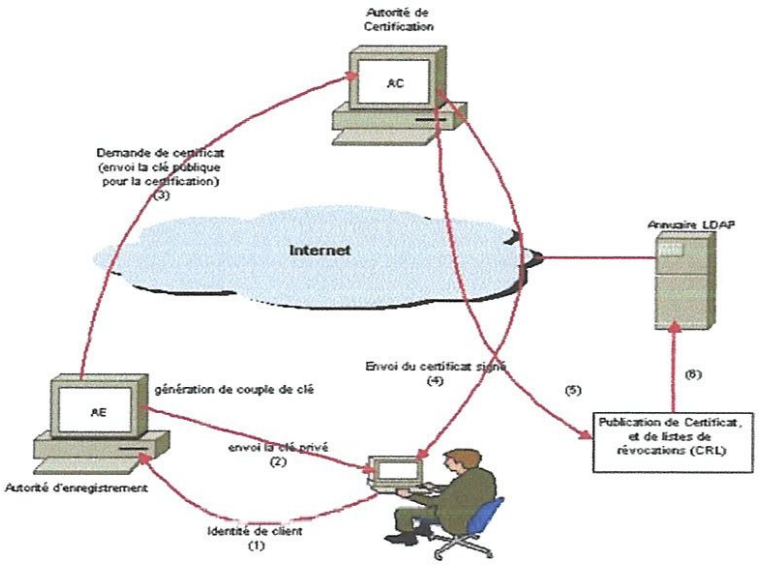


Figure 1.5: Organisation d'une PKI

1.7 Conclusion

Dans ce chapitre, nous sommes intéressés par deux principes qui sont liés directement à la confidentialité des données personnelles, les textes juridiques encadrant la protection des données à caractère personnel à l'échelle international et nous avons pris la législation française (CNIL), puis l'Algérienne a partir la début du 21 siècle, jusqu'à la Loi N 15-04 qu'elle considéré comme la base de règlementation Algérienne fixant les règles générales relatives la signature et la certification électroniques.

Ensuite, nous avons fait le point sur l'aspect technique de la sécurité informatique en général et la cryptologie en particulier ,et nous montrons les deux modèles de cryptographie symétrique, asymétrique et les signatures, certificats électronique. Ces études donnent l'occasion a identifier le PKI (*Public Key Infrastructure*), qui permet de gérer les certificats électronique et d'en assurer la fiabilité de transmission d'information.

Chapitre 2

Contrôle d'accès

2.1 Introduction

La confidentialité des données est devenue un aspect important dans la plupart des domaines tel que le commerce électronique, la gestion de données personnelles, des systèmes d'information en ligne, et même pour la préservation de secrets scientifiques ou industriels. Dans le chapitre précédant, nous avons discuté les concepts fondamentaux de la sécurité informatique dans ses aspects juridique et technique.

Dans ce chapitre , nous étendons ces concepts par l'exploration du domaine du contrôle d'accès. qui est parmi les techniques les plus efficaces et le plus utilisée dans le but de se protéger ou préserver la confidentialité .Le besoin de contrôle d'accès apparaît naturellement lorsqu'un système multi-utilisateur offre un accès sélectif à l'information partagée. L'objectif de ce chapitre est de décrire les différents modèles de contrôle d'accès et de présenter les avantages et les inconvénients de chacun d'eux.

2.2 Définitions

plusieurs définitions existantes dans la littérature, nous retenons les définitions suivantes :

Le contrôle d'accès physique : est un dispositif permettant un accès contrôlé à un lieu, un bâtiment, un local, une machine ou des équipements spécifiques (comme un coffre ou un véhicule)
Le contrôle d'accès logique : est un système de contrôle d'accès à un système d'information. Il est souvent couplé avec le contrôle d'accès physique et permet de restreindre le nombre d'utilisateurs du système d'information.

Il existe deux classes de ressources dans n'importe quel système d'information: les sujets qui se réfèrent a des entités actives telles que des utilisateurs ou des programmes exécutés en leur nom, et des objets qui sont des entités passives telles que des fichiers, des dossiers, des imprimantes. La façon dont un sujet accède à un l'objet est appelé privilège d'accès (ou mode d'accès). Le privilège d'accès permet aux sujets de soit manipuler des objets (lire, écrire, exécuter, etc.) ou modifier les informations de contrôle d'accès (transfert de propriété, privilèges, etc.). L'exactitude du contrôle d'accès dépend fortement de ce qui suit:

- L'identification adéquate du sujet
- La protection des mécanismes de contrôle d'accès

Le contrôle d'accès peut être basé sur des politiques différentes. Le choix de la politique de sécurité Est important car il influence la flexibilité, la facilité d'utilisation et la performance de la système. En établissant une politique de sécurité appropriée, il faut procéder avec Il selon "un bon guide de conception" qui comprend les principes suivants [15]

2.2.1 Principe de privilège minimum et maximum

Selon le principe de privilège minimum, les sujets devraient utiliser le minimum Ensemble de privilèges nécessaires à leur activité . Le principe de privilège maximal est fondé sur le principe de la disponibilité maximale des données. Les sujets ont accès à la plus grande gamme de information.

2.2.2 Principe du système ouvert vs fermé

Dans le système ouvert, tous les accès qui ne sont pas explicitement interdits sont autorisés. Tandis que dans un système fermé, tous les accès ne sont autorisés que s'ils sont explicitement autorisés. Un système ouvert offre une grande souplesse lorsque le système fermé est intrinsèquement plus sécurisé. La combinaison entre systèmes est possible, une autorisation positive (Lorsque l'accès est autorisé) et une autorisation négative (lorsque l'accès est interdit) peut être exprimé, ce qui entraîne des conflits lorsque des autorisations opposées sont spécifiées pour le même sujet et sur le même objet. Ainsi, en plus des politiques de contrôle d'accès, les politiques de résolution des conflits doivent également être définies.

2.2.3 Principe d'administration centralisé ou décentralisé

Le principe traite de la question «*qui est responsable de la maintenance et Gestion des privilèges dans le modèle de contrôle d'accès* ". Dans une administration centralisée, une seule autorité (ou groupe) contrôle tous les aspects de sécurité du système, tandis que dans un système décentralisé, différentes autorités contrôlent différentes parties d'une base de données par exemple. Le choix entre administration centralisée ou décentralisée doit être effectué selon certains critères liés à la nature de les exigences de sécurité des systèmes et des utilisateurs. Cependant, certains choix intermédiaires comme: la délégation, l'administration propriétaire et l'administration coopérative peuvent être adopté:

Délégation

Peut être utilisé dans un système de base des données centralisé pour éviter les encombrements et soutenir l'autonomie locale dans un système de base des données distribué. Une autorité centrale délègue leurs droits administratifs pour un sous-ensemble de la base de données locale les autorités. L'autorité centrale peut désigner et congédier les autorités locales.

d'autres sujets par les propriétaires d'objets. Dans DAC, la manière dont les sujets individuels manipulent des données spécifiquesles objets sont spécifiés par des règles d'accès explicites.le DAC a été étudié dans le cadre du modèle de matrice de contrôle d'accès.Ce modèle a été développé par Lampson [18] et prolongé par Graham et Denning [17]. Plus tard, Harrisson, Ruzzo et Ullman [19] ont développé un plus général version du modèle. Ils ont défini le problème de sécurité et ont montré que c'était indécidable.

Le modèle de matrice d'accès est défini pour trois composantes: *sujets* (entités actives tels que les utilisateurs, leurs processus, etc.), les *objets* (entités passives telles que les fichiers, les enregistrements,classes, instances, vues, etc.) et les *privilèges* (lire, écrire, supprimer, créer, exécuter, etc.). La classe d'objets contient tous les sujets.

La matrice d'accès a est défini comme suit: Les lignes sont indexées par des sujets (leurs noms) (S) et des colonnes par noms de tous les objets (O). Chaque entrée A [s, o] contient une collection des privilèges détenus par le sujet s à l'objet o. Une représentation de la matrice de contrôle d'accès est indiquée dans (le tableau 2.1):

sujcts	objcts
	o1 . . . . . oj . . . . . om
S1	A[s1, o1] .... A[s1, oj]... A[s1, om]
Si	A[si, o1]..... A[si , oj]..... A[si,om]
.	
.	
Sn	A[sn, o1] ... A[sn, oj] ... A[sn, om]

Table 2.1: Matrice de contrôle d'accès

La matrice de contrôle d'accès est dispersée. Le stockage direct de la matrice gaspille beaucoup de mémoire. Une solution simple est de le stocker comme une séquence de deux lignes (Liste de Capacité) ou des colonnes (Liste de contrôle d'accès). Les listes de capacités (CL) permettent au système de identifiez rapidement la collecte de tous les objets accessibles pour un sujet donné. Accès les listes de contrôle (ACL) sont associées à leurs objets (colonnes du contrôle d'accès matrice A), pour un objet donné, cette liste

comprend toutes les entrées non-vides du colonne de A. ACL permet une identification rapide des sujets qui peuvent accéder à un objet, et ils sont principalement utilisés dans les systèmes de base de données.

La matrice de contrôle d'accès peut être étendue. Chaque entrée A [s, o] de la matrice contient (Une partie des privilèges) une condition appropriée qui doit être satisfaite par le sujet S pour accéder à l'objet O. La condition peut intégrer différents types d'accès tels que dépendants du contenu, dépendant du contexte, etc. Fernandez [20] et Conway [21] ont montré comment généraliser le modèle de matrice d'accès en utilisant les prédicats et les autres composants. Cependant, le modèle de matrice d'accès fournit un modèle flexible qui peut être utilisé pour analyser les propriétés de sécurité. Il est bien connu que le problème de sécurité général est indécidable[1] [15], c'est-à-dire qu'il n'y a pas d'algorithme qui puisse être utilisé pour vérifier la sécurité du modèle de matrice de contrôle d'accès. Mais il est encore possible de restreindre le modèle et de concevoir un algorithme qui peut être utilisé pour prouver certaines propriétés de sécurité. Certains travaux ont été réalisés pour étendre le modèle de matrice d'accès pour rendre le problème de sécurité décide. Cela inclut le modèle de protection schématique [22] et le modèle de matrice d'accès dactylographié [23]. Donc on peux déduire , qu'il existe en pratique trois approches pour implémenter la matrice d'accès:

1. **Table d'autorisation** : les entrées vides de la matrice ne sont pas reportées dans la table. La table est composée de trois colonnes qui correspondent aux sujets, aux actions et aux objets. Chaque n-uplet de la table correspond à une autorisation.

2. **ACL** (*Access Control List*) : la matrice est stockée par colonne. Chaque objet est associé à une liste indiquant pour chaque utilisateur les actions pouvant être exercées par ce dernier sur cet objet(Figure 2.1).

3. **Liste de Capacité** (*capability*) : la matrice est stockée par ligne. Chaque utilisateur a une liste voir Figure appelée une liste de capacité,

[1] en logique, qualité d'une proposition, d'un énoncé qui ne peut être ni démontré ni réfuté

indiquant pour chaque objet les actions que l'utilisateur est autorisé à effectuer sur cet objet

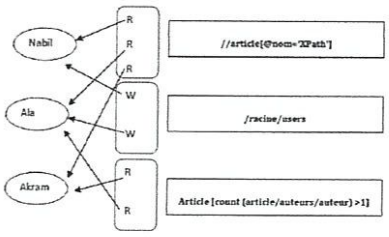


Figure 2.1: Acess Control List (ACL)

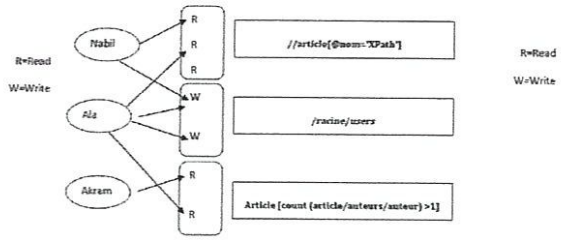


Figure 2.2: Capability List (CL)

2.3.2 Contrôle d'accès obligatoire (*MAC*)

Le contrôle d'accès obligatoire (MAC) a été développé pour établir un réseau des politiques de confidentialité [24] face aux chevaux de Troie [2] Les politiques de ce modèle régissent les accès aux données selon les sujets sur la base d'une classification prédéfinie des sujets et des objets dans le système. La classification est basée sur un ensemble de classes d'accès (souvent appelées étiquettes) associées à chaque sujet et objet dans le système (Figure 2.3).

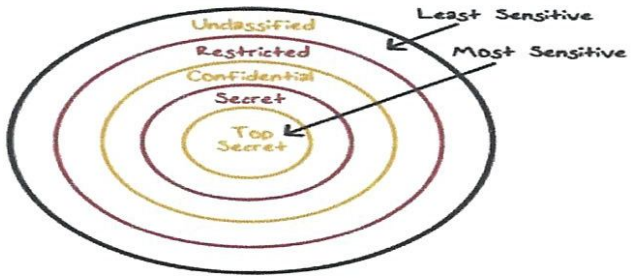


Figure 2.3: Niveaux de sécurité

Un sujet a accès à un objet donné si et seulement si une relation d'ordre, Selon le mode d'accès, sont satisfaite par la classe d'accès de l'objet et l'objet.

[2]Trojan est un programme dans lequel une partie nuisible (par exemple, un sous-programme qui provoque des fuites d'informations sur des lecteurs non autorisés) est contenue dans un code apparemment inoffensif

En ce qui concerne les classes d'accès de sujet et d'objet, les auteurs du modèle (Bell et LaPadula) [25] ont défini des propriétés de contrôle d'accès conformément aux règles de remorquage: propriété de *Simple security property*(Figure 2.1) et *Star Property*(Figure 2.2). Dans la propriété de sécurité simple, un sujet est autorisé à lire l'accès à un objet si la classe d'accès du sujet domine la classe d'accès de l'objet.

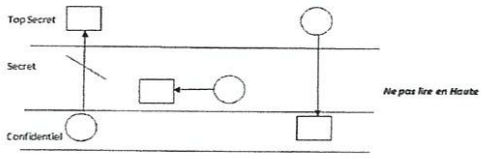


Figure 2.4: Propriété de sécurité simple

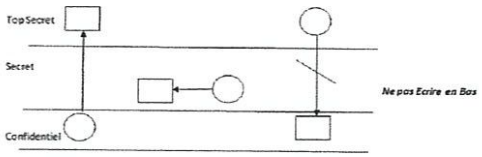


Figure 2.5: Propriété de sécurité étoile

Dans *Star Property*, un sujet est autorisé à accéder en écriture à un objet si la classe d'accès de l'objet **domine** la classe d'accès du sujet. Le modèle Bell et LaPadula propose des classes d'accès en tant que paire de composants: un niveau de sécurité et un ensemble de catégories. Les niveaux de sécurité peuvent inclure Non classés (Unclassified), Confidentiel (C), Secret (S) et Top Secret (TS), où TS > S> C> U. L'ensemble de catégories est un ensemble non ordonné (Ex. OTAN, nucléaire, armée).

2.3.3 Contrôle d'accès basé sur les rôles (*RBAC*)

Le contrôle d'accès basé sur les rôles (RBAC)[26] a reçu une attention considérable en tant que alternative prometteuse aux contrôles d'accès DAC et MAC traditionnels. RBAC est le modèle de contrôle d'accès standard actuel et un axe de recherche depuis deux décennies. Le paradigme RBAC encapsule les privilèges en rôles et les utilisateurs sont affectés à des rôles pour acquérir des privilèges, ce qui le rend simple et facilite l'examen des autorisations attribuées à un utilisateur. Cela rend également la tâche de l'administration des politiques moins encombrante, car chaque changement dans un rôle se reflète immédiatement sur les autorisations disponibles pour les utilisateurs

affectés à ce rôle, la figure illustre les relations entre les utilisateurs, les rôle et permissions .

Une étude [27] indique que l'adoption du RBAC dans les organisations commerciales augmente constamment.

Le modèle est basé sur trois ensembles d'entités appelés User U, rôles R et Permissions P [28]. Intuitivement, le User est un utilisateur (humain ou agent autonome), un rôle est un une action ou un travail au sein de l'organisation avec des sémantiques associées concernant a l'autorité et à la responsabilité confiées au membre du rôle. L'autorisation est l'approbation d'un mode d'accès particulier à un ou plusieurs objets dans le système. la relation d'affectation les Permissions (P A) aux Utilisateurs (U A) .

Sont des relations de plusieurs à plusieurs. Un utilisateur peut être membre des plusieurs rôles, et un rôle peut avoir nombreux utilisateurs. De même, un rôle peut avoir de nombreuses autorisations, et la même permission peut être attribuée à plusieurs rôles. Il existe une hiérarchie de rôle RH, également écrite comme

$$x \geq y$$

signifie que le rôle x hérite des autorisations attribuées au rôle y. L'héritage de la hiérarchie des rôles est transitif et l'héritage multiple est autorisé dans les commandes partielles. La figure (2.6) montre un ensemble de sessions S. Chaque session relie un utilisateur à éventuellement plusieurs rôles. Intuitivement, un utilisateur établit une session au cours de laquelle l'utilisateur active un sous-ensemble de rôles auxquels il est membre . Plusieurs rôles peuvent être activés simultanément. Les autorisations disponibles pour un utilisateur sont l'union des autorisations de tous les rôles activés dans cette session. Chaque session est associée à un seul utilisateur. Cette association reste constante pour la vie d'une session.

Le modèle RBAC est Caractérisé par:

- la notion que les permissions sont attribuées aux rôles, et pas directement aux utilisateurs.
- Les usages sont affectés Rôles appropriés en fonction de leurs fonctions

professionnelles, et Acquérir indirectement les autorisations associées à Ces rôles.

- RBAC simplifie la gestion du contrôle d'accès car les permissions sont déterminés en fonction des rôles et cette association ne change pas souvent.

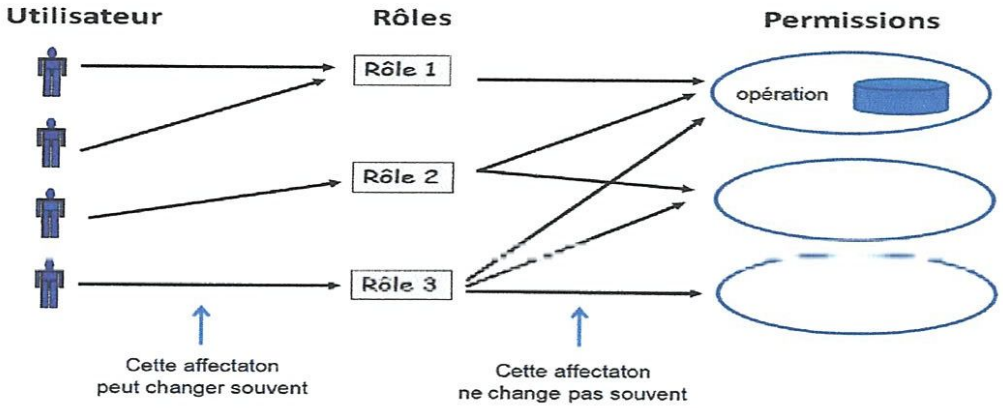


Figure 2.6: Modèle conceptuel du RBAC

2.4 Conclusion

Le contrôle d'accès est un aspect fondamental de la sécurité et est primordial pour la protection privée et confidentielle Information des cybercriminalité. Compréhension Les bases du contrôle d'accès sont fondamentales pour Comprendre comment gérer la sécurité de l'information.

Nous avons étudies dans ce chapitre les modèle (MAC,DAC,RBAC), et malgré les différences théoriques et techniques entre eux mais ils ont été développés au cours des décennies dans le but d'améliorer la confidentialité, l'intégrité, la disponibilité et flexibilité d'administration.

Chapitre 3

Introduction à XML

3.1 Introduction

XML (*eXtensible Markup Language*) [35] est un langage de balise a été créé en 1998 afin que les documents structurés puissent être utilisés ,stockés et transmettre, il arrive souvent que les documents XML contiennent des informations de différents degrés de sensibilité qui doivent être partagés entre les utilisateurs Il existe donc une nécessité de la conception des mécanismes permettant l'organisation et la spécification les droits de chaque utilisateur. Les Politiques de contrôle d'accès pour les documents XML nous permettons a atteindre cet objective .

Dans ce chapitre nous introduisons les principes du langage XML les interets du XML les modèles de contrôle d'accès pour les documents XML pour donner des rôles a chaque utilisateurs sur les documents ciblés, selon les autorisations donner a chaque personne.

3.2 Intérêts du XML

XML est devenu un élément indispensable dans le monde de l'informatique, Cela est dû aux caractéristiques importantes qui sont disponibles dans cet outil , ses nombreux caractéristiques montrent la haute qualité de ce langage et son succès :

- Format libre.
- Format texte avec gestion des caractères spéciaux.
- Séparation stricte entre contenu et présentation.
- Simplicité, extensibilité et universalité.
- Structuration forte.
- Structuration des documents (DTD et Schémas XML).

3.3 Syntaxe et Structuration des documents XML

un document XML en général composer de trois parties :

- le prologue permet d'indiquer la norme XML et les caractères (encoding) utilisée pour comprendre tous types de caractère spéciaux.
- déclaration de type de document et Document Type Définition (DTD).
- l'arbre des éléments les parents et fils.

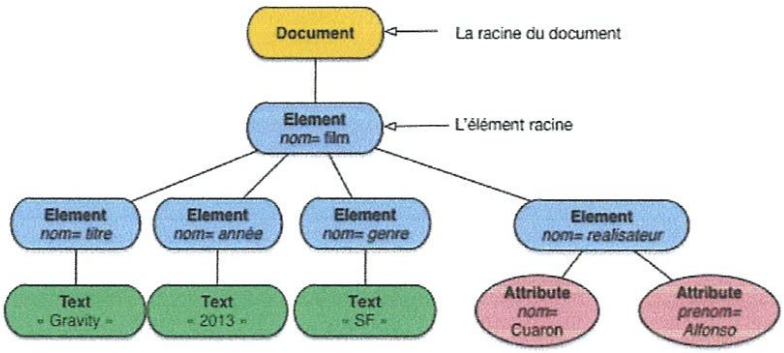


Figure 3.1: structure d'un document XML

3.4 Validation du document XML

La validation est un processus par lequel un document XML est validé et garantit que la structure de données utilisée respecte ce schéma.

Document Type Définition (DTD)[36] et *XML Schéma* (XSD)[35] sont deux principaux mécanismes de spécification de validation. Avant que document XML puisse être Validé et utilisé, il doit être analysé par des parseurs (analyseurs) XML. On dit qu'un document XML est valide si ses contenus s'accordent avec les éléments, les attributs et la déclaration de type de document (DTD) associée ou avec le XML Schéma. La validation est distribuée de deux façons par le parseur XML. Ils sont: ***bien formé*** et ***Valide***

3.4.1 La document type definition (DTD)

DTD(*Document Type Declaration*) est la première et la plus ancienne langue à définir la structure et le contenu de Documents XM .
DTD vérifié le vocabulaire et la validité de la structure de documents XML contre les règles grammaticales de langue XML appropriée. Un DTD peut être spécifié à l'intérieur du document (DTD interne), ou il peut être gardé dans un document séparé (DTD externe).

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2  <!DOCTYPE personne [
3  <!ELEMENT personne (prenom, nom)>
4  <!ELEMENT prenom (\#PCDATA)>
5  <!ELEMENT nom (\#PCDATA)>
6                fin de la DTD interne
7  ] >
8                debut du document
9  <personne>
10    <prenom>Alaeddine</prenom>
11    <nom>Bougherara</nom>
12 </personne>
13               fin du document
```

Figure 3.2: Document XML Validé par le DTD

3.4.2 XML Schema

Schéma XML décrit la structure d'un document XML,la langue de Schéma XML est aussi appelée *XML Schema Definition* (XSD), il peut aussi Interpréter tel qu'un fichier XML par les parseurs XML Le but d'un schéma XML est de définir les blocs de construction légal d'un document XML:

- les éléments et les attributs présentent dans le document.
- le nombre des éléments fils.
- les types des éléments et attributs.
- les valeurs par défauts et fixées de chaque élément ou attribut.

Éléments

les éléments sont les éléments constitutifs du document XML. Un élément peut être défini dans un XSD comme suit:

```
1 <xs:element name="x" type="y"/>
```

Types de définition

- ***Type simple***: l'élément de type simple est utilisé uniquement dans le contexte du texte. Certains types simples prédéfinis sont: xs: integer, xs: boolean, xs: string, xs: date. exemple:

```
1     <xs:element name="phone_number" type="xs:int" />
```

- ***Type complexe***: un type complexe est un conteneur pour les autres définitions d'éléments. Cela nous permet de spécifier les éléments.

exemple:

```
<personne>
            <nom>Mohamed</nom>
            <prenom>Ali</prenom>
</personne>
<personne sexe="feminin">Axel ROBERT</personne>
```

Les attributs

Les attributs dans XSD fournissent des informations supplémentaires dans un élément. Les attributs ont la propriété name et type comme indiqué ci-dessous:

```
    <xs:attribute name="country" type="xs:string" />
```

3.5 Exploration des données XML

XML est considérée comme une structure arborescente. L'arbre est construit sur les tags, avec des nœuds, des éléments, des attributs et des groupes d'attributs, et des nœuds enfants de sous-éléments et d'attributs.

L'utilisateur ou les applications peuvent utiliser des langages spécifiques tel que XPath et XQuery pour localiser et accéder à des éléments à partir de documents XML. Une caractéristique importante est que XPath et XQuery[35] peut sélectionner des nœuds en fonction des attributs d'un document XML.

3.5.1 XPath

XPath[37] est une langue (est présenté comme une norme par W3C) qui permet d'extraire des informations , des éléments ou des attributs spécifiques. Cela fonctionne de la même façon que les chemins d'accès d'un système de fichiers, en commençant par la racine et progressant dans les différentes couches jusqu'à ce que la cible soit trouvée. exemple :

```
<utilisateur>
          <contact>
               <nom>BOUGHERARA </nom>
               < prenom >Alaeddine</prenom >
               <mobile>+21312345678</mobile>
          </contact>
</utilisateur>
```

Dans notre exemple, le but va être de récupérer le prénom de l'utilisateur, commençons par décrire les étapes à suivre:

- Etape 1 : descendre au nœud "utilisateur" .
- Etape 2 : descendre au nœud contact.
- Etape 3 : descendre au nœud prénom.

Ce qui nous donne :

```
/utilisateur/contact/prénom/
```

3.5.2 XQuery

XQuery[38] est une langue permettant de rechercher et de manipuler tout ce qui peut être représenté en tant qu'arbre à l'aide du modèle de données XQuery et XPath. Les expressions XQuery peuvent accéder à plusieurs documents, voire plusieurs bases de données, et extraire des résultats très efficacement. XQuery est Semblable à SQL pour les BD sauf que SQL travaille sur des bases de données relationnelles: données fortement en relation.

Une requête XQuery est composée de trois parties :

- une ligne d'entête commencée par "XQuery" et contenant la version et, éventuellement l'encodage du document ;
- un ensemble de déclarations :

- déclarations de variables ou constantes.
- déclarations de fonctions utilisateur locales.
- importation de modules (bibliothèques XQuery).
- détermination du schéma du résultat.
- détermination des espaces de nom et de leur utilisation.
- détermination du format du résultat et autres paramètres .

la Figure 3.5 montre un exemple d'une requête XQuery

```
1  for $x in doc("books.xml")/bookstore/book
2  where $x/price>30
3  order by $x/title
4  return $x/title
```

Figure 3.3: requête XQuery

3.6 Manipulation XML

XML permet de définir la structure du document, ce qui permet d'une part de pouvoir définir séparément la présentation de ce document, d'autre part d'être capable de récupérer les données présentes dans le document pour les utiliser. Le parseur a pour rôle d'analyser le document XML et de servir de lien avec une application de traitement. Il existe des parseurs non validant qui n'offrent qu'une vérification syntaxique et des parseurs validants qui offrent également le support des DTD/schéma W3C. Sur ces deux catégories de parseurs se greffent principalement deux catégories de services :

un service événementiel, qui ne vise pas à représenter un document XML dans son intégralité, de type SAX (Simple API for XML), par exemple, et un service objet, qui permet de représenter un document XML sous une forme objet, par exemple DOM (Document Object Model). Dans le premier cas, la représentation du document n'est que partielle, alors que dans le second cas, elle est complète.

3.6.1 DOM (*Document Object Model*)

Document Object Model (DOM) est un API , qui permet de lire un document XML et d'extraire des informations différentes (éléments, Attributs, commentaires, etc.) pour les exploiter.

Lorsque un document XML est lu par un analyseur DOM, le document est représenté en mémoire en tant qu'arbre dans lequel les différents éléments sont liés l'un à l'autre par une relation parent / fils.

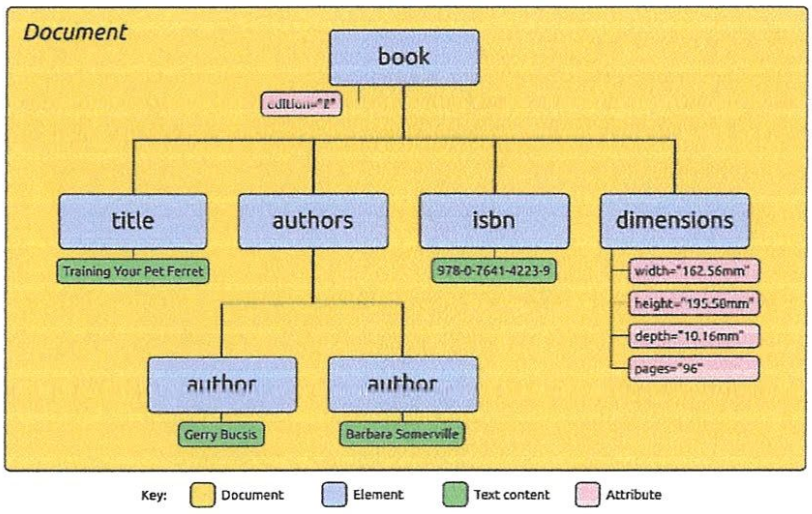


Figure 3.4: Une représentation simplifiée du *Document Object Model*

3.6.2 SAX

SAX (Simple API for XML) définit un mode de communication entre le parseur et l'application, lié à un mécanisme événementiel. C'est un projet Open Source (sax.sourceforge.net/) qui propose deux versions d'une API, où seule la deuxième version est capable de prendre en compte les espaces de noms . Plutôt que de représenter la totalité du document XML en mémoire, le parseur réalise un découpage du document en petites unités et transmet ces unités à l'application au fur et à mesure de l'analyse du document.

- déclarations de variables ou constantes.
- déclarations de fonctions utilisateur locales.
- importation de modules (bibliothèques XQuery).
- détermination du schéma du résultat.
- détermination des espaces de nom et de leur utilisation.
- détermination du format du résultat et autres paramètres .

la Figure 3.5 montre un exemple d'une requête XQuery

```
1    for $x in doc("books.xml")/bookstore/book
2    where $x/price>30
3    order by $x/title
4    return $x/title
```

Figure 3.3: requête XQuery

3.6 Manipulation XML

XML permet de définir la structure du document, ce qui permet d'une part de pouvoir définir séparément la présentation de ce document, d'autre part d'être capable de récupérer les données présentes dans le document pour les utiliser. Le parseur a pour rôle d'analyser le document XML et de servir de lien avec une application de traitement. Il existe des parseurs non validant qui n'offrent qu'une vérification syntaxique et des parseurs validants qui offrent également le support des DTD/schéma W3C. Sur ces deux catégories de parseurs se greffent principalement deux catégories de services :

un service événementiel, qui ne vise pas à représenter un document XML dans son intégralité, de type SAX (Simple API for XML), par exemple, et un service objet, qui permet de représenter un document XML sous une forme objet, par exemple DOM (Document Object Model). Dans le premier cas, la représentation du document n'est que partielle, alors que dans le second cas, elle est complète.

Une unité représentera, par exemple, l'ouverture d'un élément ou sa fermeture, la rencontre d'un texte... On le comprendra, dans ce système, l'application n'a pas de représentation globale du document ou plutôt le parseur ne fournit qu'un cheminement dans le document, que l'application est libre de stocker ou non.

Feature	**DOM**	**SAX**
Difficulté d'utilisation	Difficile	Moyen
Capacité xpath	Oui	Non
Navigation complete	Oui	Oui
Lire XML **Ecrire dans XML**	Oui Oui	Oui Non

Table 3.1: Tableau comparatif entre le DOM et SAX

3.7 Conclusion

Comme nous l'avons vu, l'objectif du XML est de faciliter les échanges de données entre les machines. A cela s'ajoute un autre objectif important : décrire les données de manière aussi bien compréhensible par les hommes qui écrivent les documents XML que par les machines qui les exploitent. Le XML se veut donc standardisé, simple, mais surtout extensible et configurable afin que n'importe quel type de données puisse être décrit.
Nous avons étudier dans ce chapitre la structuration général du document XML, les méthodes et techniques pour valider, explorer et manipuler les documents XML par (DTD et XSD),(XPath et XQuery),(DOM et SAX) perspectivement.

Chapitre 4

contrôle d'accès pour XML

4.1 Introduction

Un défi majeur dans la sécurité des Données XML est la conception d'un contrôle d'accès efficace qui peut répondre adéquatement aux défis de sécurité uniques impliqués par leur répartition aspect [30]. Les Données XML permettent aux systèmes d'information de fonctionner dans une plate-forme indépendante, la promotion d'une collaboration et de partage d'informations dans des environnements très dynamiques. Cependant, cela implique un paradoxe de sécurité. D'une part, les systèmes collaboratifs exigent que les informations soient accessibles a tous ceux qui en ont besoin; D'autre part, les organisations doivent continuer à protéger les informations sensibles et confidentielles exposées pensées leurs services.

Les solutions précédentes pour le contrôle d'accès (DAC, MAC et RBAC) sont conçus pour les applications LAN (telles que les applications bancaires), ce qui les rend pas appropriés pour résoudre ce défi, à moins qu'ils ne soient étendus pour répondre à l'exigence spécifique des services Web . Plusieurs modèles ont été proposés pour gérer efficacement le contrôle d'accès dans le contexte de l'environnement distribué. Ces modèles allant de la sécurisation du document XML et de la demande SOAP[1] défaillante au contrôle d'accès

[1]SOAP (Simple Object Access Protocol)Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur

aux services mondiaux. Dans la section suivante, nous essayons de donner un aperçu de certains travaux importants dans ce domaine. Dans ce chpitre nous essyons de montrer quelques modèles utiliser dans la littirature .

4.2 *XACML*: (eXtensible Access Control Markup Language)

XACML (eXtensible Access Control Markup Language)[29] est une langue de contrôle d'accès standard OASIS[2]. XACML décrit à la fois un langage de politique de contrôle d'accès et un langage de demande / réponse. Le langage de politique est utilisé pour exprimer les politiques de contrôle d'accès (qui peut faire quand) alors que le langage de demande / réponse exprime des requêtes sur la question de savoir si un accès particulier doit être autorisé (demandes) et décrit les réponses à ces requêtes (réponses).

XACML standardise trois aspects essentiels du processus d'autorisation décrits dans la Figure (4.1):

- Langage de politique XACML - utilisé pour exprimer les règles et conditions de contrôle d'accès. de nombreuses régles peuvent être combinées dans une seule politique. de nombreuses politiques et ensembles de politiques peuvent être combinés dans des ensembles de politiques plus larges. Les algorithmes de combinaison flexibles déterminent la façon dont les règles sont jointes pour saisir le sens exact des politiques d'entreprise, de même que la grammaire d'une langue naturelle nous permet d'exprimer des directives précises.

- XACML request / response protocol - utilisé pour interroger un moteur de décision qui évalue les requêtes d'accès du monde réel aux politiques XACML existantes. Le résultat, soit autoriser ou Refuser, est retourné en réponse XACML.

[2](Organization for the Advancement of Structured Information Standard) est un consortium mondial sans but lucratif qui stimule le développement, la convergence et l'adoption des normes de commerce électronique. Vous trouverez des informations sur OASIS à http://www.oasisopen.org.

- Architecture de référence XACML - fournit une norme pour le déploiement des modules logiciels nécessaires pour assurer une application efficace des politiques XACML. Au cœur, un Policy Decision Point (PDP) évalue les politiques en fonction des demandes d'accès fournies par Policy Enforcement Points (PEP). Le PDP ou PEP peut également avoir besoin d'interroger un point d'information de stratégie (PIP) pour recueillir des attributs descriptifs sur l'utilisateur ou l'élément d'information auquel l'accès est demandé. Les politiques sont maintenues via un Policy Administration Point (PAP).

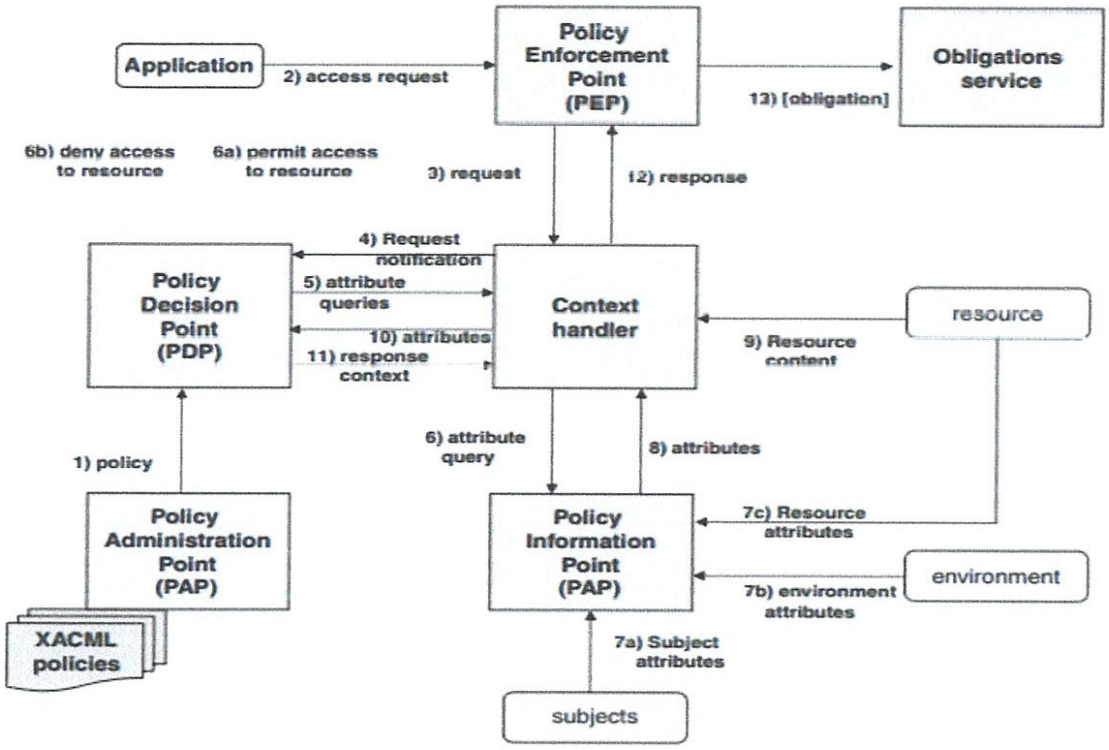


Figure 4.1: Composants principaux XACML

4.3 *WS-AC*: (Contrôle d'accès pour les services Web)

Ce modèle consiste en une approche innovante pour fournir un Mécanisme de contrôle d'accès dans un environnement distribué, comme les services Web. *WS-AC* est un contrôle d'accès au document XML. Il permet d'exprimer, de valider et appliquer des politiques de contrôle d'accès sans assumer la confiance préétablie entre les utilisateurs du service web [31], les conditions d'accès sont exprimées en terme de attributs utilisateur et paramètres caractérisant le service Web. Il est possible de spécifiez qu'un utilisateur peut utiliser un service Web donné mais seulement avec des valeurs spécifiques de paramètres de service.

Les politiques de contrôle d'accès sont spécifiées en contraignant les paramètres et les attributs du service que les sujets devraient posséder afin d'obtenir un service. quand un service demande, le système vérifie les politiques de contrôle d'accès correspondantes, afin d'établir la demande peut être acceptée telle qu'elle est ou doit être rejetée ou doit être négocié. La négociation consiste à éliminer ou à modifier certains paramètres du service.

4.4 *AuthorX:* Contrôle d'accès à Haute précision pour les documents XML

AuthorX est un modèle pour limiter l'accès aux documents XML et aux services Web. Comme XACML, ce modèle exploite les propres capacités de XML pour définir un balisage XML décrivant les exigences de protection des documents XML. La caractéristique principale est qu'elle fournit des autorisations de niveau d'instance et au niveau de schéma avec la granularité du niveau d'élément / d'attribute XML [32] et utilisent la structure hierarchique XML pour les propager à d'autres éléments ou attributs inclus à moins qu'une autorisation plus spécifique soit indiquée.

4.5 Contrôle d'accès basé sur les rôles pour les données XML

Ce modèle présente une nouvelle approche du contrôle d'accès aux données XML car elle prend en charge les politiques de contrôle d'accès pour les services Web global [33] . Il consiste en une adaptation du contrôle des RBAC au contexte des services Web où les sujets sont des services Web et les objets sont des services Web et leurs attributs , Et les autorisations sont définies à la fois sur les services Web individuels, leurs attributs et sur les services globaux (qui sont des services réalisés par composition d'autres services spécifiques). Ce modèle est proposé par Wonohoesodo et al. [34] p divisé en deux parties. Le premier définit un RBAC adapté sur le service Web tandis que le second est étendu pour prendre en charge les services Web global (RBAC).

- (*SWS-RBAC*): Modèle RBAC pour les services Web .
- (CWS-RBAC) : Modèle RBAC pour les services Web globaux.

4.6 Comparaison des modèles de contrôle d'accès

L'objectif de notre travail dans ce chapitre est d'étudier les modèles de contrôle d'accès et extraire les principales fonctionnalités permettant d'améliorer le contrôle du flux d'information. Nous présentons dans cette section, une comparaison entre les modèles présentés selon des critères que nous croyons qu'ils sont important pour un modèle de contrôle d'accès aux données XML. Ces critères tiennent au compte des principes fondamentaux de la conception du contrôle d'accès que nous présentons dans (chapitre 2). Les différences entre les modèles de contrôle d'accès pour les données XML sont résumés dans le tableau (4.1) prendre en compte les critères précédents.

Critères	*XACML*	*AuthorX*	*WS-AC*	*WS-RBAC*
Granularité	Éléments et les attributs	Éléments, les attributs, instance du document, schéma	Éléments et les attributs	Service composé, Service simple
Administration	centralisée	centralisée	centralisée	décentralisée
Prise en charge les groupes et les rôles	supporter	supporter	non supporter	supporter les rôles (global et local)
Prise en charge les conditions	supportée	supportée	supportée	non supportée les rôles
Prise en charge les obligations	supportée	non supportée	non supportée	non supportée
Propagation des politiques	non supportée	supportée	non supportée	non supportée
Négociation	non supportée	non supportée	supportée	non supportée
Extensibilité	extensible	non extensible	non extensible	non extensible
Flexibilité	Possible	Limiter	Flexible	Limiter

Table 4.1: Table de comparaison entre les modèles de contrôle d'accès

4.7 Conclusion

Notre objectif dans ce chapitre est notre contribution principale de ce mémoire est d'empêcher la divulgation des informations aux personnes non concernées grâce aux modèles de contrôle d'accès présentés. Cependant, certains des modèles décrits fournissent des fonctionnalités importantes que nous pouvons exploiter pour définir un contrôle d'accès. Par exemple, la propriété de négociation dans *WS-AC* peut être utilisée pour établir une relation de confiance entre différents modules de contrôle d'accès situés sur différents fournisseurs de services. La propriété Propagation dans *AuthorX* permet de définir des règles de propagation de contrôle d'accès sur des objets.

Comme les rôles sont très importants pour le contrôle d'accès pour faciliter les spécifications de politique. Les conditions sont également indispensables pour énoncer les contraintes de contrôle d'accès. Les obligations ou actions provisoires définies dans XACML régissent et enrichissent les politiques de contrôle de flux d'information en appliquant des opérations de dérivation par rapport à l'évolution du système.

Enfin, les modèles de contrôle d'accès pour les données XML donnent un bon cadre à la conception d'une gestion du contrôle d'accès et l'application de la loi. cependant, l'absence d'une approche claire pour étendre ces modèles pour contrôler aux données sensible rend notre travail difficile. a cause de ce déficit, dans le chapitre suivant nous travaillons à mettre en œuvre notre propre modèle .

Chapitre 5

Conception et Implémentation

5.1 Introduction

En arrivant à ce chapitre, nous avons recueilli tous les ingrédients pour concrétiser notre contribution en matière de contrôle d'accès aux données XML. En effet, dans le chapitre précédent plusieurs modèles existant ont été l'objet de notre étude et analyse. Nous avons réussi à dégager les éléments principaux pour un système de contrôle d'accès et en même temps nous avons mis le doigt sur leurs points faibles. Durant ce chapitre, notre objectif et de concevoir et mettre en œuvre notre propre solution (modèle de contrôle d'accès aux données XML) tout en prouvant sa puissance comparativement avec les modèles existants. La suite de ce chapitre contient en premier lieu la spécification des besoins qu'on veuille à satisfaire, puis une architecture globale de notre solution accompagné de l'ensemble des outils et procédure utilisés. Une conception détaillée sera présenté par la suite avant d'entamer la parité implémentation dans laquelle des captures sont prisé et des expériences pour tester la faisabilité de notre solution sont effectuées.

5.2 Spécification des besoins

Avec l'augmentation de l'utilisation des données XML comme support pour le stockage et l'échange d'information. Le besoin aux solutions de sécurité

pour ce type spécifique des données devient une exigence réelle et principale. Contrôler l'accès aux données XML est évidemment l'un ces besoins. Cependant les solutions de contrôle d'accès générales ne sont pas adaptées à la nature d'XML qui offre une structure textuelle arborescente. Cette arborescence engendre un fort couplage entre les différentes données disponible sur le même document (ou ensemble des documents) XML. il est donc très difficile d'exprimer des règles de sécurité sur un fragment de données indépendamment des autres.

Nous souhaitons alors concevoir et mettre en œuvre une solution de contrôle d'accès adaptée aux données XML. La solution fonctionne en mode Client/Serveur. Le client peut formuler une requête pour accéder à une ressource (document XML, partie d'un document, interrogation des données XML) et l'envoyer à un serveur. Le Serveur -en recevant le requête du client- procède d'abord à l'évaluation de la permission (autorisé ou interdire le client à accéder à la ressource) à travers d'un mécanisme de contrôle de d'accès qui lui est associé. Le mécanisme de contrôle d'accès peut interdire ou autorise l'intégrité de la ressource demandée ou bien juste une partie de cette ressource. Il utilise pour calculer la permission sur une ressource une politique de contrôle d'accès contenant un ensemble des règles d'accès. Chaque règle définit qui accède à quelle ressource et de quelle manière.

La politique de contrôle d'accès doit pouvoir exprimer explicitement l'autorisation et l'interdiction d'accès aux ressources. En cas de conflit (existence des règles contradictoires pour la même ressource) Cette politique doit définir comment régler ce conflit. De plus, lorsqu'aucune règle n'existe pour une ressource demandée, La politique doit préciser comment le serveur doit réagir[1] . La règle d'une politique se compose de cinq (05) composantes :

1. **Le sujet** : c'est une abstraction du client demandeurs d'une ressource, le client peut appartenir à un groupe ou se trouve affecté à un rôle au sein d'une entreprise. De ce fait, le sujet peut désigner un utilisateur simple mais aussi un groupe ou bien un rôle.

[1]Dans cette situation la politique doit préciser si le système de sécurité est ouvert ou bien fermé. Dans le cas où il est ouvert , tout ce qui n'est pas explicitement interdit est autorisé. Le contraire où le système de sécurité est fermé

2. **La ressource** : c'est une description d'un document XML, ou bien un fragment d'un document XML .

3. **L'action** : c'est l'ensemble des facettes dont le sujet veut accéder à la ressource. Les actions peuvent être les éléments de la liste non exhaustive suivante :

   A -Lire

   B -Écrire

   C -Modifier

   D -Ajouter

   E -Supprimer

   F -Transformer

4. **L'effet** : peut prendre deux (02) valeurs : autorisé ou bien interdit.

5. **La liste d'application négative** : en s'adaptant à la nature es données XML, nous souhaitons avoir la possibilité de faire une exception lorsque une règle s'applique sur une ressource XML, cette règle ne s'applique pas à certaines de ses sous éléments. Par exemple dans le cas d'une règle qui autorisé l'accès à une ressource, nous pouvons interdire quand même l'accès à des sous éléments (voir figure 5.1)

En plus de cette composante chaque règle possède les attributs suivants :

1. Type de règle : s'applique pour un utilisateur, groupe ou pour un rôle

2. Identifiant : permettant de localiser et identifier d'une manière unique une règle au sein d'une politique

3. Priorité de la règle : indiquant le niveau d'importance de la règle par rapport d'autre règles dans la même politique

Dans la figure 5.1 la règle autorise au sujet Mohammed de lire les information sur l'utilisateur 1 dans le le document *users.xml* à l'exception de son

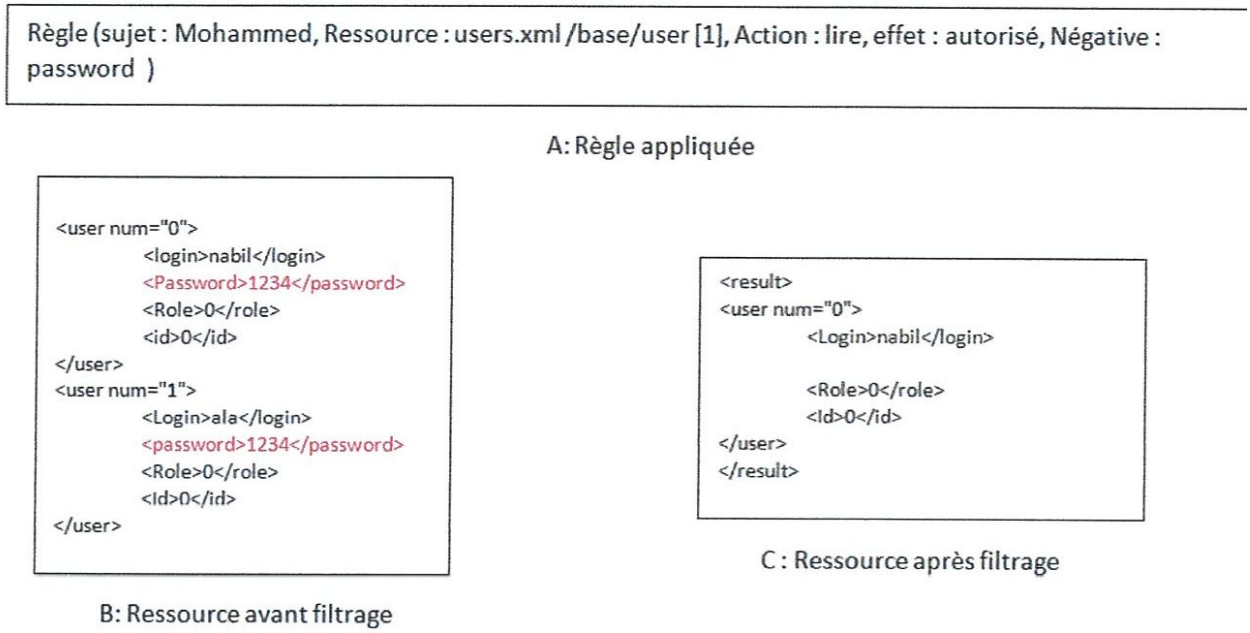


Figure 5.1: liste d'application négative

password. Revenant à la résolution de conflit lorsque plusieurs règles contradictoires sont applicables pour la même ressource. Dans ce cas, plusieurs stratégies sont à adopter. Dans ce qui suit l'ensemble minimal des stratégies fournis :

1. *Allow Over Deny* : en cas de plusieurs règles applicables, si l'une d'entre elles autorisé l'accès. Alors l'accès à la ressource est autorisé.

2. *Deny Over Allow* : en cas de plusieurs règles applicables, si l'une d'entre elles interdit l'accès. Alors l'accès à la ressource est prohibé.

3. *First Applicable Rule* : la première règle selon l'ordre d'écriture est appliquée.

4. *Priority application*: la règle la plus prioritaire est sélectionné parmi les règles applicable pour une ressource demandé.

Dans les deux premier cas, la liste d'application négative est l'union de la

liste d'application négative des règles positive[2] dans les premiers points et des règles négatives[3] dans le deuxième point L'application la liste négative de diffère selon le type de la règle :

1. Dans le cas des règles positives : les éléments qui appartiennent à la liste d'application négative sont supprimé du résultat final .

2. Dans le cas des règles négatives : les éléments qui appartiennent à la liste d'application négative sont ajouté au résultat final si et seulement si 'ils ne contiennent pas des enfants au sens XML.

Comme nous le constatons, plusieurs intervenons sont impliqués dans notre système. Il s'avère donc indispensable de recourir à un mécanisme d'authentification permettant de s'assurer de la bonne identité de tous les utilisateurs du système qui se soit du côté client comme dans le coté serveur. Nous utilisons pour ce propos une plateforme à clé publique basé sur l'encryptage asymétrique avec l'algorithme RSA avec une taille de clé de 2048 bits et des certificats de type X509[39] pour une sécurité parfaitement renforcé.

Avant chaque conversation client/serveur une authentification réciproque doit être effectuée. Cela permet au client de s'assurer qui communique avec le bon serveur. Et au serveur de ne se trompe pas dans le calcul des permissions. Le serveur a le choix de choisir une communication confidentiel en encryptant les données envoyé au client suite à sa demande.

5.3 Architecture de notre solution

En réponse à la spécification des besoins exprimés dans le paragraphe précédents.Nous proposons la solution exprimée dans la figure 5.2 L'architecture de notre solution est composée de trois (03) modules :

[2]Les règles avec un effet autorisé
[3]Les règles avec un effet interdit

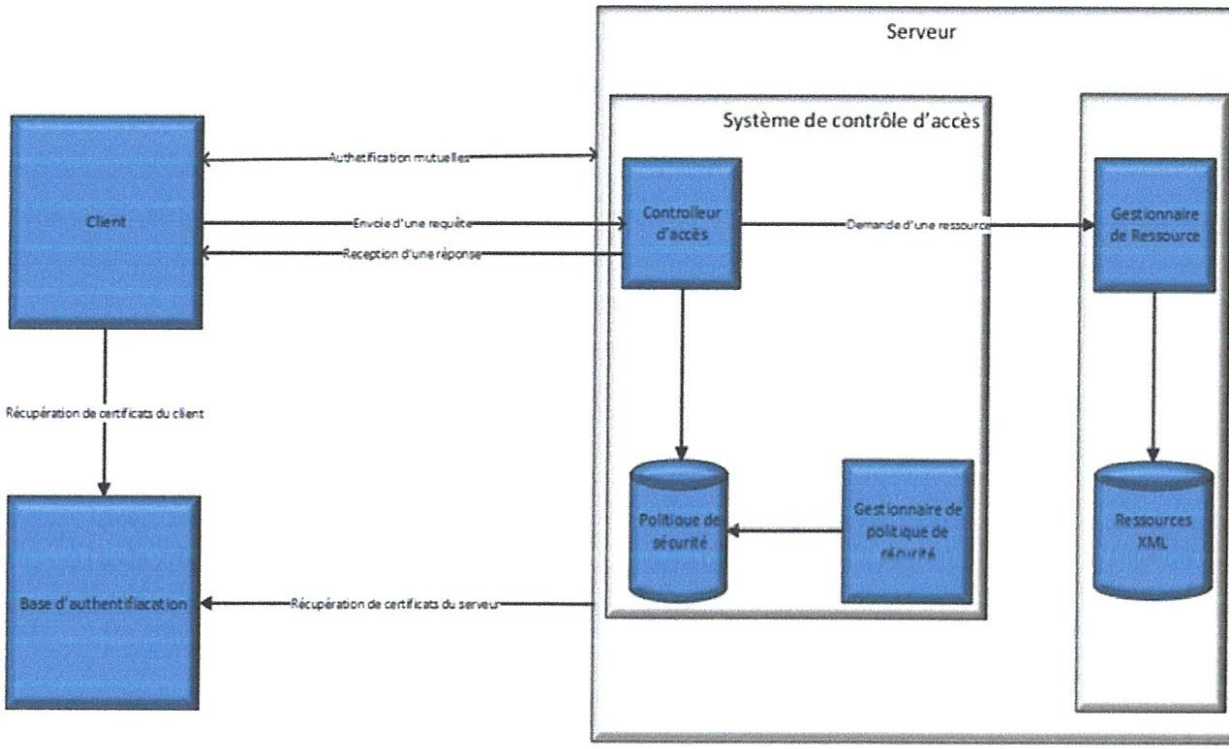


Figure 5.2: Architecture de notre solution

5.3.1 Serveur

le serveur est module principale, il est à son tour divisé en deux sous modules:

1. **Le gestionnaire des ressources :**qui est pour rôle de maintenir la base des ressources en état sein, il a comme tache :

   A -Ajout, modification et suppression des ressources

   B -Exécution des requêtes d'interrogation de type XPath ou XQuery sur des ressources ou un ensemble de ressources.

   C -Répond aux requêtes du contrôleur d'accès

   D -Fournir des informations sur les ressources (exemples des statistiques)

2. **Le système de contrôle d'accès ::** il s'agir de premier récepteur des requêtes des clients. Son rôle est multiple :

A - Gère une ou plusieurs politiques de sécurité à travers son gestionnaire de politique de sécurité.

- Configuration des stratégies de résolution de conflit et le type de sécurité du système.
- Ajout, modification et suppression des règles d'accès

B - Une et une seule politique à la fois.

3. **Calcul des permissions suite à la réception des requêtes auprès des clients.**
4. **Filtrage des ressources reçu du gestionnaire des ressources avant de les remettre au client.**
5. **Remise des ressources filtrées aux clients**

5.3.2 Le client

le client offre à l'utilisateur la possibilité de basculer entre plusieurs serveurs disponibles et de leurs envoyer des requêtes sur leurs ressources disponibles. Le client offre à ses utilisateurs d'effectuer des intégrations pour récupérer uniquement des informations pertinentes.

5.3.3 La base d'authentification

elle agit comme une autorité de certification réduite. Son rôle est de:

1. Générer des pairs de clé et des certificats pour les différents utilisateurs et pour les serveurs.
2. Maintenir une base de donnée de clé (KeyStore) (ajout, modification et suppression)
3. Obtenir des informations sur des certificats stockées
4. Peut exporter des certificats afin qu'ils soient certifiés par des autorités de certification agrées.

5. Répondre aux requêtes des clients et des serveurs pour récupérer des certificats ou des clés privés. dans le cas de demande de clé privé des informations un mot de passe est demandé parce que sauf le propriétaire de cette clé peut la récupérer.

5.4 Conception Détaillée

Après avoir présenté l'architecture de notre solution. Nous passons maintenant à la conception logicielle en s'appuyant sur un formalisme bien connu de modélisation en l'occurrence UML. Nous résumons notre conception dans trois (03) vue essentielles qui sont :

1. Diagramme de cas d'utilisation pour montrer l'interaction entre les exploitants du système et le système lui-même.
2. Diagramme de classe pour illustrer la structure statique de notre système
3. Diagramme de séquence pour voir les différentes interactions au sein de notre système et comprendre les relations entre les différents modules et leurs sous modules.

5.4.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation montre clairement l'existence de trois acteurs dans notre système :

1. Les utilisateurs : tout au début et au lancement de l'application l'utilisateur est invité à s'authentifier. Trois (03) résultats possibles de l'opération d'authentification :

   A - Authentification échouée : un message d'erreur est affiché et l'utilisateur est invité à ressayer.

   B - Authentification réussie avec comme rôle Client

   C - Authentification réussie avec comme rôle Administrateur.

Figure 5.3: Diagramme de cas d'utilisation

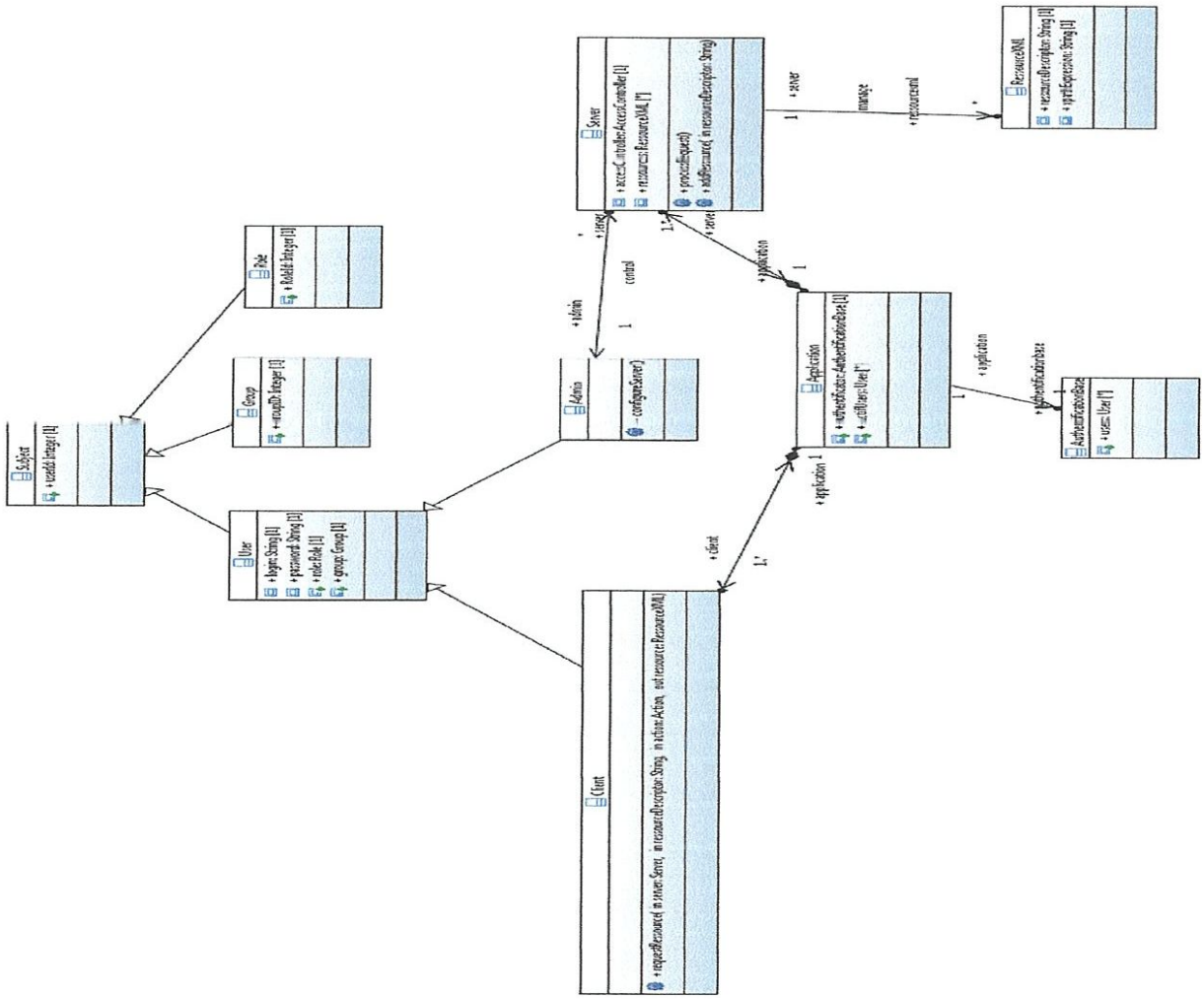
On voit bien qu'il existe une relation d'héritage entre l'utilisateur, le client et enfin l'administrateur.

2. Les clients : sont les utilisateurs qui ont réussi l'étape de l'authentification et possède un rôle autre que administrateur.

3. Les administrateurs : sont les utilisateurs qui ont réussi l'étape de l'authentification et possède un rôle administrateur.

5.4.2 Diagrammes de classe

Les diagrammes de classe de notre application sont divisés en deux vues. La première vue concerne l'organisation structurelle de tourte l'application et la deuxième concerne l'architecture du système de contrôle d'accès . les classes principales de la première partie sont : Server, client, application. On voit aussi la relation d'héritage entre les classes subject , user , client et Admin. Quant au deuxième partie les deux classes principales sont : accessController et Policy



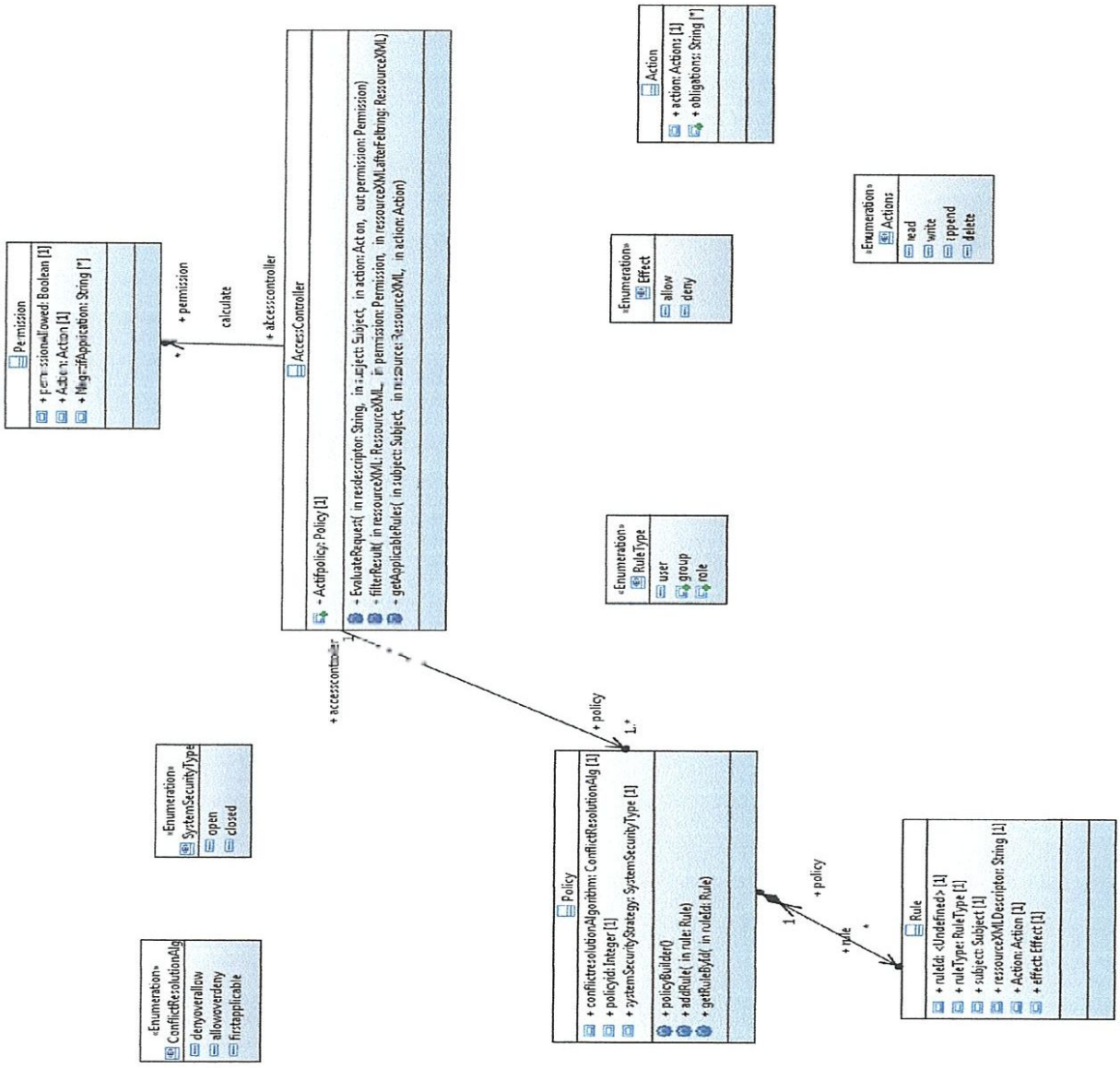


Figure 5.4: Diagrammes de classe

5.4.3 Diagrammes de séquence

Les diagrammes de séquence donnent une vue dynamique de système. Le premier diagramme montre la séquence de création des clients et des serveurs et l'authentification de leurs utilisateurs. Le deuxième diagramme explicite l'exécution du mécanisme du contrôle d'accès applique par notre application.

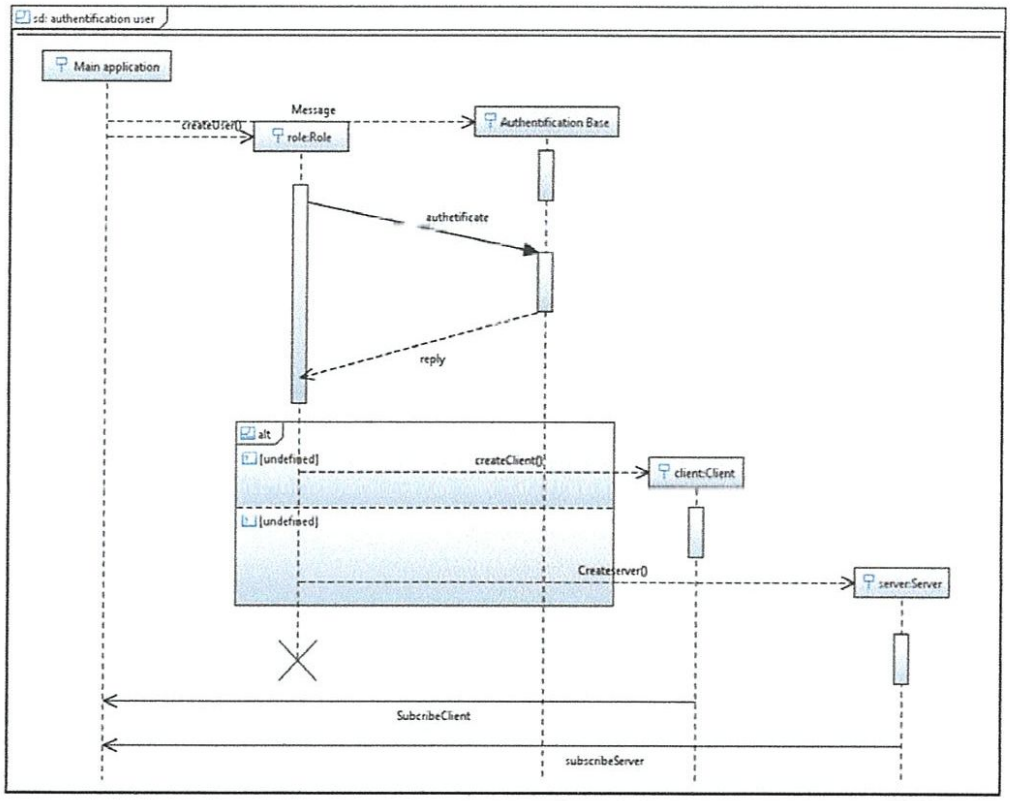


Figure 5.5: Diagramme de séquence d'authentification

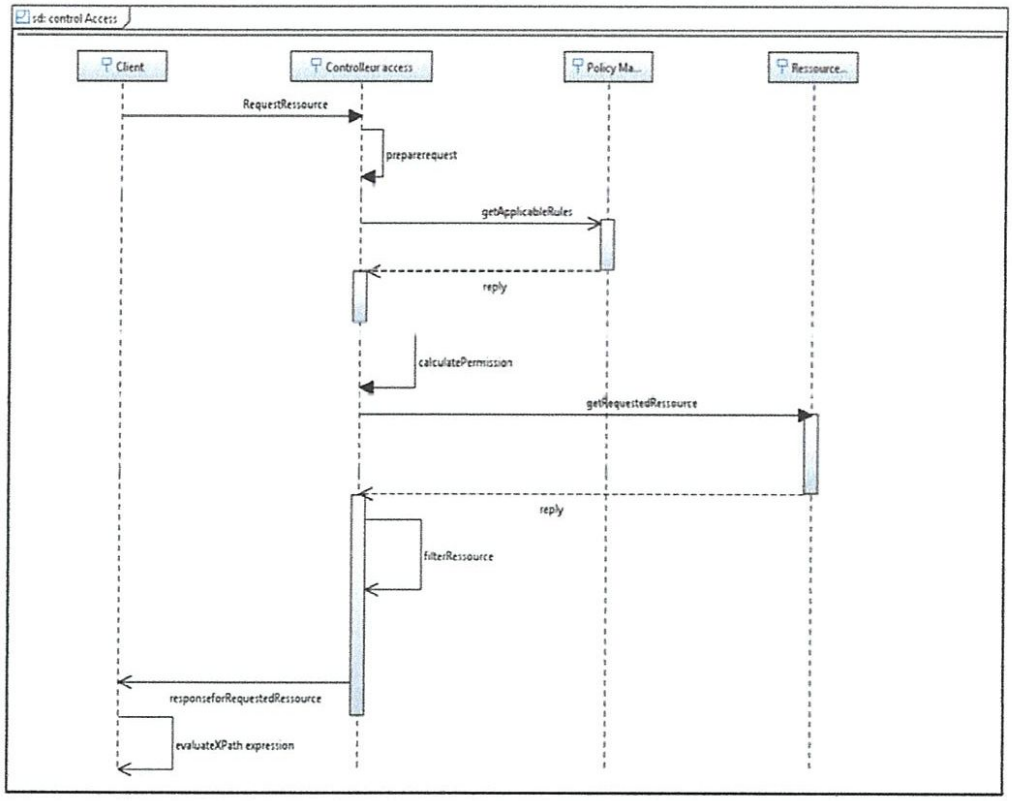


Figure 5.6: Diagramme de séquence de contrôle d'accès

5.5 Implémentation

5.5.1 Outils utilisés

JAVA

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

Eclipse

Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

XML

Extensible Markup Language (XML) est un format de texte simple et très flexible dérivé de SGML (ISO 8879). À l'origine conçu pour relever les défis de l'édition électronique à grande échelle, XML joue également un rôle de plus en plus important dans l'échange d'une grande variété de données sur le Web et ailleurs.

XACML

Extensible Access Control Markup Language (XACML) est un langage de politique de contrôle d'accès et un modèle de traitement qui évalue les demandes d'accès selon les règles définies dans les politique.

XACML est largement utilisé pour découper les applications clientes des décisions d'accès. Bien que les décisions d'accès et les autorisations pour la gestion des ressources puissent être codées par les utilisateurs.

JAXP

JAXP est l'acronyme pour *Java API for XML Processing*, qui est en fait constituée de packages permettant de :

- lire et interpréter un fichier XML ;
- créer des fichiers XML ;
- transformer des fichiers XML.

JAXB

Java Architecture for XML Binding (JAXB)[40] fournit un moyen rapide et pratique de lier des schémas XML et des représentations Java, ce qui facilite les taches pour les développeurs Java d'intégrer des données XML et des fonctions de traitement dans des applications Java.

Dans le cadre de ce processus, JAXB fournit des méthodes pour lire les documents d'instance XML dans des arbres de contenu Java, puis regrouper écrire les éléments de contenu Java dans les documents d'instance XML. JAXB fournit également un moyen de générer un schéma XML à partir d'objets Java.

DOM

Le « *Document Object Model* », ou modèle objet de document, est une API pour les documents HTML et XML. Il fournit une structure de représentation du document, vous permettant de modifier son contenu et sa présentation

visuelle. En résumé, il relie les pages Web à des scripts ou des langages de programmation tel que JavaScript. le tableau 5.1 illustre des méthodes principales de cette API.

Méthode	**Rôle**
short getNodeType()	Renvoyer le type du nœud.
String getNodeName()	Renvoyer le nom du nœud.
String getNodeValue()	Renvoyer la valeur du nœud.
NamedNodeList getAttributes()	Renvoyer la liste des attributs ou null.
void setNodeValue(String)	Mettre à jour la valeur du noeud
boolean hasChildNodes()	Renvoyer un booléen qui indique si le nœud a au moins un nœud fils.
Node getFirstChild()	Renvoyer le premier nœud fils du nœud ou null
NodeList getChildNodes()	Renvoyer une liste des noeuds fils du noeud ou null
Node getParentNode()	Renvoyer le noeud parent du nocud ou null
short getNodeType()	Renvoyer le type du noeud.
Node getPreviousSibling()	Renvoyer le noeud frère précédent
Node getNextSibling()	Renvoyer le noeud frère suivan
Node insertDefore(Node, Node)	Insérer le premier noeud fourni en paramètre avant le second noeud

Table 5.1: Méthodes du DOM

KeyTool

Keytool[41] est un utilitaire de gestion des clés et des certificats. Il permet aux utilisateurs d'administrer leurs propres paires de clés public-privé et les certificats associés pour leur utilisation dans l'auto-authentification (où l'utilisateur s'authentifie auprès d'autres utilisateurs / services) ou de l'intégrité des données et des services d'authentification, en utilisant des signatures numériques. Il permet également aux utilisateurs de mettre en cache les clés publiques (sous forme de certificats) de leurs pairs communicants. Un certificat est une déclaration signée numériquement d'une entité (personne, société, etc.), en disant que la clé publique (et certaines autres informations)

d'une autre entité a une valeur particulière. (Voir les certificats.) Lorsque les données sont signées numériquement, la signature peut être vérifiée pour vérifier l'intégrité et l'authenticité des données. L'intégrité signifie que les données n'ont pas été modifiées ou falsifiées, et l'authenticité signifie que les données proviennent effectivement de ceux qui prétendent l'avoir créé et signé.

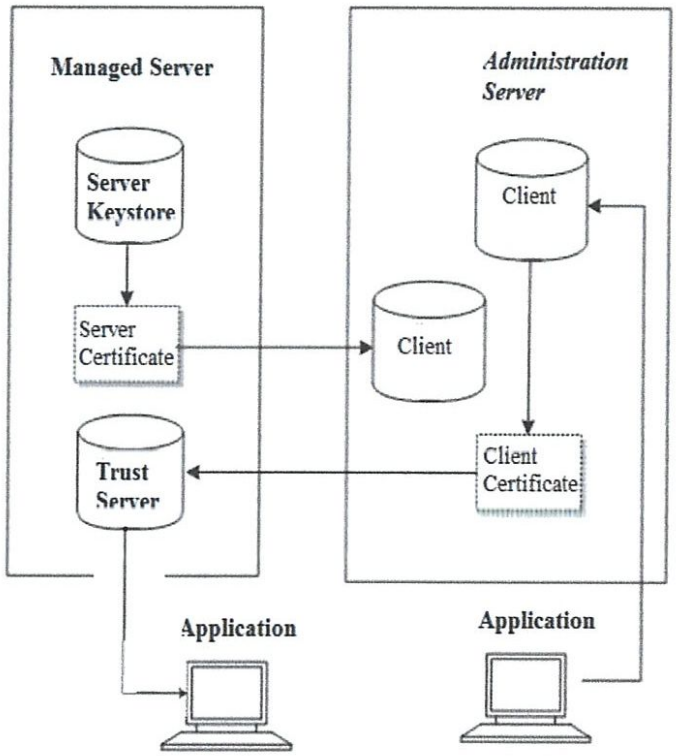


Figure 5.7: L'échange des certificats entre Client est Serveur

5.5.2 Manipulation de notre application

Notre application contient plusieurs fenêtres. Vue le manque d'espace, nous essayons de montrer que quelques captures.

1. **lancement de l'application et authentification**

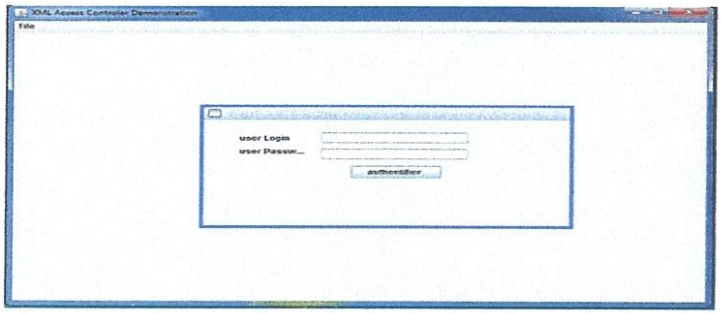


Figure 5.8: Interface d'authentification

2. **Interface Serveur:** constitué d'un menu contenant les différentes options et de deux régions à gauche la liste des ressources chargés et à droite le contenu du ressource sélectionné.

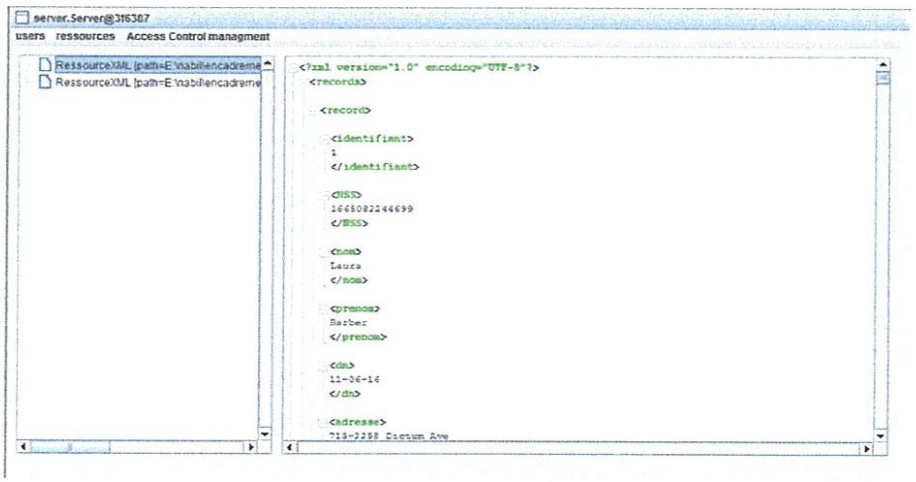


Figure 5.9: Interface Serveur

3. **Interface Client:** constitué d'un ensemble de champs permettant le choix d'un serveur la personnalisation d'une requête. Et d'un espace d'affichage du résultat obtenu auprès d'un serveur.

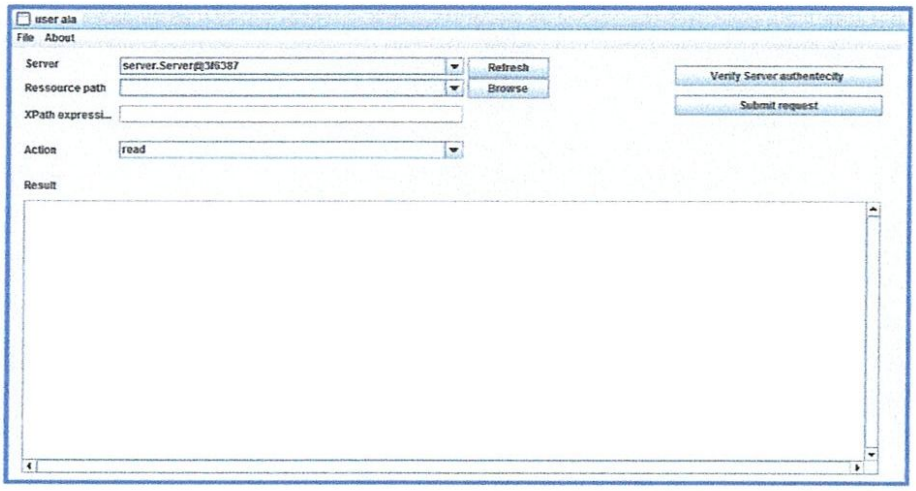


Figure 5.10: Interface Client

4. **Fenêtre de configuration d'une politique:** géré par le serveur, permettant d'ajouter de modifier ou de supprimer des règles.

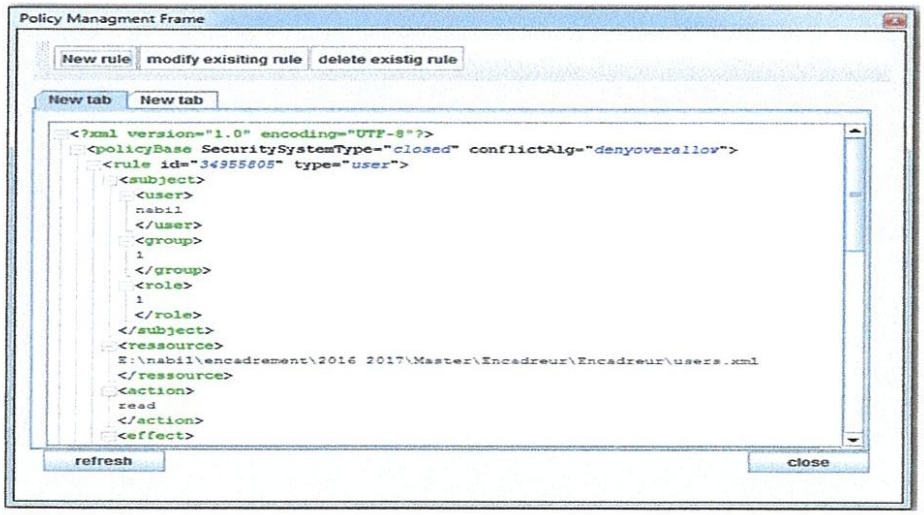


Figure 5.11: Interface manipulation les politiques

5. **Fenêtre d'ajout d'une nouvelle règle à une politique:** permet d'ajouter une nouvelle règle à une politique chargé

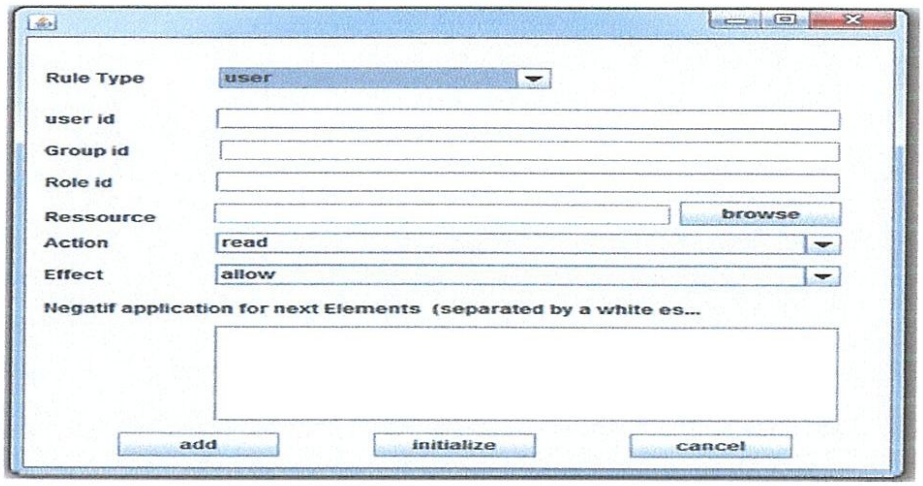


Figure 5.12: Interface d'ajout les règles

5.6 Expérimentation

Afin d'expérimenter notre application, nous avons utilisé une base de données (composé de plusieurs documents XML) contenant des informations médicales. La base de données est générée d'une manière aléatoire à partir du site `generatedata.com`

Notre base de données contient fichier XML :

1. **paitents.xml** :contient les informations générales concernant les patients (identifiant du patient, numéro de sécurité sociale nom, prénom, date de naissance, adresse, numéro de téléphone, email).

2. **medicale.xml** :contient les informations médicales sur les patients (id du patient, médecin traitant, date d'admission, description pathologique, régime alimentaire suivie).

3. **comptabilite.xml**: contient des informations financières sur le patient (id du patients, numéro CCP clé CCP, Numéro carte de crédit, code secret, totales payé, reste à payer).

```
1   <?xml version="1.0" encoding="UTF-8" ?>
2   <records>
3   <record>
4   <identifiant>1</identifiant>
5   <NSS>1665082244699</NSS>
6   <nom>Laura</nom>
7   <prenom>Barber</prenom>
8   <dn>11-06-16</dn>
9   <adresse>713-2258 Dictum Ave</adresse>
10  <phone>05 25 31 09 87</phone>
11  <email>sed@libero.net</email>
12  </record>
13  <records>
```

Figure 5.13: Résultat de génération

Pour contrôler l'accès à cette base de données, les règles suivantes sont appliquées :

1. Les docteurs, les infirmiers et les comptables peuvent consulter l'intégrité des enregistrements des patients sans exception.

2. L'utilisateur 'Saïd' n'a pas le droit de voir les numéros des téléphones et l'email des patients.

3. Sauf les docteurs peuvent consulter les dossiers médicaux des patients.

4. Les infirmiers peuvent savoir la date d'admission, et le régime alimentaires suivi par les patients.

5. L'infirmier « Khaled » peut voir uniquement l'identifiant du patient et la description de son pathologie sans connaitre son nom.

6. Les comptables peuvent voir toutes les informations financières du patient sauf le code de la carte de crédit.

7. Sauf le docteur traite du patient et le patient lui même peut avoir le code de la carte de crédit du patient

8. En cas ou plusieurs règles sont applicable pour une situation donnée, l'interdiction passe en priorité.

9. Si aucune règle ne s'applique pour une situation donnée. L'interdiction passe en priorité.

10. Chaque client doit s'authentifier en fournissant un certificat valide lors de la l'envoie de la requête.

régle	User	Rôle	Group	Ressource	action	effect	Negatif application
1	-	-	docteur	Patients.xml	read	Allow	-
	-	-	infirmiers	Patients.xml	read	Allow	-
	-	-	comptables	Patients.xml	read	Allow	-
2	Saïd	-	-	Patients.xml	read	Deny	Phone,E-mail
3	-	-	docteur	medicale.xml	read	allow	-
4	-	-	infirmiers	Patients.xml	read	deny	Date
5	Khaled	-	infirmiers	Patients.xml	read	deny	ID, NSS
6	-	-	comptables	comptabilite.xml	read	deny	ID, CCP, code secret, totales payé, reste à payer
7	-	-	docteur	comptabilite.xml	read	allow	-
	user	-	-	comptabilite.xml			-

Table 5.2: Table des règles

5.7 Étapes de réalisation de l'expérimentation

(a) Génération des paires de clé et des certificats pour les utilisateurs du système (clients et administrateurs serveurs création des, avec l'outil *KeyTool* disponible avec le kit JDK de java. Afin de faciliter la tâche, nous avons utilisé l'application *keyStore* explorer qui offre une interface graphique pour la gestion du keystore.

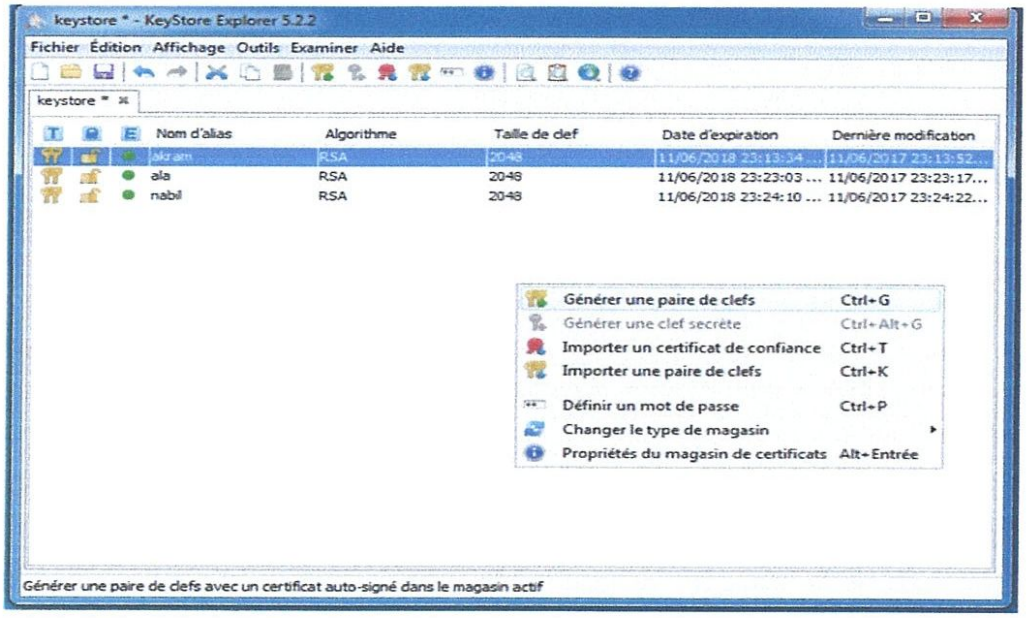


Figure 5.14: Génération des paires de clé par Keytool

(b) Configurations des utilisateurs, de leurs rôles et de leurs groupes.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<base>
  <user num="0" type="0">
       <login>nabil</login>
       <password>123456</password>
       <role>none</role>
       <id>0</id>
       <group>patients</group>
  </user>
       <user num="1" type="0">
       <login>ala</login>
       <password>123456</password>
       <role>chefService</role>
       <group>docteurs</group>
       <id>0</id>
  </user>
  <user num="2" type="1">
      <login>akram</login>
      <password>123456</password>
      <role>ServerAdmin</role>
      <group>none</group>
      <id>0</id>
  </user>
</base>
```

Figure 5.15: Extrait de user base

(c) Traductions des règles dans notre modèle : La figure suivante montre l'ajout de la règle selon la description donnée.

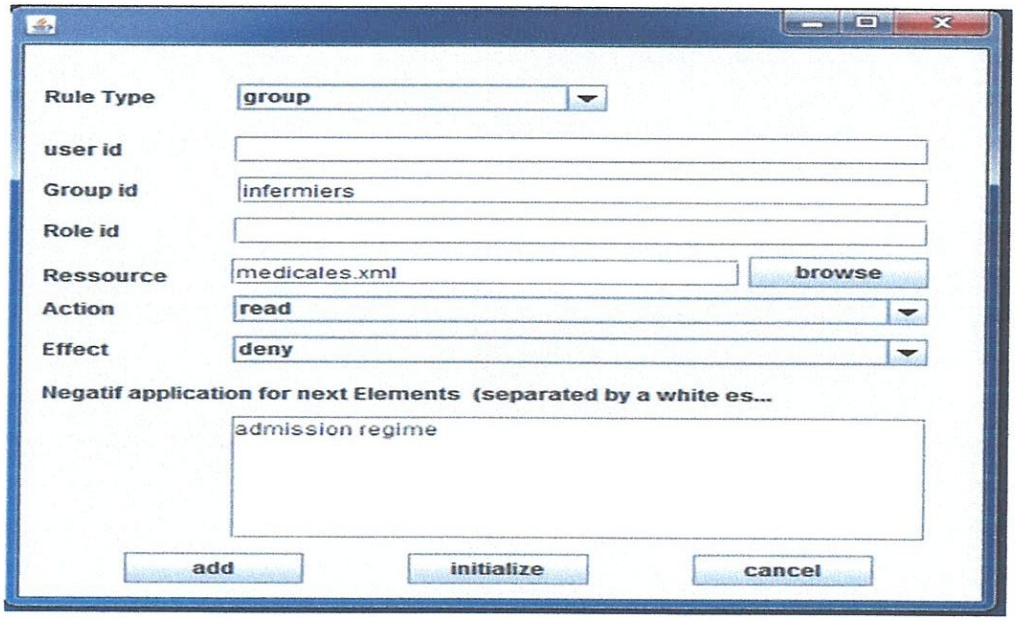


Figure 5.16: Interface de l'ajout d'une règle

(d) Lancements des requêtes et interprétation des résultats : les extrait suivantes montres un extrait du document patients.xml avant (à gauche) et après (à droite) l'exécution d'une requête par l'utilisateur 'Saïd'

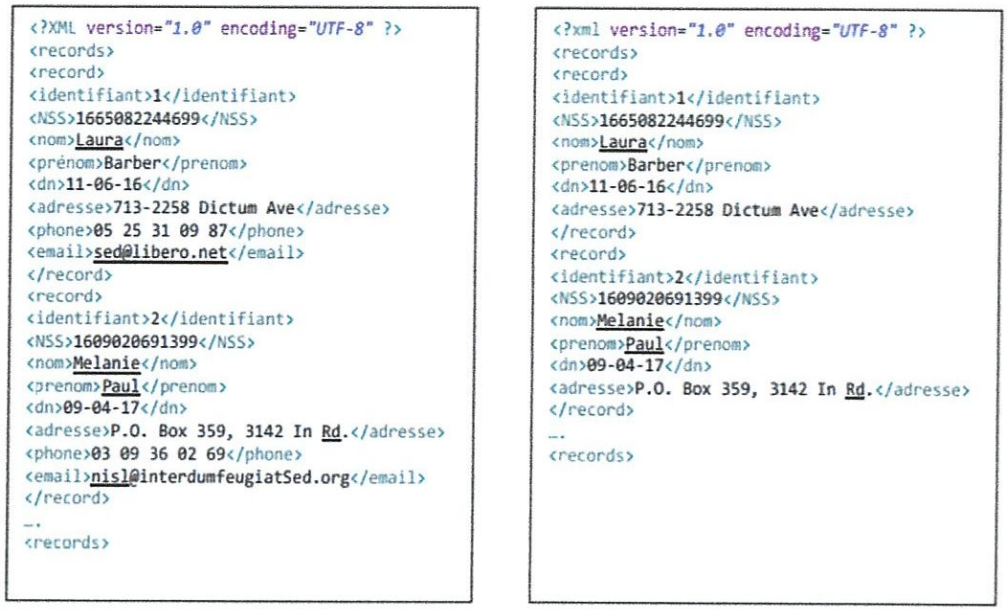


Figure 5.17: Extrait du document patients.xml avant et après l'exécution

5.8 Conclusion

A la fin de ce chapitre, nous estimons que nous somme arrivé à contribuer dans le domaine de contrôle d'accès aux données XML en proposant un modèles qui offre pas mal d'avantages comme le support des systèmes ouvert et fermés simultanément ainsi que la prise en charge de plusieurs stratégies de résolution de conflit. Notre modèle agit avec un niveau de granularité très fin au niveau élément. La gestion efficace des certificats rend notre modèle très robuste face à l'intrusion de type *man in the middle.*

Nous avons proposé une conception en UML très clair permettant une compréhension rapide de notre application et une implémentation qui mit en évidence les capacités du modèle proposé bien qu'elle ne s'agit pas d'une application distribuée. Notre modèle est toujours en phase d'expérimentation, nous somme en train e penser de l'élargir pour tenir en compte d'autres formalité de contrôle d'accès au document XML comme la propagation des politiques, et l'expression des ressource dans les règles avec XPath.

Conclusion

Durant ce mémoire nous avions l'occasion de traiter de pré un problème épineux en sécurité informatique s'agissant de fournir une solution qui permet à la fois de faire un contrôle d'accès rigoureux au données XML qui sont complexes par leur nature et de ne pas perturber le bon fonctionnement du logique métier du système protégé. Notre démarche pour réaliser cette solution était globale où nous n'avons pas seulement intéressé par les aspects purement techniques mais on a aussi donnée une importance à la nature juridique. En effet notre solution s'intègre dans une plateforme de gestion de la clé publique où les modalités d'application font l'objet d'un texte réglementaire paru récemment dans le journal officiel de l'état algérien. Du même, nous avons survolé abondamment tous les ingrédients nécessaires pour l'élaboration des systèmes de contrôles d'accès commençons par l'étude des éléments de base de sécurité informatique en l'occurrence : la confidentialité, l'intégrité l'authentification , la non répudiation et les éléments de la cryptographie et en passant par la suite par les grandes familles des systèmes contrôles d'accès MAC, DAC et RBAC en finissant par l'établissement d'un état de l'art des travaux réalisé dans le même sillage que le nôtre.

Notre modèle de contrôle d'accès aux données XML est un modèle hybride permettant l'expression des politiques de contrôle d'accès ouvertes (tout ce qui n'est pas explicitement interdit est autorisé)et fermés (tout ce qui n'est pas explicitement autorisé est interdit). Il peut prendre en charge les situations de contradiction en fournissant plusieurs stratégies de résolution de conflit (interdiction avant permission, permission avant interdiction, application des niveaux de priorité, ou l'application des premières règles ren-

contrées, etc.). Les données XML protégées ne se limitent pas aux documents intégrales mais peuvent être des fragments des documents avec un niveau de granularité qui descend jusqu'au niveau d'un élément XML. La particularité de notre modèle réside dans le concept de la liste d'application négative qui permet l'expression des exceptions lors de définition des règles, une chose fortement demandée pour les données de type XML.

Afin de montrer la force de notre modèle, nous avons implémenté une application client/serveur où le serveur utilise notre modèle pour appliquer sa politique de contrôle d'accès. Nous avons montré à travers un cas d'étude s'agissant d'une base de données médicale ses capacités d'expressivité. Les clients et les serveurs s'authentifient mutuellement en utilisent leurs certificats délivrés par une autorité de certification qu'on a implémenté aussi. Notre autorité de certification permet de générer des paires de clé avec des algorithmes variés (RSA, DSA, etc.) et de stocker et délivrer des certificats de type X509 selon la demande. A travers tout ça, nous estimons que notre travail est original au niveau dans notre établissement de plusieurs angles :

1. Le premier qui à considérer l'aspect juridique.

2. Le premier qui implémente une autorité de certification et manipulent des certificats pour l'achèvement d'une communication sure entre des interlocuteurs mutuellement authentifiés.

3. Le premier qui considère les données de type XML, souvent on traite les données des bases de données relationnelles.

Notre modèle est dans ses premières genèses, plusieurs lacunes sont à achever notamment dans la concrétisation des requêtes d'écritures et de modification où on n'a pas fait beaucoup de chose. Nous avons aussi besoin d'exprimer des conditions spatiotemporelles et sur le contenu de la ressource demandée. L'implémentation doit être dans un environnement distribué afin de montrer d'avantage l'efficacité de notre modèle. Tous ces points forment les perspectives du travail qui va embellir le nôtre.

Réferences

[1] Journal officiel, http://www.joradp.dz/

[2] http://www.garykessler.net/library/crypto.html :Consulté le Janvier 2017.

[3] Initiation à la cryptographie : théorie et pratique , Université Paris 13 Villetaneuse. Décembre 2016

[4] Loi N 03-05 du 19 Joumada El Oula 1424 correspondant au 19 juillet 2003

[5] Loi N 09-04 du 14 Chaâbane 1430 correspondant au 5 août 2009 portant règles particulières relatives à la prévention et à la lutte contre les infractions liées aux technologies de l'information et de la communication.

[6] Loi N 15-03 du 11 Rabie Ethani 1436 correspondant au 1er février 2015 relative à la modernisation de la justice

[7] Loi N 15-04 du 11 Rabie Ethani 1436 correspondant au 1er février 2015 fixant les règles générales relatives à la signature et à la certification électroniques

[8] La cryptologie moderne , Anne Canteaut Francoise Levy-dit-Vehel . Janvier 2017

[9] http://www.omnisecu.com/security/public-key-infrastructure/asymmetric-encryption-algorithms.php Consulté le 12/02/2017.

[10] Cryptographie et Sécurité informatique , Université de Liège (cours INFO0045-2).Janvier 2017

[11] https://www.securiteinfo.com/cryptographie/hash.shtml Avril 2017.

[12] https://www.securiteinfo.com/cryptographie/pki.shtml : Mars 2017.

[13] Securite des systemes d'exploitation repartis ; Mathieu Blanc

[14] Computer Security Subsystem Interpretation of the Trusted Computer System Evaluation Criteria.Janvier 2017

[15] A. Baraani, J. Pieprzyk, and R. Safavi-Naini. Security in databases: A survey study.

[16] B. W. Lampson. A note on the confinement problem. Communications of the ACM, 16(10):613–615.

[17] G. S. Graham and P. J. Denning. Protection - principles and practice. AFIPS Conference Proceedings Volume 40 Spring Joint Computer Conference, 40:417–429.

[18] B. W. Lampson. Protection. SIGOPS Oper. Syst. Rev., 8(1):18–24.

[19] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman. Protection in operating systems. Commun. ACM, 19(8):461–471.

[20] E. B. Fernandez, R. C. Summers, and C. D. Coleman. An authorization model for a shared data base. In SIGMOD '75: Proceedings of the ACM SIGMOD international conference on Management of data, pages 23–31, New York, NY, USA, ACM Press.

[21] R. W. Conway, W. L. Maxwell, and H. L. Morgan. On the implementation of security measures in information systems. Commun. ACM, 15(4):211–220.

[22] R.S. Sandhu and P.E. Ammann. Safety analysis for the extended schematic protection model. sp, 00:87.

[23] Sun Microsystems Inc. Java security architecture. Disponible sur http://java.sun.com/j2se/1.3/docs/guide/security/spec/securityspec.doc12.html.Mars 2017.

[24] R. S. Sandhu. Access control: The neglected frontier. In First Australian Conference on Information Security and Privacy, Wollong, Australia.

[25] D. Bell and L. LaPadula. Secure computer systems: Mathematical foundations and model. Technical report m74-244, Mitre Corporation, Belford, MA.

[26] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., Chandramouli, R.: Proposed NIST Standard for Role-Based Access Control. ACM Transactions on Information and System Security (TISSEC), 4(3), 224-274 (2001).

[27] O'Connor, A. C., Loomis, R. J.: Economic Analysis of Role-Based Access Control.NIST Report. (2010).

[28] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, 29(2):38–47, 1996.

[29] Erik Rissanen. eXtensible Access Control Markup Language (XACML) version 3.0 (committe specification 01). Technical report, OASIS, http://docs.oasisopen.org/xacml/3.0/xacml-3.0-core-spec-cd-03-en.pdf, Février 2017.

[30] R. Bhatti, E. Bertino, and A. Ghafoor. A trust-based context-aware access control model for web-services. Distrib. Parallel Databases, 18(1):83–105, 2016.

[31]] E. Bertino, A. C. Squicciarini, I. Paloscia, and L. Martino. Ws-ac: A fine grained access control system for web services. World Wide Web, 9(2):143– 171, 2016.

[32]] E. Bertino, M. Braun, S. Castano, E. Ferrari, and M. Mesiti. Author-X: A Java-Based System for XML Data Protection. In Proceedings of the IFIP TC11/ WG11.3 Fourteenth Annual Working Conference on Database Security, pages 15–26, Deventer, The Netherlands, The Netherlands, 2001. Kluwer, B.V.

[33] Z. Tari and R. Wonohoesodo. A role based access control for web services. scc, 00:49–56, 2004.

[34] Wonohoesodo, R., Tari, Z.: A Role based access control for Web services. In Proceedings of IEEE International Conference on Services Computing (SCC 2004), Shangai, China, September 2004, 49-56, IEEE Computer Society.

[35] World Wide Web Consortium (W3C), Extensible Markup Language (XML), http://www.w3.org/XML , Avril, 2017.

[36] World Wide Web Consortium (W3C), Document Type Definition (DTD), http://www.w3.org/XML/DTD , Avril, 2017.

[37] World Wide Web Consortium(W3C). XML path language (XPath). Disponible at. http://www.w3.org/TR/xpath, Avril 2017.

[38] XQuery: An XML Query Language W3C Recommendation Janvier 2017.

[39] Web Services Security X.509 Certificate Token Profile (OASIS, Janvier 2017).

[40] Oracle Java Documentation https://docs.oracle.com/javase/tutorial/jaxb/intro/ , Mai 2017.

[41] Oracle Java Documentation https://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html , Mai 2017.