République Algérienne démocratique et populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Saad Dahleb Blida

Faculté des Sciences

Département Informatique

Mémoire fin d'étude pour l'obtention du diplôme de Master en Informatique

Option : sécurité des systèmes d'information

Présenté par :

BERRICHE Hadjer

BRAHIMI Anfel

Thème

Etude et mise en œuvre d'une couche de sécurité pour un système de gestion du dossier médical électronique

Organisme d'accueil : Centre de Développement des Technologies Avancées (CDTA)

Soutenu le 23/09/2020 devant le jury composé de :

Présidente : M^me^ Aroussi Sana

Examinatrice : M^me^ Cherfa Imane

Promotrice : M^me^ Ghebghoub Yasmine

Encadreur : M^r^ Lakhneche Ramzi

Encadreur : M^r^ Oudjoudi Idir

Promotion : 2019/2020

Remerciement

Avant toute chose, on tient à remercier Allah, le tout puissant, pour nous avoir donné la force et la patience d'achever ce modeste travail, grâce à notre foi, nous croyons au destin, nous pouvons traverser les moments difficiles en regardant toujours le bon côté de la chose.

On exprime nos profonds remercient à Mme. GHEBGHOUB, d'avoir accepté de nous encadrer ; ses conseils, orientation et corrections précieuses ont été très profitables pour nous.

Nos plus sincère remerciements pour notre Promoteur M. OUDJOUDI Idir pour la confiance que nous avez accordée en nous proposant ce travail, Aussi pour notre Co-promoteur M. LAKHNECHE RAMZI qui nous a aidé et qui a mis toute sa compétence à notre disposition

Nous remercions également le chef de département ainsi tous le corps d'enseignants du département d'informatique et surtout les enseignants de la spécialités Sécurité des Systèmes d'Information qui ont contribué de près ou de loin à notre formation pendant les cinq ans qu'on a passé El Hamduli'Allah

Dédicace

Je dédie ce modeste travail aux êtres qui me sont les plus chers, je cite:

Mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

À mes chers frères Oussama et Abd El Khader ainsi qu'à ma sœur Naila, en reconnaissance de leur affection toujours constante

À mon cher mari Amine,

Mon neveu Nazim et ma nièce Dounia,

À ma binôme Hadjer,

Mes amies particulièrement Bouchra, Amina, Zizou, Chahinez Et Soumia.

Anfel

Dédicaces

Je dédie ce travail à

Mon cher père Qu'Allah lui accueille dans son vaste paradis

« J'étais toujours fière de toi et je resterai à jamais, J'avais l'impression que quand je vais arriver à ce stade tu seras fière de moi aussi »

Ma mère, Pour son affection, sa patience, son encouragement et le soutien pendant les épreuves difficiles ainsi que ses prières qui m'apportent le bonheur et la réussite.

Mes frères Amine et Chakib

Mes sœurs Meriem et Asma et leurs enfants Islem, Razane et Alya

« Qu'Allah Nous protège et nous garde solide »

Mes cousines Hadjer, Asma, Loubna et Aya.

Mon cher cousin Nadjib dont nous avons le perdu dans cette période paix à son âme

Hadjer

Résumé

Le Dossier Médical électronique (DME) comprend toutes les informations relatives à la santé d'un malade. Ce dossier peut être stocké, recherché et manipulé lors des consultations des patients. Les données du patient seront également partagées entre les professionnels médicaux et les établissements de santé, ce qui peut provoquer la réticence des patients ou empêcher la coordination et la continuité des soins donc la protection du DME doit être garantie, afin de garantir le respect des droits des patient à la vie privée et la déontologie médicale.

La gestion des données du patient suppose l'existence d'une couche de sécurité qui peut être composée de plusieurs techniques qui ont été proposées dont nous pouvons citer : le contrôle d'accès, la sécurité par chiffrement de l'information (la cryptographie), la sécurité logique, la sécurité physique, la sécurité administrative, la sécurité des systèmes d'exploitation, la sécurité des communications, etc.

L'objectif de ce travail de recherche est centré sur les modèles de contrôle d'accès dans les systèmes d'information en santé et les méthodes de chiffrement des données. Il s'agit donc de proposer une modélisation rigoureuse permettant de prendre en charge tous les aspects liés à la gestion sécurisée du DME.

Dans un premier temps, nous avons développé un modèle de contrôle d'accès à ce dossier en se basant sur le modèle OrBAC avec un nombre très réduit de règles d'accès et le rajouter une technique de chiffrement de données par attributs CP-ABE.
Les résultats de notre solution valide l'efficacité de la couche de sécurité proposée pour la gérer en toute sécurité les données médicales des patients et assurer la confidentialité des utilisateurs

Mots-clés : Dossier Médical électronique, Sécurité, Contrôle d'Accès, OrBAC, Cryptographie, Chiffrement CP-ABE.

Abstract

The Electronic Medical Record (EMR) includes all information relating to the health of a patient, this record can be stored, searched and manipulated during patient consultations.
Patient data will also be shared between medical professionals and healthcare establishments .
What may cause patient reluctance or prevent coordination and continuity of care, therefore, the protection of the EMR must be guaranteed, in order to guarantee respect for patients' rights to privacy and medical ethics.

The management of patient data supposes the existence of a layer of security, which can be composed of several techniques, which have been proposed of which we can mention:
Access control, security by encryption of information (cryptography), logical security, physical security, administrative security, security of operating systems, Security of communications, etc.

The objective of this research work is centered on access control models in health information systems and data encryption methods.
It is therefore to propose a rigorous modeling to support all aspects related to the secure management of the EMR.

As a first step, we developed an access control model for this file based on the OrBAC model with a very small number of access rules and add a data encryption technique by CP-ABE attributes. The results of our solution validate the effectiveness of the proposed security layer to securely manage patient medical data and ensure user privacy.

Keywords: Electronic Medical Record, Security, Access Control, OrBAC, Cryptography, CP-ABE Encryption.

ملخص

يتضمن السجل الطبي الإلكتروني (DME) جميع المعلومات المتعلقة بصحة المريض، ويمكن تخزين هذا الملف، والبحث فيه والتلاعب به أثناء استشارة المريض. كما سيتم تبادل بيانات المرضى بين المهنيين الطبيين ومرافق الرعاية الصحية. ولذلك، يجب ضمان ما يمكن أن يسبب إحجام المريض أو منع التنسيق واستمرارية الرعاية، من أجل ضمان احترام حقوق خصوصية المرضى وأخلاقيات الطب.

تفترض إدارة بيانات المريض وجود طبقة أمان يمكن أن تتكون من عدة تقنيات تم اقتراحها، والتي يمكن أن نذكر منها: التحكم في الوصول، والأمن عن طريق تشفير المعلومات (التشفير)، والأمن المنطقي، والأمن المادي، والأمن الإداري، وأمن أنظمة التشغيل، وأمن الاتصالات، إلخ.

يتركز الهدف من هذا العمل البحثي على نماذج التحكم في الوصول في أنظمة المعلومات الصحية وطرق تشفير البيانات، وبالتالي اقتراح نمذجة صارمة للعناية بجميع الجوانب المتعلقة بالإدارة الآمنة لDME .

كخطوة أولى، قمنا بتطوير نموذج التحكم في الوصول لهذا الملف بناءً على نموذج OrBAC مع عدد قليل جدًا من قواعد الوصول وإضافة تقنية تشفير البيانات بواسطة سمات .CP-ABE تتحقق نتائج الحل الذي نقدمه من فعالية طبقة الأمان المقترحة لإدارة البيانات الطبية للمريض بشكل آمن وضمان خصوصية المستخدم.

الكلمات المفتاحية :السجل الطبي الإلكتروني، الأمن، التحكم في الوصول، OrBAC، التشفير، تشفيرCP-ABE .

Tables des matières

Liste de Figures

Liste des Tableaux

Liste des acronymes

ABE	Attribute Based Encryption
ACL	Access Control List
CBC	Chipher Block Chaining
CP-ABE	Ciphertext-Policy Attribute Based Encryption
DAC	contrôle d’accès discrétionnaire
DEM	dossier électronique du malade
DEP	dossier électronique du patient
DES	Data encryption standard
DME	Dossier Médicale Electronique
DMI	le dossier malade informatisé
DMP	Dossier Médical Personnel
DPE	Dossier Patient Electronique
DPI	le dossier du patient informatisé
DSA	Digital signature Algorithm
ECB	Electronic Code Book
FIPS	Fideral Information Processing standard
HRU	Harison Ruzzo et Ullman
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol

KP-ABE	Key-Policy Attribute Based Encryption
MAC	contrôle d’accès mandataire
Mk	Principal Key
MVC	Model, View, Controller
O.M.S	Organisation Mondiale de la Santé
OrBAC	Organization-based Access Control
ORM	Object-Relational-Mapping
PK	Public Key
RBAC	Role-Based Acces Control
RC4	Rivest Cipher 4
RSA	Rivest - Shamir – Adleman
SGDB	Système de gestion de base de données
SGDP	systèmes de gestion des dossiers patients électroniques
SI	Système d'information
SIH	Système d'information Hospitalier
SK	Secret Key
SSL	Secure Sockets Layer
UML	Unified Modeling Language
URL	Uniform Resource Locator
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
XOR	eXclusive OR

Introduction Générale

1. Contexte et motivation

A l'heure actuelle, l'informatisation s'impose dans des domaines complexes, coopératifs et largement distribués. Il est de plus en plus nécessaire d'avoir confiance dans les traitements et la distribution des données et services informatiques.

Alors que le développement des systèmes d'information hospitalier par le monde connait une effervescence accrue, au vu de l'importance que revêtent dans l'amélioration de stockage et de la gestion des informations médicales relatives à des personnes. Ils permettent aux utilisateurs un accès rapide à ces informations.

Face à ce constat, il apparait certain que le secteur de la santé en Algérie doit se doter d'un système d'information moderne construit sur la base d'un socle de base pour le dossier médical du patient. Le développement de ce socle de base doit se construire dans le respect des normes et des standards internationaux.

2. Problématique

Toutefois les données à caractère médical ont une nature hautement sensible, et sujettes au secret médical. Ces dernières sont toutes les données médicales relatives à la santé d'une personne, d'un groupe de personnes ou de populations.

En effet, les menaces qui pèsent sur les systèmes d'information e-santé peuvent provoquer la réticence des patients ; les erreurs de saisie ou de conception peuvent entrainer des erreurs de diagnostic de soins. Les défaillances peuvent empêcher le personnel soignant d'accéder à des informations indispensables, la peur d'un manque de confidentialité, d'intégrité, de disponibilité ou d'audibilité de tels systèmes peut inciter des patients à refuser de divulguer des informations pourtant vitales. C'est pourquoi la partie sécurité dans le développement d'un système e-santé détient une place importante.

La sécurité des données des patients est un processus complexe qui affecte l'ensemble des acteurs intervenants dans le circuit de prise en charge du patient.

La protection des données médicales doit être garantie, afin d'assurer le respect des droits des patients à la vie privée et la déontologie médicale.

3. Objectives et solution proposée

L'objectif principal du projet est le développement d'une couche de sécurité pour un système de gestion du dossier médical du patient prenant en charge des règles de sécurité des données en général et tenant compte des règles de la déontologie médicale.
Pour atteindre un niveau de protection satisfaisant, il convient de définir une politique de sécurité correspondant aux besoins.

La couche de sécurité doit aborder les éléments suivants :

- Sécurité de la donnée (donnée médical sensible).
- Sécurité des transferts et partage,
- Gestion des accès des professionnels de santé.

Nous allons au cœur de ce travail essayer de développer un ensemble de module informatique formant la couche de sécurité des données médicales. Ces modèles représentent une solution qui consiste à implémenter une approche de contrôle d'accès OrBAC basé sur une technique de chiffrement de donnée CP-ABE (Ciphertext-Policy Attribute-Based Encryption) afin d'assurer la gestion d'accès des professionnels de santé et sécuriser les données et leurs transferts.

4. Organisation du mémoire

Ce mémoire est organisé comme ceci :

Chapitre 1 : Dans ce chapitre nous exposons des généralités sur le dossier médical électronique en donnant quelques définitions, la structure, les menaces, notamment la sécurité des systèmes d'information inclus la sécurité de DME.

Chapitre 2 : Nous présentons les différents modèles de contrôle d'accès et nous détaillons le contrôle d'accès basé sur l'organisation OrBAC dont nous avons choisi.

Chapitre3 : Nous présentons les méthodes de la cryptographie en citent quelques algorithmes de chiffrement notamment le chiffrement par attribut ABE.

Chapite4 : montre la conception et la modélisation de notre solution.

Chapitre5 : ce chapitre représente la réalisation et la mise en œuvre de la couche de sécurité

Et enfin dans la conclusion générale, on résume les limites et les perspectives futures de notre travail.

Chapitre 1 Introduction au Dossier Patient Electronique (DPE)

1. Introduction

Un système d'information (SI) est un système qui permet d'acquérir, de stocker, de traiter et de communiquer les informations circulant dans un établissement (où le dit système est installé).

Le Dossier médicale électronique représente un système d'information dans le domaine e-santé qui est un ensemble qui fait partie des systèmes d'information hospitaliers.

2. Le Système d'information hospitalier 'SIH'

Le Système d'Information Hospitalier est une des composantes du Système d'Information de santé. Il est appliqué aux établissements de santé (hôpitaux, cliniques, cabinets de santé, etc.). Il gère toutes les informations administratives et médicales du centre hospitalier [1].

Le SIH est capable, selon des règles et modes opératoires prédéfinis, d'acquérir des données, de les évaluer, de les traiter par des outils informatiques ou organisationnels, de distribuer des informations contenant une forte valeur ajoutée à tous les partenaires internes ou externes de l'établissement.

Le terme « SIH » renvoie explicitement au Système d'Information interne à une organisation de santé et plus précisément aux hôpitaux. Les établissements visés sont typiquement [2] :

- Les hôpitaux ou structures publique ;
- Les cliniques ou structures privées.

Bien que dotées de Systèmes d'Information, le terme SIH ne sera pas approprié pour les autres organisations de santé telles que [2] :

- Les centres de radiologie ;
- Les laboratoires d'analyses de biologie médicale ;
- Les centres de soins ;
- Les cabinets médicaux.

Le SIH améliore ainsi l'efficience des organisations sanitaires. Son bénéfice se fait également ressentir chez les patients qui, grâce à ce nouveau traitement de l'information, voit leur prise en charge s'améliorer. Les professionnels de santé disposant en effet, de toutes les données nécessaires en temps réel pour traiter la maladie (antécédents, examens réalisés…).

D'un point de vue fonctionnel, il permet à l'hôpital de gérer ses ressources, évaluer et planifier et gérer l'information. Pour le patient, il permet la gestion médico-administrative, de réaliser les actions médicales et gérer le dossier patient [3].

Le SIH est composé essentiellement de trois systèmes : Le système administratif, logistique, le système médical [4].

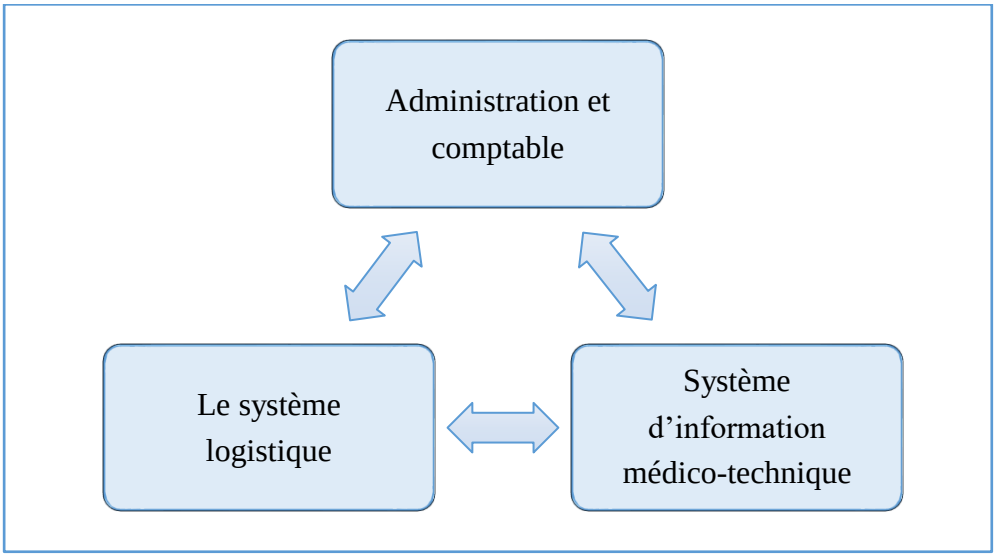


Figure 1: Les composants d'un SIH [4]

3. Dossier Médicale Électronique 'DME'

3.1 Définition

Le dossier médical électronique c'est un sous-ensemble des systèmes d'informations hospitaliers (SIH).

Les professionnelles de santé autorisées par le patient et en respectant la déontologie médicale peuvent consulter ou ajouter des informations utiles à la prise en charge du patient [5].

En effet ; le dossier électronique du patient « DEP », le dossier malade informatisé « DMI », le dossier électronique du malade « DEM », le dossier du patient informatisé « DPI » sont tous identiques et désignent effectivement le même objet [6].

Ce dossier médical permet une représentation du parcours de soins du patient. Il peut être structuré en plusieurs espace par exemple [5] :

- Un espace de synthèse et données médicales générales ;
- Un espace traitements et soins ;
- Un espace compte rendus ;
- Un espace imagerie médicale ;
- Un espace analyses de laboratoires ;
- Un espace prévention ;
- Un espace certificats et déclarations.

Le Dossier Patient assure la traçabilité de toutes les actions effectuées par les Professionnels de Santé. Il est un outil de communication, de coordination et d'information entre les Professionnels et avec les patients. Il permet de suivre et de comprndre le parcours hospitalier du patient [2].

3.2 La structure de DME

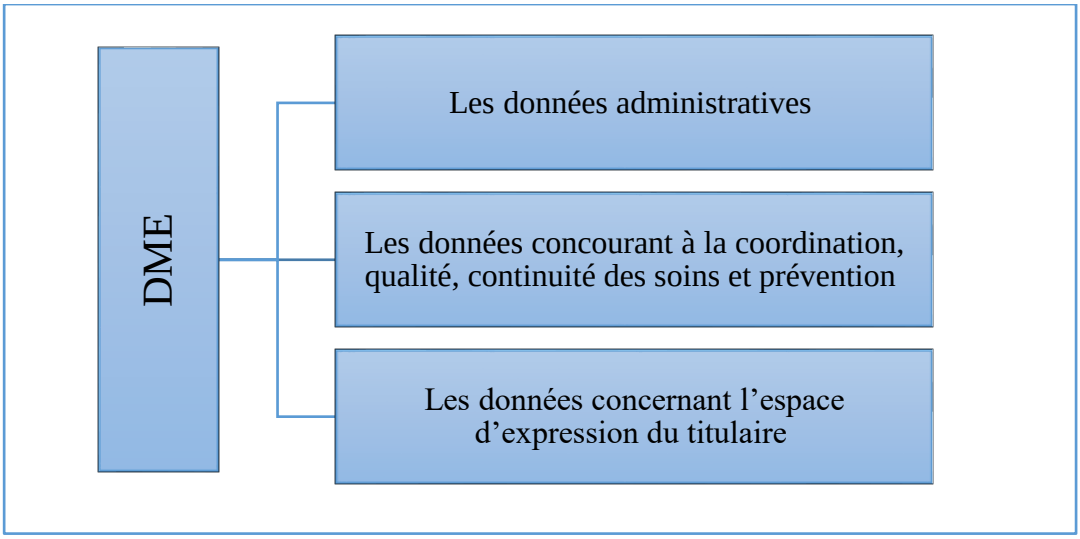


Figure 2: Structure de DME [7]

Le DMP se compose en trois parties [7] :

a. Les données administratives

Elles regroupent l'identification du titulaire (nom, prénom, la date de naissance), le médecin traitant et les informations techniques qui permettent de s'assurer de l'identification de la personne et de la contacter.

b. Les données concourant à la coordination, qualité, continuité des soins et prévention

Elles regroupent les données médicales générales (antécédents, synthèses, historiques des consultations, allergies et intolérances reconnues, prothèses et appareillage), les données de soins (résultats d'examens biologiques, bilans, pathologies en cours, traitements prescrits et administrés), les données de prévention (facteurs de risques individuels, traitements préventifs prescrits, calendrier des vaccinations) et les images radiologiques ou autre imagerie.

c. Les données concernant l'espace d'expression du titulaire

Elles regroupent le don d'organes et les coordonnées d'une personne à prévenir en cas de nécessité pour l'accès au dossier médical (Personne de confiance).

3.3 Intérêt du DME

Le dossier patient est un outil à usages multiples [8] :

- **Outil de suivi du patient**

  C'est son usage principal et c'est dans le dossier que les demandes d'examens et leurs résultats sont colligés et que le médecin exprime ses réflexions, ses interrogations et ses conclusions.

- **Outil de communication**

  Le travail médical est de plus en plus un travail d'équipe (cabinets de groupe, centres de santé, hôpital). Les informations pertinentes doivent être disponibles

à tous les professionnels qui ont traité le malade. Le dossier est un des meilleurs moyens d'assurer la communication de ces informations.

- **Outil de gestion hospitalier**
  Connaître les diagnostics, les actes thérapeutiques, le coût entrainé par la population de malades qui fréquente un service, un département ou un hôpital est indispensable à celui qui a la responsabilité de gérer ces structures.

- **Outil d'étude épidémiologique**
  Dossiers patients hospitaliers peuvent donner des aperçus intéressants sur la santé de la population.

C'est aussi un outil pour [6] :

- L'amélioration de la qualité des soins (faciliter la recherche d'information, la prise de décisions médicales et la communication et la coopération entre professionnel de santé.
- Permettre le regroupement de données afin de faciliter l'évaluation, la recherche et la planification.
- La traçabilité et la précision sur le plan des données, la sécurisation des traitements, en plus d'économiser le papier et d'épargner des inconvénients.

3.4 Les bénéfices de DME par rapport au dossier papier

Le tableau suivant illustre parfaitement la différence des caractéristiques fonctionnelles et leurs impacts dans le dossier traditionnel et informatisé.

L'intervalle [0, +++] représente le degré de l'impact des caractéristiques dans les deux dossiers :

'0' : aucun impact, '+' : impact faible et '+++' : impact élevé.

Caractéristique fonctionnelle	Type de dossier	
	Traditionnel	**Informatisé**
Stockage et communication des informations		
- intégration des données (+multimédia)	+	+++
- lisibilité du dossier	+	++
- prise en charge ensemble des problèmes	+	++
- complétude	+	+++
- accès	Séquentiel	Simultané
- disponibilité	Local	Globale
- accès à distance	0	+++
- chaînage d'épisodes de soins	+	+++
- chaînage de dossiers distribués	0	++
Regroupement des données		
- évaluation des soins	+	+++
- recherche clinique, épidémiologique	+	+++
- contrôle de gestion, planification	0	+++
Formation, éducation		
- facilité d'utilisation du dossier	+++	+
- formalisation de la démarche de soins	+	+++
- adhésion aux protocoles de soins	+	+++
- connexion à des banques de données documentaires ou de connaissances	0	+++
Sécurité, protection		
- sécurité de l'information	+	+++
- confidentialité	++	+

Tableau 1: Bénéfices de l'informatisation du dossier patient [4]

3.5 Caractéristiques d'un bon dossier patient

Les qualités d'un bon dossier patient sont définies dans une notice de l'O.M.S (Organisation Mondiale de la Santé) comme étant les suivantes [8] :

- Identifier sans ambiguïté la personne qu'il concerne,
- Être lisible et pouvoir être compris par tous ceux qui seront amenés à l'utiliser,
- Être précis, concis et logique,
- Être cohérent quant à la disposition et au format des documents qui le constituent,
- Désigner les personnes qui sont amenées à y porter des écritures afin qu'on puisse, le cas échéant, leur demander un complément d'informations,
- Pouvoir être rapidement retrouvé quand on en a besoin.

Ainsi définies, toute conception ou préconception d'un dossier patient doit être guidée par ces caractéristiques.

4. Sécuriser le Dossier Médical Electronique

4.1 C'est quoi la sécurité

« La sécurité est l'ensemble des mesures permettant d'assurer la protection des biens / valeurs » [6]

Dans le monde informatique on distingue deux types de biens à savoir :

- **L'information, les données :** comme Les contacts, les données financières, les données stratégiques, les données patients etc…
- **Les systèmes permettant de traiter, véhiculer et stocké l'information** comme : Les applications, les serveurs etc…

4.2 Les composants de la sécurité informatique

La sécurité est essentielle pour la protection de trois caractéristiques critiques des systèmes et de l'information dont ils traitent et maintiennent, à savoir [9] :

1. **Confidentialité** : Assurer que l'information soit protégée contre toutes divulgation accidentelle ou malveillante aux parties non autorisé
2. **Intégrité** : assure que l'information et les systèmes soient protégés contre tout modification ou destruction accidentelle ou malveillante
3. **Disponibilité** : assure que l'information et les systèmes soient accessibles et utilisables par les parties autorisées au moment où elles en ont besoin

A côté de ces caractéristiques de bases nous rencontrons également les composantes suivantes [9] :

- **Authentification** : assure l'identification d'un individu, d'une entité mais également l'origine de l'information an encore d'une opération effectuée sur celle-ci.
- **Autorisation** : assure le contrôle du type d'activités ou d'informations qu'une personne ou entité est autorisé à effectuer ou accéder
- **Non-répudiation ou irrévocabilité** : assure le fait qu'une personne ou entité ne puisse nier avoir effectuer une activité. Dans le domaine du courriel. L'irrévocabilité est utilisée pour assurer que le destinataire ne pourra nier avoir reçu l'information, et assurer que l'expéditeur de la source de l'information ne peut nier avoir envoyé l'information.
- **Journalisation** : assure que tout accès à un système. Tout accès à une information ainsi que toute opération exercée sur ceux-ci soient journalisés / répertoriée.

4.3 La sécurisation du DME

Le DMP (Dossier Médical Personnel) pose de manière forte le problème de la sécurité et de la protection des données personnelles de santé. Censé être un facilitateur de contact entre les médecins, les professionnels de santé et les patients, et il peine encore à se faire un chemin dans le monde médical, en délivrant une information fiable et sécurisée attachée au patient [10].

La sécurité du dossier médicale informatisé repose sur [10] :

- **La mise en œuvre de contrôles à priori**
  Tous les utilisateurs sont authentifiés de manière forte pour accéder au dossier médical, le contrôle d'accès aux informations qui est appliqué laisse la possibilité à un professionnel de santé d'accéder sous son entière responsabilité aux données de santé des patients qu'il prend en charge dans la limite de l'autorisation d'accès donnée par le patient et des documents autorisés pour sa profession.

- **Le contrôle à posteriori des actions des utilisateurs :**
  Ce contrôle est fondé sur une traçabilité et une responsabilité totale des actions effectuées par l'ensemble des utilisateurs, et toute personne a une mauvaise utilisation sera pénalisée. L'entrée en service du dossier médical informatisé favorise le partage des données de santé, et entraîne une évolution significative de la nature des risques relatifs à la sécurité de l'information de point de vue d'évolution des menaces et des vulnérabilités potentielles portant sur les données. Les établissements de santé doivent donc toujours protéger les données personnelles de santé de leurs patients au sein de leur Système d'Information Hospitalier. Les gestionnaires des dossiers médicaux contrôlent l'accès aux dossiers des patients et préserve la confidentialité et l'intégrité des données personnelles contenues dans ces dossiers. Il trace les accès et enregistre toutes les actions d'un patient sur son Dossier Médical. Toutefois, lorsqu'un utilisateur accède à des données dans un système en ligne, ces données sont aussi

temporairement présentes dans la machine utilisée. Si cette machine n'est pas protégée, il peut faire l'objet d'une attaque et héberger un code malveillant capable d'exploiter ces données. Dans ce cas l'accès au système à partir d'un terminal protégé contre les attaques Internet et les codes malveillants est une précaution essentielle de la sécurité.

5. Conclusion

Dans ce chapitre, nous avons présenté la structure du Dossier médicale électronique pour les organisations de santé algériennes.

La sécurisation du DME constitue un enjeu essentiel pour assurer un climat de confiance qui encourage le partage des données médicales, d'où le problème de comment peut-on partager et gérer le DME tout en respectant la vie privée du patient.

Pour ce faire, plusieurs modèles de contrôle d'accès ont été proposés. Ces modèles doivent imposer ce qui est permit, ce qui est interdit et ce qui est obligé. Dans le chapitre suivant, nous allons décrire ces modèles pour passer ensuite au choix d'un modèle approprié supportant la structure et le contenu du DME.

Chapitre 2 Politiques et Contrôles d'Accès

1. Introduction

Nous avons lors du chapitre précédent introduit le DME, sa structure et ses avantages ainsi que le problème de sécurité et la protection des données qui doivent être traités.

La sécurisation des systèmes d'information est un aspect très important qui est devenue un enjeu majeur pour les différents systèmes ainsi que pour l'ensemble des acteurs qui l'entourent, dont l'objectif est la protection des ressources informatiques contre l'utilisation non-autorisée, le mauvais usage, la divulgation et la modification, tout en garantissant l'accès pour les utilisateurs légitimes.

Pour assurer cette sécurité, plusieurs techniques ont été proposées dont nous présentons dans ce chapitre.

Le but de ce chapitre est de dresser un état de l'art en ce qui concerne les modèles de contrôle d'accès et politiques de sécurité. Nous discuterons les limites et les avantages des différents modèles, afin de déterminer quel modèle pourrait convenir à nos exigences.

2. Politique de sécurité

« La politique de sécurité d'un système spécifie l'ensemble des lois, règlements et pratiques qui régissent la façon de gérer, protéger et diffuser les informations et autres ressources sensibles au sein d'un système spécifique » [11].

Un modèle de sécurité est défini comme un formalisme permettant de représenter la politique de sécurité, de l'abstraire, d'en réduire la complexité, et d'aider à en vérifier la complétude (c'est-à-dire que les propriétés satisfont toutes les exigences), la cohérence (c'est-à-dire que les règles sont suffisantes pour satisfaire les objectifs) et la conformité (c'est-à-dire que les mécanismes mis en œuvre implémentent les règles).

La plupart des politiques de sécurité sont des modèles de contrôle d'accès [12].

3. Les modèles de contrôle d'accès

Le modèle de contrôle d'accès décrit uniquement les entités régies par la politique et énonce les règles qui constituent la politique.

Plusieurs modèles sont été proposés pour encoder les politiques de contrôle d'accès, ils peuvent être classés en trois catégories principales :

3.1 Modèles de contrôle d'accès classiques

Les modèles de contrôle d'accès discrétionnaire (DAC) et le modèle de contrôle d'accès mandataire (MAC) se sont rapidement imposés pour faciliter la gestion des accès aux ressources au sein des systèmes d'exploitation et des systèmes de gestion de bases de données. Ces deux modèles reposent sur l'exploitation du triplet sujet (S), objet (O), Action (A)) pour représenter les politiques de contrôle d'accès [14].

3.1.1 Modèle de contrôle d'accès discrétionnaire(DAC)

Le *Discretionary Access Control* est basé sur les notions de sujets, objets et droits d'accès. Chaque ressource (objet) du système a un propriétaire (un sujet), lequel peut déterminer les privilèges (droits) d'accès à cet objet.

Le sujet a un contrôle complet sur tous les objets qui lui appartiennent, il peut changer les permissions d'accès, transférer des objets authentifiés ou des accès à l'information à d'autres sujets. C'est pourquoi il est dit discrétionnaire.

Dans ce modèle, les autorisations sont attribuées directement à des sujets en fonction de leur identité ; l'inconvénient d'une telle approche est que, dans les grands systèmes, déterminer l'octroi de l'autorisation sur une ressource donnée à des utilisateurs individuels, est laborieux et difficile à gérer.

La révocation de la permission est également complexe lorsque l'utilisateur quitte l'entreprise ou change de fonction, par exemple. L'information peut être copiée d'un objet à un autre, de sorte que l'accès à une copie est possible même si le propriétaire initial ne donne pas accès à l'originale.

Puisque les politiques du DAC peuvent être facilement modifiées par le propriétaire, un programme malveillant s'exécutant en son nom pourra aussi changer ces mêmes politiques, ce qui constitue une faiblesse [13].

L'implantation de ce modèle a donné lieu à la constitution de matrices d'accès initialement introduite en 1971 par Lampson [Lampson 1971] qui a été généralisée en 1976 par Harison, Ruzzo et Ullman (HRU) [Harrison et al. 1976]. Dans ce dernier, l'état du système est défini par un triplé (*S, O, M*) où *S* représente l'ensemble des sujets (e.g. utilisateur, processus etc.) pouvant exercer un ensemble d'actions. *O* représente l'ensemble des objets (e.g. fichier, table, classe, programme etc.). Enfin, *M* représente la matrice d'accès, où les lignes correspondent aux sujets et les colonnes correspondent aux objets [14].

Le tableau suivant présente un exemple d'une matrice d'accès :

Objet / Sujet	Fichier	Table
Alice	Lire Ecrire Exécuter	Exécuter
Bob	Lire Ecrire	Exécuter Lire

Tableau 2: Exemple d'une matrice d'accès [14]

Il existe en pratique deux approches pour implémenter la matrice d'accès [14] :

- Par une liste de contrôle d'accès (ou ACL pour Access Control List) : la matrice est stockée par colonne. A chaque objet est associée une liste de règles indiquant pour chaque utilisateur les actions pouvant être exercées par ce dernier sur cet objet.

- Par une liste de capacité (ou capability) : la matrice est stockée par ligne. A chaque utilisateur correspond une liste, appelée liste de capacité, indiquant pour chaque objet les actions que l'utilisateur est en droit d'effectuer sur cet objet.

3.1.2 Modèle de contrôle d'accès obligatoire (MAC)

« Un contrôle d'accès est dit obligatoire lorsque l'accès aux objets est basé sur le niveau de sensibilité de l'information contenue dans les objets. L'autorisation d'accéder à un objet est accordée à un sujet si le niveau d'autorisation de celui-ci est en accord avec le niveau de sensibilité de l'information » [15].

Seul l'administrateur de sécurité peut gérer les autorisations. C'est lui qui définit la politique d'utilisation et d'accès ; Les utilisateurs ne pourront pas déterminer qui peut accéder à leurs fichiers.

Dans ce modèle, toutes les informations sont affectées à un niveau de sécurité, et chaque utilisateur est affecté à une habilitation de sécurité.

Le sujets et l'objets possèdent des habilitations et des étiquettes, respectivement, comme par exemple « confidentiel », « secret », et « très secret ». Il garantit que tous les utilisateurs

n'ont accès qu'aux données pour lesquelles ils possèdent une habilitation égale ou supérieure à l'étiquette de l'objet.

Pour éviter les fuites d'information, l'accès aux objets doit obligatoirement respecter les deux principes fondamentaux :

- No read up : un sujet est autorisé à lire un objet donné uniquement si sa classe d'accès domine la classe d'accès de l'objet.
- No write down : un sujet est autorisé à écrire dans un objet donné uniquement si la classe d'accès de l'objet domine sa classe d'accès [14].

MAC préserve la confidentialité et l'intégrité des informations, empêche certains types d'attaques comme Trojan Horse et prévient l'altération non autorisée des objets. Mais ses inconvénients majeurs sont le manque de flexibilité et la difficulté à mettre en œuvre et à programmer ce modèle [13].

Les modèles de contrôle d'accès classiques définissent une relation directe entre les sujets et les objets. Ces modèles classiques sont développés pour résoudre des problèmes de sécurité traditionnels comme la confidentialité et l'intégrité. Mais, ils ont trouvé leurs limites : trop rigides, insuffisamment sûrs ou difficiles d'administration. A l'usage, une limite importante de ces modèles est apparue : la politique d'autorisation devient rapidement complexe à exprimer et administrer.
Pour cela, d'autres modèles qui brisent cette relation directe entre sujet/objet en y insérant de nouveaux concepts : les tâches, les rôles etc. ont été proposés par la suite [14].

3.2 Modèle de contrôle d'accès à base de rôle (RBAC)

Role-Based Acces Control a été proposé pour fournir un modèle et des outils qui permettent de gérer le contrôle d'accès dans un système complexe avec un très grand nombre d'utilisateurs et d'objets [16].
C'est un modèle de contrôle d'accès basé sur les rôles. Un rôle représente de façon abstraite une fonction identifiée dans l'organisation (par exemple, médecin, infirmier, etc.). À chaque rôle, on associe des permissions (représentées par des droits d'accès). Une permission est un ensemble de droits correspondant aux tâches qui peuvent être effectuées par un rôle.

RBAC définit quels utilisateurs ont accès aux ressources en fonction des rôles qui leur sont assignés, et l'accès aux ressources est limité aux utilisateurs auxquels on a attribué un rôle qui permet d'accéder à ces ressources. Chaque utilisateur se voit attribuer un ou plusieurs rôles et une ou plusieurs permissions sont attribuées à chaque rôle. Ainsi, dans RBAC, la politique de contrôle d'accès ne s'applique pas directement aux utilisateurs comme dans les précédents modèles de contrôle d'accès (DAC, ou MAC), les permissions ne sont plus associées de manière directe aux sujets, mais par le biais de rôles, qui regroupent des sujets qui remplissent les mêmes fonctions.

L'un des avantages de RBAC est qu'il n'est pas nécessaire de mettre à jour l'ensemble de la politique de contrôle d'accès si un nouveau sujet est créé, il suffit juste d'assigner un rôle à ce sujet. Les politiques basées sur les rôles visent donc à faciliter l'administration de la sécurité [12].

Cependant dans le modèle RBAC il n'est pas possible de spécifier une permission qui dépend d'un certain contexte car si une permission particulière est donnée à un certain rôle, tous les utilisateurs qui possèdent ce rôle hériteront de cette permission. Un autre inconvénient de RBAC est que seul l'administrateur est capable de spécifier des permissions [13].

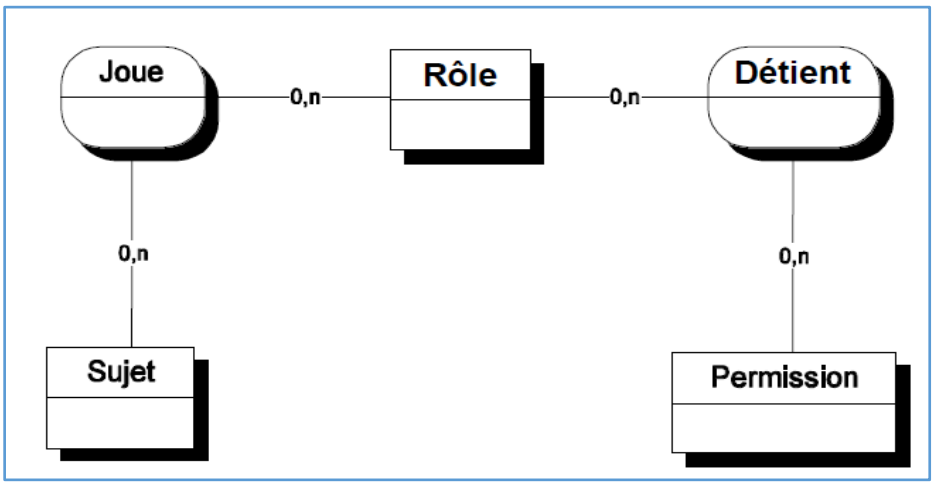


Figure 3: Le modèle RBAC [17]

3.3 Modèle de contrôle d'accès à base d'organisation (ORBAC)

Le modèle OrBAC (*Organization-based Access Control*) est basé sur les mêmes principes que son prédécesseur RBAC, en intégrant de nouvelles notions.

L'idée principale est d'exprimer la politique de sécurité avec des entités abstraites et de séparer complètement la représentation de la politique de sécurité de son implémentation. Le modèle OrBAC, est centré sur le concept d'organisation (une organisation est un groupe structuré d'entités actives), et tous les autres concepts d'OrBAC sont définis par rapport à l'organisation [12].

Par exemple dans le domaine médical, l'organisation pourra être l'hôpital, une clinique ou bien des services dans un hôpital tel que le service des urgences, le service de chirurgie ou le service de radiologie.

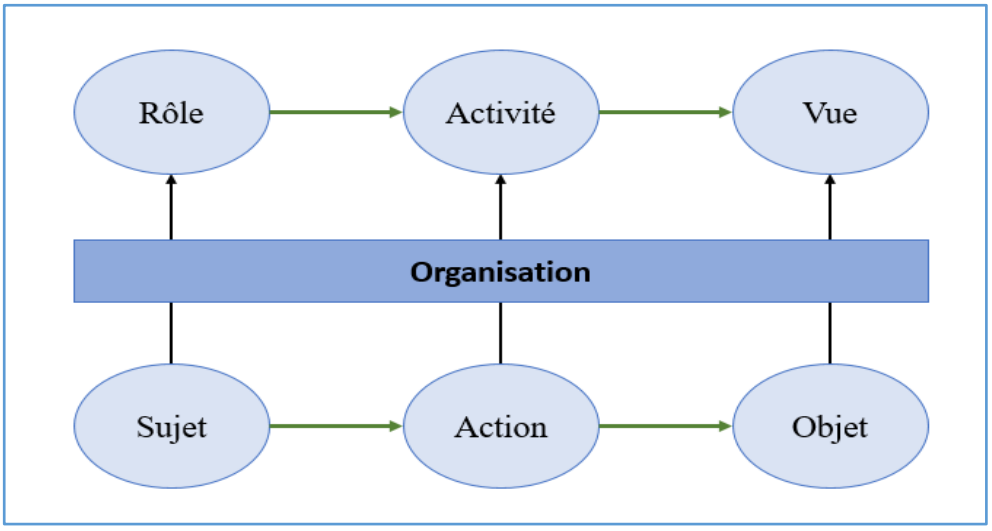


Figure 4: Le modèle ORBAC [12]

On distingue deux niveaux dans OrBAC [12] :

- Niveau abstrait dans lequel l'administrateur définit la politique de sécurité par des règles sur les entités abstraites : rôle, activité et vue sans s'inquiéter de la façon dont l'organisation implémente ces entités.
- Niveau concret où des entités actives (sujet) exécutent des actions sur des objets, sous le contrôle de mécanismes de protection qui mettent en œuvre les règles définies dans la politique.

Nous décrivons les relations existantes entre les entités du niveau concret et les entités de niveau abstrait d'OrBAC :

3.3.1 Concept de rôle et relation Habilite ()

L'entité *Rôle* est utilisée pour représenter la relation entre des organisations et des sujets. L'entité *Sujet* dans ce modèle peut être soit un utilisateur, soit une organisation.

La relation **Habilite (**org, r, s**)** a été introduite pour exprimer le fait que des sujets jouent des rôles dans une organisation. Cela signifie que l'organisation org habilite le sujet s à jouer le rôle r [13].

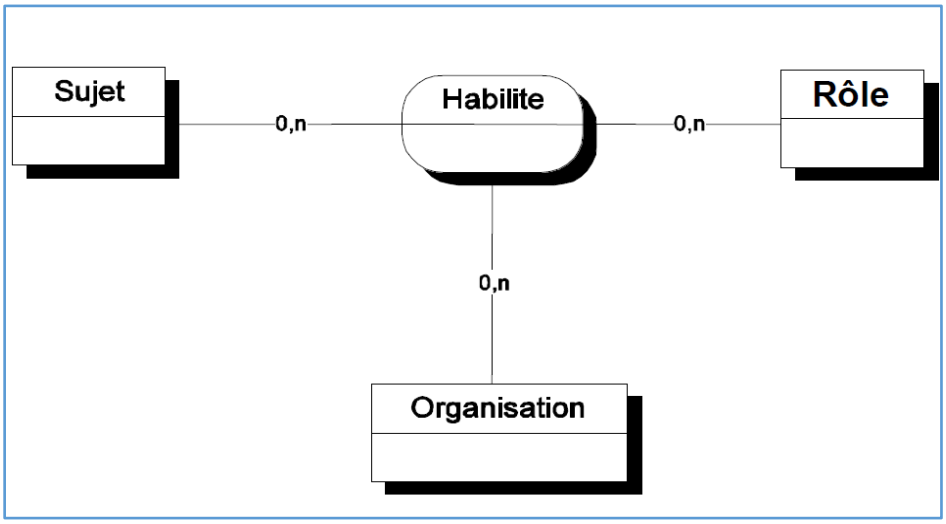


Figure 5: La relation Habilite [17]

Les deux exemples suivants illustrent le fait que les sujets sont soit des utilisateurs, soit des organisations :

- Habilite (CHU, Mohamed, cardiologue) : signifie que l'hôpital CHU habilite Mohamed dans le rôle cardiologue.
- Habilite (CHU, ICU31, unité_de_soins_intensifs) signifie que l'hôpital CHU habilite l'unité ICU31 dans le rôle d'unité de soins intensifs.

3.3.2 Concept de vue et relation Utilise ()

L'entité *Objet* représente des entités passives comme des dossiers médicaux, des fichiers, des imprimantes ou des matériels dans l'hôpital. Ils sont abstraits en *Vue* pour faciliter la mise à jour des politiques de sécurité et structurer des objets quand un nouvel objet est ajouté dans le système.

Intuitivement une vue correspond à un ensemble d'objets qui satisfait une propriété commune et elle caractérise la manière dont les objets sont utilisés dans l'organisation.

La relation **Utilise (**org, o, v**)** est nécessaire pour représenter le lien entre une organisation, un objet et une vue, cela signifie que l'organisation org utilise l'objet o dans la vue v [13].

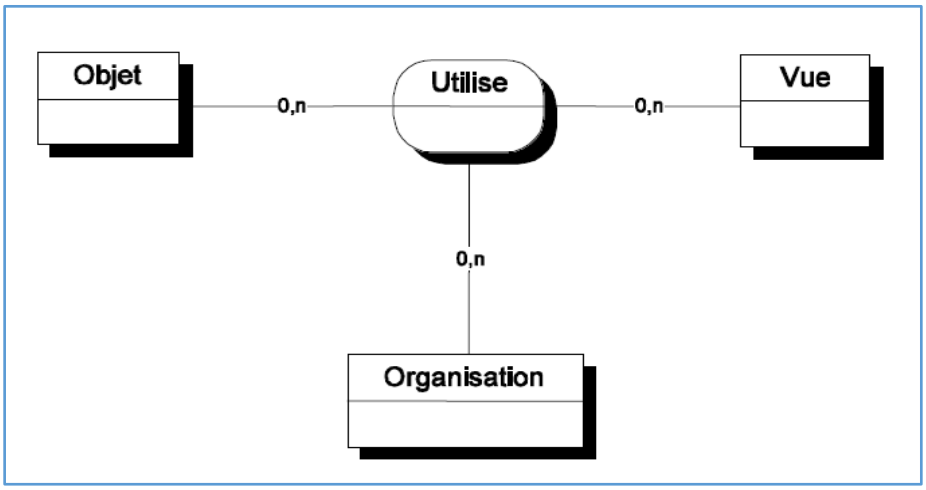


Figure 6: La relation Utilise [17]

Ainsi une même vue peut être définie différemment suivant l'organisation considérée. La vue "dossier médical" peut être définie à l'hôpital CHU comme un ensemble de documents Word et comme un ensemble de documents Latex à l'hôpital Belfort :

- Utilise (CHU, F31.doc, dossier_médical) signifie que l'hôpital CHHU utilise le fichier F31.doc comme un dossier médical
- Utilise (Belfort, F32.tex, dossier_médical) signifie que l'hôpital Belfort utilise le fichier F32.tex comme un dossier médical.

3.3.3 Concept d'activité et relation Considère ()

L'entité *Action* correspond aux actions informatiques qui peuvent être opérées sur le système (*i.e.* lire, écrire, modifier, exécuter).

L'entité *Activité* a été utilisée comme une abstraction des actions qui ont un objectif commun.

Des organisations peuvent considérer qu'une même action réalise des activités différentes selon l'organisation qui l'utilise.

La relation **Considère (**org, act, a**)** a été introduite pour associer les entités Organisation, Action et Activité. Cela signifie que l'organisation org considère l'action act comme faisant partie de l'activité a [13].

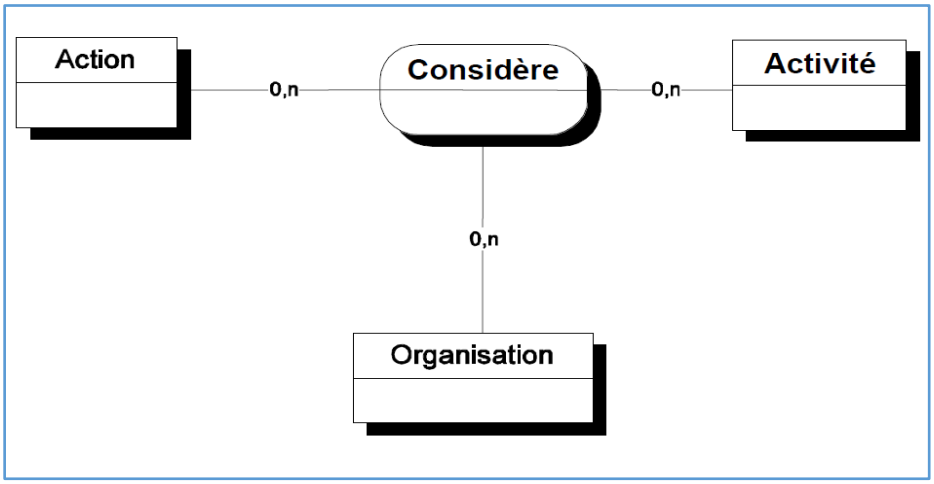


Figure 7: La relation Considère [17]

L'activité pourra correspondre à différentes actions (par exemple, lecture sur un fichier ou sélection dans une base de données). Ainsi, l'activité de consultation d'un dossier médical peut correspondre dans l'organisation "hôpital CHU" à l'action "lire" un fichier, mais peut correspondre à l'action "select" sur une base de données dans l'hôpital Belfort :

- Considère (CHU, lire, consultation) signifie que l'hôpital CHU considère "lire" comme une action de l'activité "consultation".
- Considère (Belfort, select, consultation) signifie que l'hôpital Belfort considère "select" comme une action de consultation.

3.3.4 Concept de contexte et relation Définit ()

L'entité *Contexte* permet aux organisations de spécifier des autorisations de rôles pour effectuer des activités sur les vues dans une circonstance concrète, ce qui n'est pas réalisable dans RBAC.

Dans le modèle RBAC, si une certaine permission est accordée à un rôle, alors tous les utilisateurs qui jouent ce rôle héritent de cette permission, alors que dans le modèle OrBAC, le contexte couvrira des circonstances concrètes ce qui est définie par des règles logiques. Seul l'utilisateur qui satisfera à ces règles pourra hériter de ce privilège.

Chaque contexte peut être vu comme une relation ternaire entre les sujets, les objets et les actions, ce qui est défini dans l'organisation.

Les entités organisation, sujet, objet, action et contexte sont liées par une nouvelle relation appelée « *Définit* ».

La relation **Définit (**org, s, o, act, c**)** signifie que dans l'organisation org, le contexte c'est vrai entre le sujet s, objet o et actions act [13].

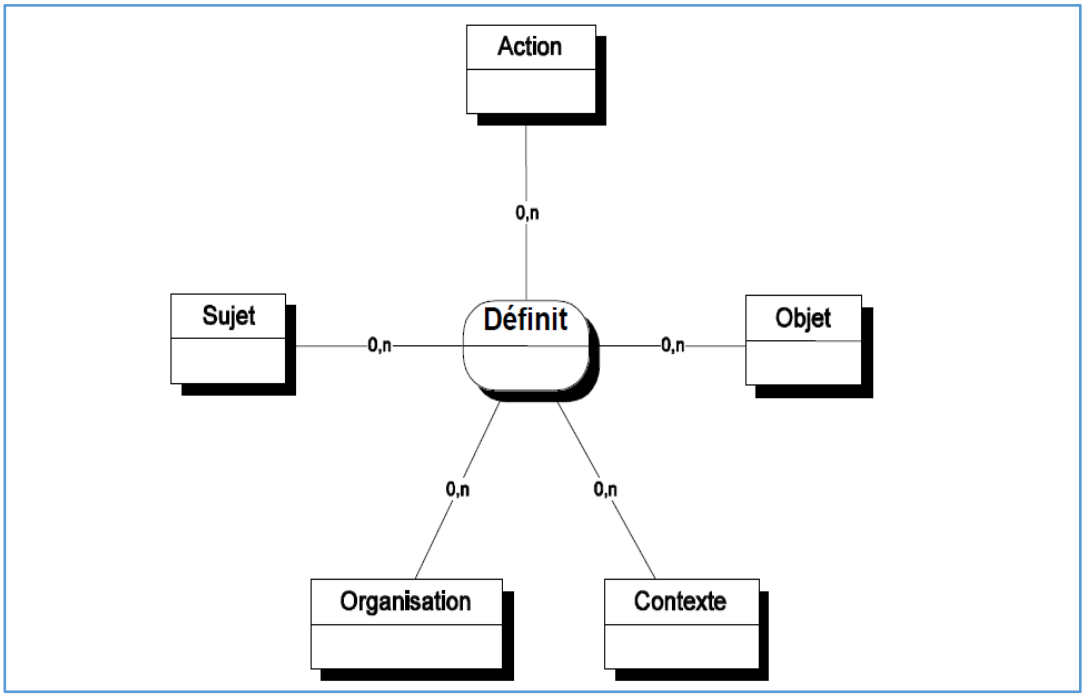


Figure 8: La relation Définit [17]

Par exemple les deux faits suivants

- Définit (CHU, Mohamed, lire, F31.doc, urgence).
- Définit (Belfort, Meriem, lire, F32.tex, médecin_traitant).

Si le premier fait est vrai, alors Mohamed n'a pas besoin d'être le médecin traitant du patient pour consulter son dossier médical F31.doc.

Si le second fait est vrai, alors Meriem doit être le médecin traitant du patient pour lire le dossier médical F32.tex.

Cela signifie que, sauf en cas d'urgence, les médecins ne peuvent consulter que les dossiers médicaux de leurs patients.

3.3.5 Expression de politiques de sécurité dans le modèle OrBAC

Une autre caractéristique d'OrBAC est que les règles exprimées dans ce modèle peuvent définir des permissions, des interdictions, des obligations et des recommandations. Ce modèle est donc beaucoup plus puissant qu'un simple modèle de contrôle d'accès.

On spécifie Les règles en utilisant la relation [13] :

Permission | *Interdiction* | *Obligation* | *Recommandation (org, r, a, v, c)* ; ce qui est une abstraction des permissions et correspond à une relation entre les organisations, les rôles, les activités et les contextes. Cette relation signifie que l'organisation org accorde au rôle r la permission de réaliser l'activité a sur la vue v dans le contexte c.

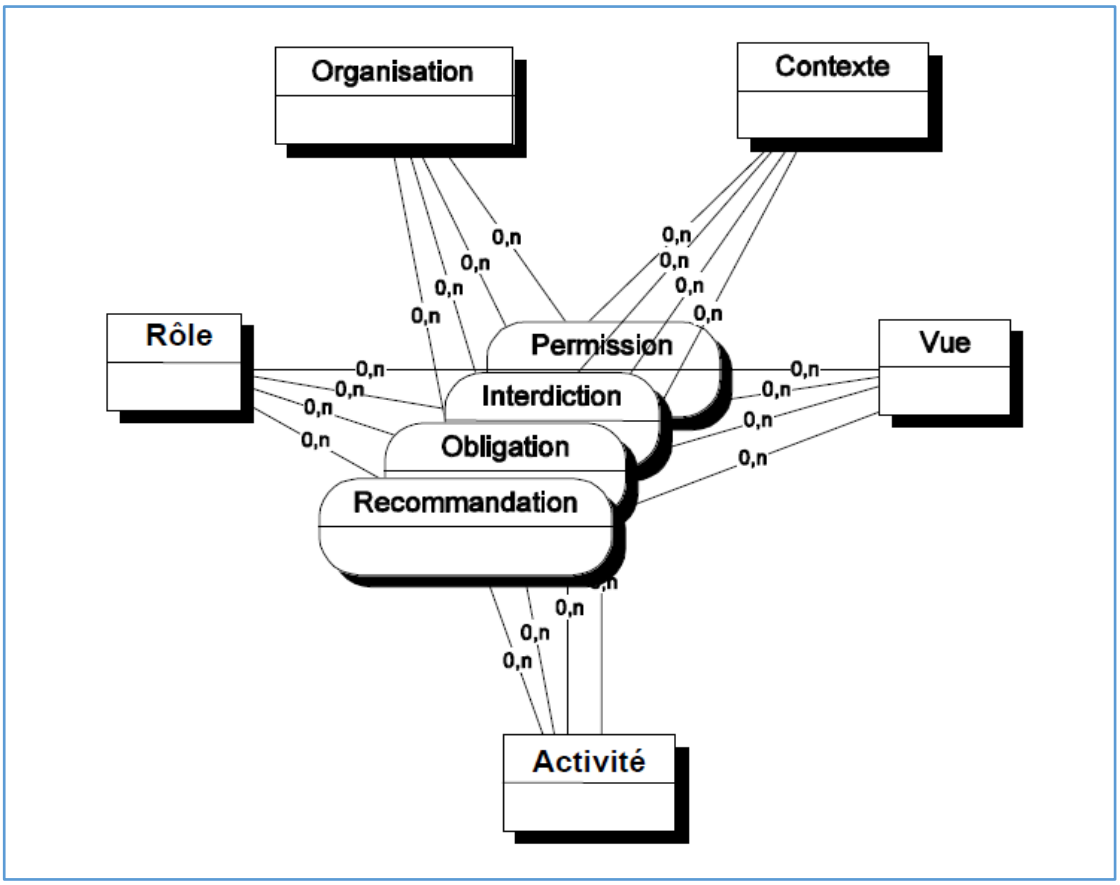


Figure 9: Les relations Permission, Interdiction, Obligation et Recommandation [17]

Par exemple, la politique de sécurité de l'hôpital CHU peut comporter les faits suivants :

- *Permission* (*CHU, médecin, consulter, dossier_médical, urgence*) qui signifie que " l'hôpital CHU accorde aux médecins la permission de consulter n'importe quel dossier médical dans le contexte de l'urgence"
- *Permission* (*CHU, médecin, consulter, dossier_médical, médecin_traitant*) qui signifie que " l'hôpital CHU accorde aux médecins la permission de consulter les dossiers médicaux des patients dont ils sont les médecins traitants ".

3.3.6 Les autorisations concrètes

Le contrôle d'accès bas niveau doit permettre de décrire les actions concrètes que réalisent les sujets sur les objets.

Dans le but de modéliser des permissions concrètes, nous introduisons la relation *Est_permis* entre les sujets, les objets et les actions : si *s* est un sujet, a est une action et *o* est un objet, alors *Est_permis* (*s*, a, *o*) signifie que le sujet *s* a la permission de réaliser l'action a sur l'objet *o*.

Notre relation *Est_permis* est similaire à la notion de permission évoquée dans le modèle HRU {Harison, Ruzzo et Ullman}. Il y a tout de même une différence de taille, dans le modèle HRU, les triplets d'autorisation <*s*, a, *o*> doivent être explicitement décrit ; alors que dans notre modèle, les triplets, qui sont des instances de la relation *Est_permis*, sont dérivés logiquement des permissions accordées aux rôles, aux vues et aux activités par la relation *Permission* [17].

Les autorisations concrètes sont dérivées des permissions abstraites par la règle suivante [18] :

Si *permission (organisation, rôle, activité, vue, contexte)* et
Habilite (organisation, sujet, rôle) et
Considère (organisation, action, activité) et
Utilise (organisation, objet, vue) et
Définit (organisation, sujet, action, objet, contexte)
Alors *Est_permis (sujet, action, objet)*

La figure 10 résume le modèle de sécurité. Il contient huit entités (*Organisation*, *Sujet*, *Rôle*, *Objet*, *Vue*, *Action*, *Activité* et *Contexte*) et douze relations (*Habilite*, *Utilise*, *Considère*, *Permission*, *Interdiction*, *Obligation*, *Recommandation*, *Est_permis*, *Est_interdit*, *Est_obligatoire*, *Est_Recommandé* et *Définit*).

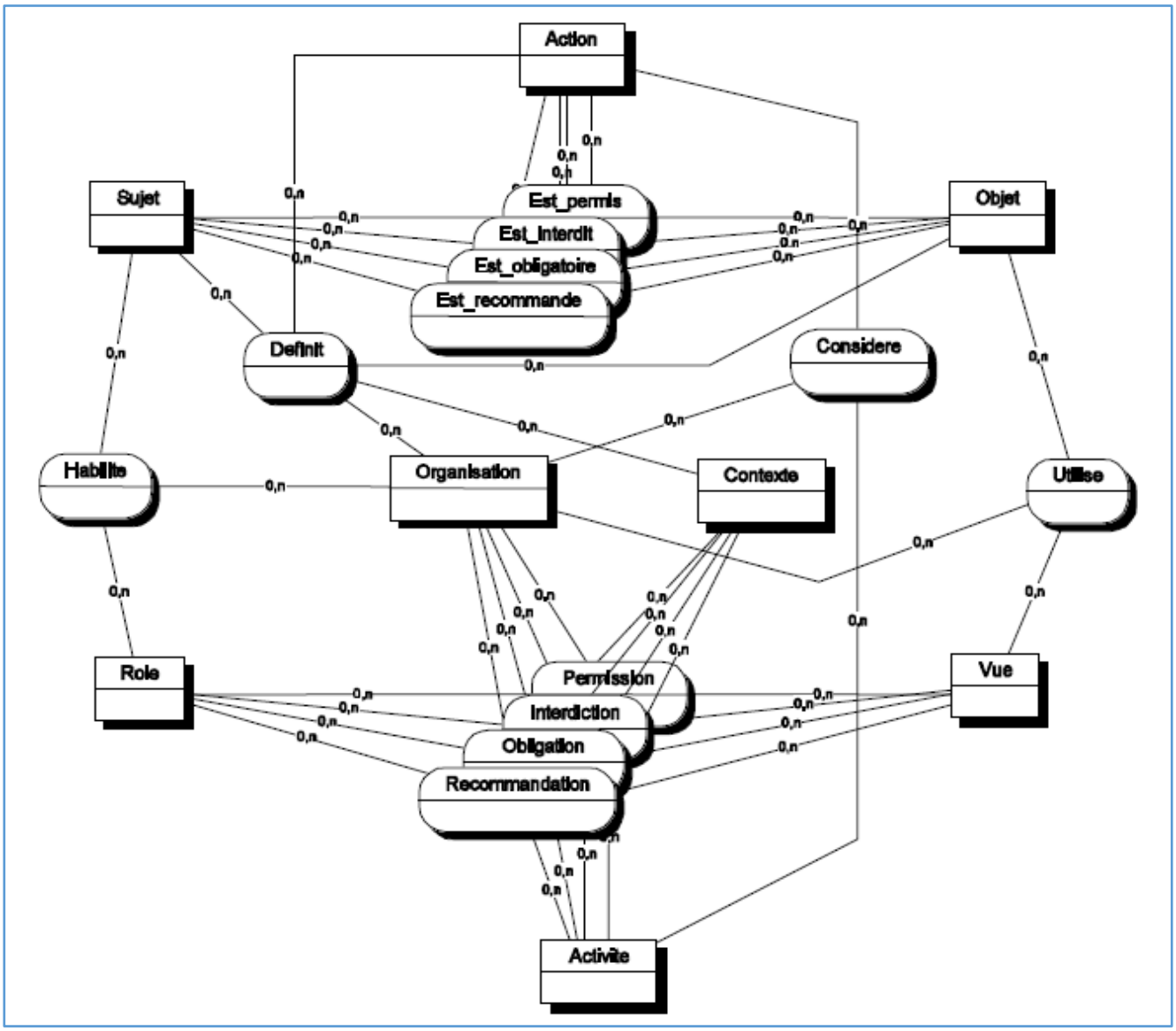


Figure 10: Le modèle OrBAC [17]

4. Conclusion

Dans ce chapitre nous avons présenté le concept de la politique d'accès ensuite nous avons dressé un état de l'art sur les différents modèles de contrôles d'accès existants.

Les modèles de contrôle d'accès comme DAC, MAC ou RBAC ne permettent de modéliser que des politiques de sécurité qui se restreignent à des permissions statiques. Ils n'offrent pas la possibilité d'exprimer des règles contextuelles relatives aux permissions, aux interdictions, aux obligations et aux recommandations. Ce type de règle est particulièrement utile pour exprimer des politiques de sécurité dans le domaine médical.

Dans notre travail nous allons utiliser le contrôle d'accès OrBAC (Organisation Based Access Control) qui permet de spécifier de telles politiques de sécurité contextuelles.

Chapitre 3 Chiffrement des Données

1. Introduction

Parmi les principales propriétés de la sécurité des données médicales recherchées sont : la fiabilité, l'intégrité, la confidentialité, ainsi que le respect de la déontologie médical.

Afin de garantir la sécurité des donnés et pour des besoins de confidentialité, les données sensibles d'un patient doivent être cryptées.

Dans ce chapitre nous allons présenter des notions sur la cryptographie, les méthodes et quelques algorithmes de chiffrement des données.

2. La cryptographie

2.1 Définition

La cryptographie vient des mots en grec ancien *kruptos* (caché) et *graphein* (écrire) et signifie « l'écriture secrète ». Son but était de protéger un message secret lors de sa transmission.

Elle est composée de trois caractéristiques fondamentales [19] :

- La confidentialité qui garantit le caractère secret du message transmis ;
- L'intégrité qui s'assure que le message transmis n'a pas été modifié ;
- L'authentification qui permet de vérifier l'identité de l'émetteur.

2.2 Vocabulaire de base

Beaucoup des termes de la cryptographie existent [20] :

Cryptologie : Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse

Cryptographie : La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.

Cryptanalyse : Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.

Cryptosystème : Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.

Chiffrement : Le chiffrement consiste à transformer une donnée (texte, message) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire.

La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de déchiffrement.

Texte chiffré : Appelé également cryptogramme, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.

Clef : Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.

3. Méthodes de la cryptographie

Il existe deux types de la cryptographie :

- Cryptographie symétrique : Chiffrement à clé privée (secrète)
- Cryptographie asymétrique : Chiffrement à clé publique

3.1 La cryptographie symétrique

Les caractéristiques de la cryptographie symétrique sont [20] :

- Les clés sont identiques : KE = KD = K,
- La clé doit rester secrète,
- Les algorithmes les plus répandus sont le DES, AES, 3DES, ...
- Au niveau de la génération des clés, elle est choisie aléatoirement dans l'espace des clés,
- Ces algorithmes sont basés sur des opérations de transposition et de substitution des bits du texte clair en fonction de la clé,
- La taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut aller jusque 256,
- L'avantage principal de ce mode de chiffrement est sa rapidité,
- Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, on préfèrera l'échange manuel. Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un système à N utilisateurs, il y aura N*(N − 1) /2 paires de clés.

Il existe deux types d'algorithmes symétrique :

- Les algorithmes de chiffrement en continu (par flot) ; qui agissent sur le message en clair un bit à la fois.
- Les algorithmes de chiffrement par bloc ; qui opèrent sur le message en clair par groupes de bits appelés blocs.

3.1.1 Algorithmes de chiffrement par flux

Le principe consiste à générer un flux pseudo aléatoire et de le combiner avec l'information bit à bit par l'opération XOR

A la réception, on applique le même mécanisme, et on restitue l'information [21].

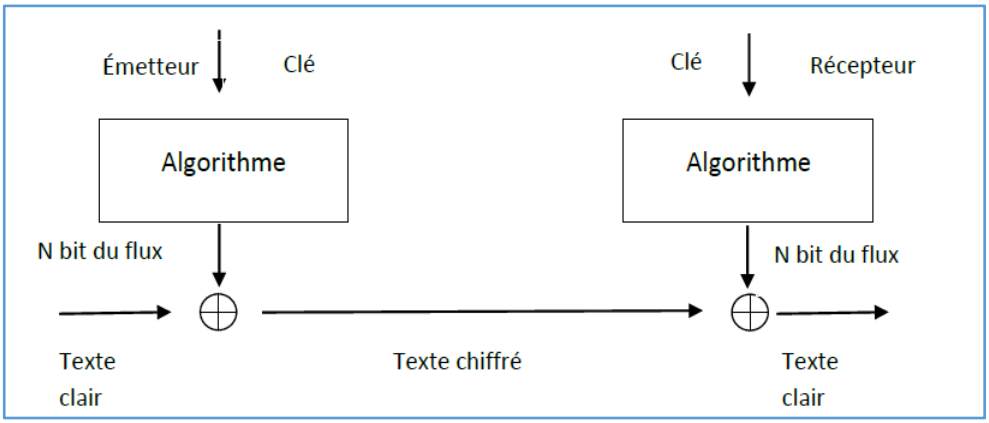


Figure 11: Schéma chiffrement par flot [21]

3.1.2 Algorithmes de chiffrement par bloc

Ils opèrent sur le message en clair par groupe de bits. La taille typique des blocs est 64 bits, ce qui est assez grand pour interdire l'analyse et assez petit pour être pratique [21].

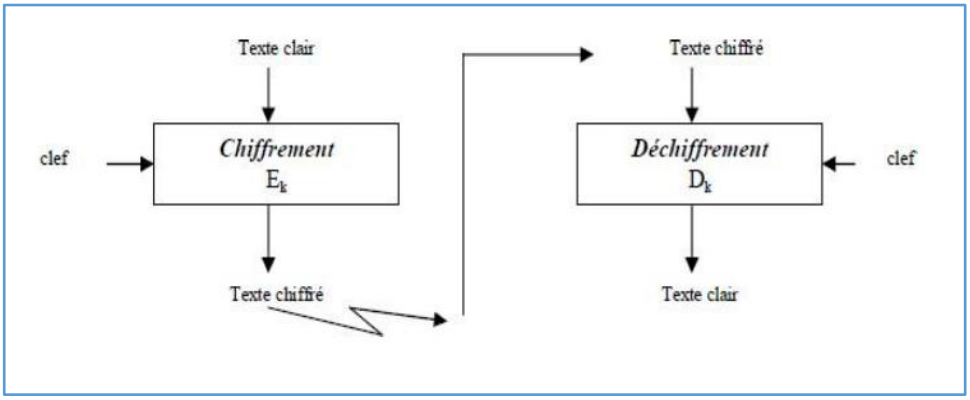


Figure 12: Chiffrement par Bloc [21]

Les algorithmes de chiffrement par blocs peuvent être utilisés suivant différents modes, dont les deux principaux sont :

- Le mode ECB (*Electronic Code Book*).
- Le mode CBC (*Chipher Block Chaining*).

3.1.2.1 Le mode ECB (Electronic Code Book)

Ce mode permet le chiffrement en parallèle des différents blocs composant un message. Même bloc de message en clair sera toujours chiffré en un même bloc de message chiffré. Une propagation d'erreur importante; si un bit du message chiffré est modifié pendant le transfert, tout le bloc de message en clair correspondant sera faux [21].

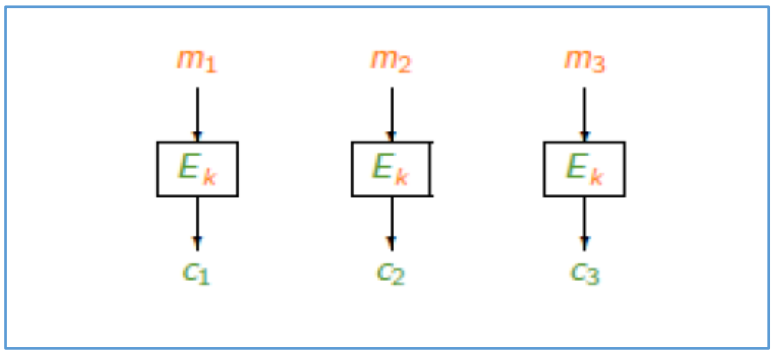


Figure 13: Le mode ECB [21]

- **Chiffrement :** Chaque bloc clair mi est chiffré indépendamment et donne un bloc chiffré Ci=Ek(Mi).
- **Déchiffrement :** Chaque chiffré est déchiffré indépendamment pour donner le clair correspondant Mi=Dk(Ci).

3.1.2.2 Le mode CBC (Cipher Block Chaining)

La structure du message en clair est masquée par le chaînage. Un attaquant ne peut plus manipuler le cryptogramme, excepté en retirant des blocs au début ou à la fin. Il n'est plus possible de paralléliser le chiffrement des différents blocs.

Le chaînage de bloc n'entraîne pas une propagation d'erreur importante ; si un bit du message chiffré est modifié au cours du transfert, seul le bloc de message en clair correspondant et un bit du bloc de message en clair suivant seront endommagés [21] :

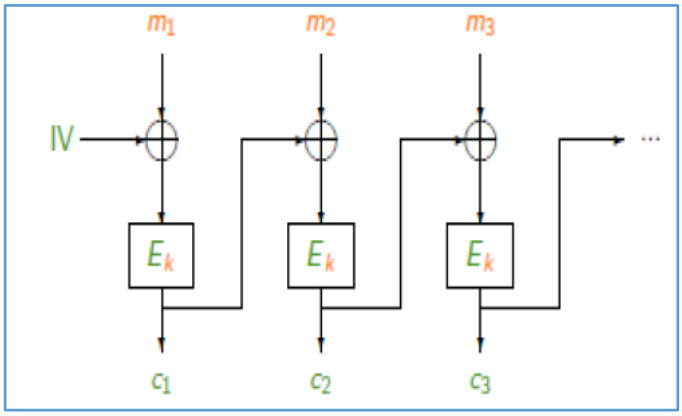


Figure 14: Chiffrement CBC [21]

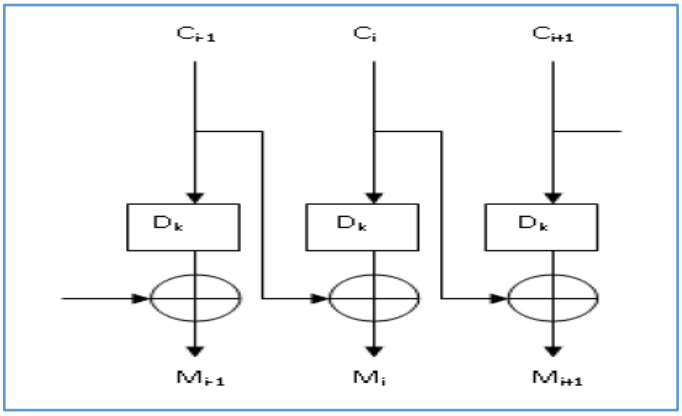


Figure 15: Déchiffrement CBC [21]

- **Chiffrement :** Un vecteur d'initialisation IV est généré aléatoirement
  Ci =Ek (Mi ⊕ Ci-1). Le vecteur IV est transmis avec les blocs chiffrés.
- **Déchiffrement :** Mi= Ci-1 ⊕ Dk(Ci).

3.1.3 Comparaisons des chiffrements par blocs et par flots

Le tableau suivant résume la différence entre le chiffrement symétrique par bloc et par flot

	Par Blocs	**Par Flots**
Avantages	• Réutilisation des clés	• Rapidité • Moins de code d'implémentation
Inconvénients		• Deux utilisations d'une même clé facilite la cryptanalyse
Applications	• Transfert de fichiers	• Chiffrement de canal de communication

Tableau 3: Comparaison des chiffrements par blocs et par flots [20]

3.2 La cryptographie asymétrique

Ce système de cryptage utilise deux clés différentes pour chaque utilisateur : L'une est privée et n'est connue que de l'utilisateur ; l'autre est publique et donc accessible à tout le monde. Un message crypté avec une clé publique ne puisse être décrypté qu'avec la clé privée correspondante. Une clé est donc utilisée pour le cryptage et l'autre pour le décryptage [22].

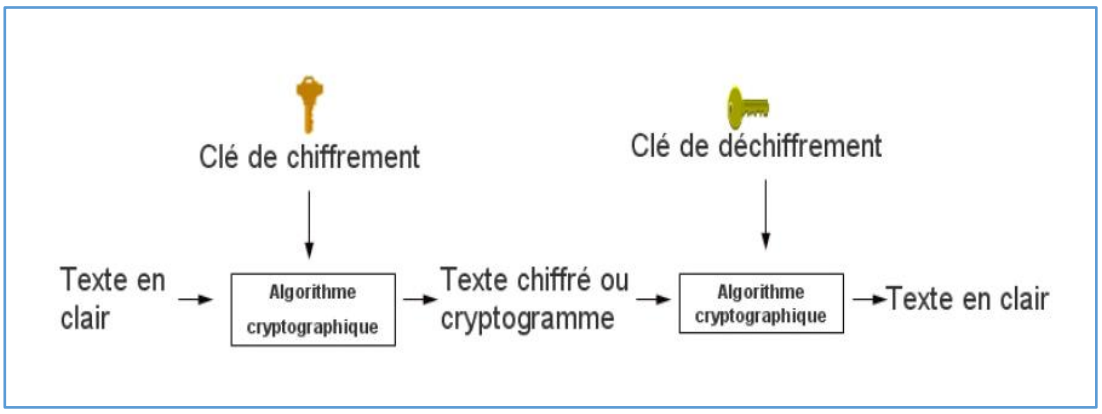


Figure 16: Chiffrement asymétrique [20]

Nous pouvons définir le chiffrement asymétrique comme suit [23] :

Définition : Un schéma de chiffrement asymétrique ε est défini pour un paramètre de sécurité *k* par un triplé (*Enc, Dec, KeyGen*)

- **KeyGen (k) → (*PK, SK*)**
  C'est un algorithme aléatoire, qui prend en entrée un paramètre de sécurité *k* et génère aléatoirement en sortie une clé publique *PK* et une clé secrète *SK*.
- **Enc (PK, M) → *C***
  C'est un algorithme déterministe, qui à partir d'un message M et d'une clé de chiffrement publique *PK*, génère un chiffré *C*. Avec *M* un message de l'espace des messages M et *C* un chiffré de l'espace des chiffrés C.
- **Dec (SK, C) → M**
  L'algorithme de déchiffrement prend en entrée une clé secrète de déchiffrement *SK* et un chiffré *C* et retourne le message claire correspondant *M*, si la clé *SK* est correcte.

Les algorithmes asymétriques possèdent deux modes de fonctionnement [21] :

- ***Le mode chiffrement:*** L'émetteur chiffre avec la clé publique du destinataire, le destinataire déchiffre avec sa clé privée. Dans ce mode, l'émetteur est sûr que seul le destinataire peut déchiffrer le fichier.
- ***Le mode signature:*** L'émetteur signe avec sa clé privée, le destinataire vérifie la signature avec la clé publique de l'émetteur. Dans ce mode, le destinataire est sûr que c'est bien l'émetteur qui a envoyé le fichier.

Le principal avantage du cryptage à clé publique est de résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé. Bien que plus lent que la plupart des cryptages à clé privée il reste préférable pour 3 raisons [22] :

- Plus évolutif pour les systèmes possédant des millions d'utilisateurs.
- Authentification plus flexible.
- Supporte les signatures numériques.

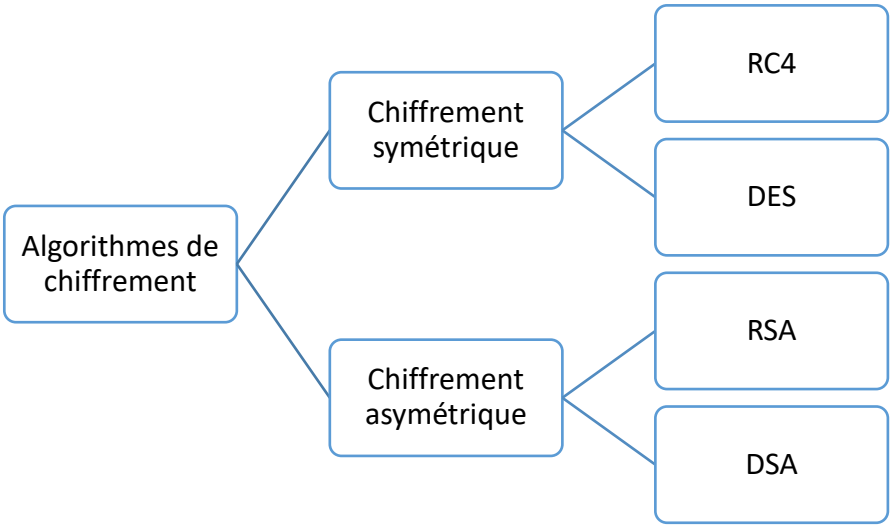
3.3 Tableau comparatif entre le cryptage symétrique et asymétrique

Le tableau 4 représente les avantages et les inconvénients des deux méthodes de la cryptographie symétrique et asymétrique [24] :

Cryptage symétrique	Cryptage asymétrique
• Chiffrement à clé privé (une seule clé est utilisée pour le cryptage et le décryptage).	• Chiffrement à clé publique (utilisation de la clé publique pour le cryptage et la clé privé pour le décryptage).
• Très facile.	• Difficile par rapport au cryptage symétrique.
• Très rapide.	• Plus lent.
• Les clés de chiffrement symétrique doivent être conservées en toute sécurité.	• les clés publiques qu'ils utilisent sont sans danger pour être publié n'importe où parce que pour obtenir la clé privée à partir d'une clé publique peut prendre de très longues durées de travail.

Tableau 4:Comparaison entre le cryptage symétrique et asymétrique [24]

4. Quelques exemples d'algorithmes de chiffrement



4.1 Le RC4 (Rivest Cipher 4)

Il est créé en 1987 par Ron Rivest. Il s'agit probablement du chiffrement par flots le plus utilisé actuellement.

On le retrouve notamment dans le standard SSL/TLS, dans Oracle Secure SQL, ou encore dans le protocole WEP (Wired Equivalent Privacy, de la norme 802.11). Ce dernier fut remplacé par le WPA (Wi-Fi Protected Access), mais celui-ci utilise toujours le RC4 [20].

Principe général du RC4

RC4 fonctionne de la façon suivante : la clef RC4 permet d'initialiser un tableau de 256 octets en répétant la clef autant de fois que nécessaire pour remplir le tableau. Par la suite, des opérations très simples sont effectuées : les octets sont déplacés dans le tableau, des additions sont effectuées, etc. Le but est de mélanger autant que possible le tableau. Finalement on obtient une suite de bits pseudo-aléatoires qui peuvent être utilisés pour chiffrer les données via un XOR.

Description détaillée

RC4 est un générateur de bits pseudo-aléatoires dont le résultat est combiné avec le texte en clair via une opération XOR, le déchiffrement se fait de la même manière.

Pour générer le flot de bits, l'algorithme dispose d'un état interne, tenu secret, qui comprend deux parties [25]:

- Permutation S de tous les 256 octets possibles
- Pointeurs i et j de 8 bits qui servent d'index dans un tableau

La permutation est initialisée grâce à la clé de taille variable, typiquement entre 40 et 256 bits, grâce au key Schedule de RC4.

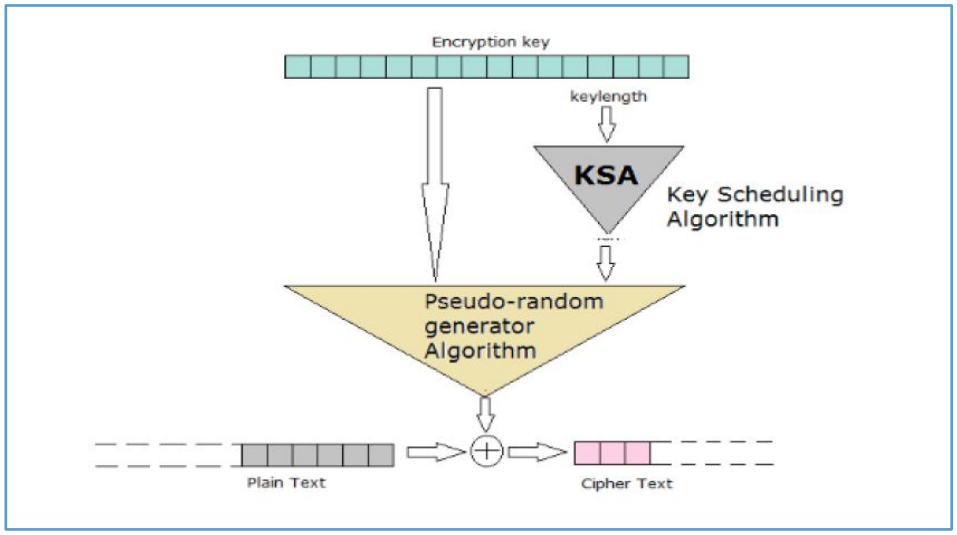


Figure 17: Schéma de représentation RC4 [25]

Un chiffrement par RC4 est très rapide, comme le montre le tableau suivant:

Algorithme	Longueur de la clé	Vitesse (en Mbps)
DES	56	9
RC4	variable	45

Tableau 5: Vitesses de quelques chiffrements symétriques [20]

4.2 Le DES (Data encryption standard)

Le *Data Encription Standard* (standard de chiffrement de données a été publié en 1977)**.** Il est un Cryptosystème agissant par blocs. Il découpe virtuellement le texte clair en blocs de 64 bits qu'il code séparément, puis qu'il concatène. Un bloc de 64 bits du texte clair entre par un coté de l'algorithme et un bloc de 64 bits de texte chiffré sort de l'autre côté. L'algorithme est assez simple puisqu'il ne combine en fait que des permutations et des substitutions.

C'est un algorithme de chiffrement à clef secrète(symétrique). La clef sert donc à la fois à chiffrer et à déchiffrer le message. Cette clef a ici une longueur de 64 bits, c'est-à-dire 8 caractères, mais dont seulement 56 bits sont utilisés. On peut donc éventuellement imaginer un programme testant l'intégrité de la clef en exploitant ces bits inutilisés comme bits de contrôle de parité.

L'entière sécurité de l'algorithme repose sur les clefs puisque l'algorithme est parfaitement connu de tous. La clef de 64 bits est utilisée pour générer 16 autres clefs de 48 bits chacune qu'on utilisera lors de chacune des 16 itérations du D.E.S. Ces clefs sont les mêmes quel que soit le bloc qu'on code dans un message [20].

Les grandes lignes de l'algorithme sont les suivantes [25] **:**

1. Diversification de la clé (64bit): fabrication de 16 sous-clés.
2. Permutation initiale.
3. Calcul médian (16 fois) : application d'un algorithme complexe appliqué en fonction de la clé.
4. Permutation finale.

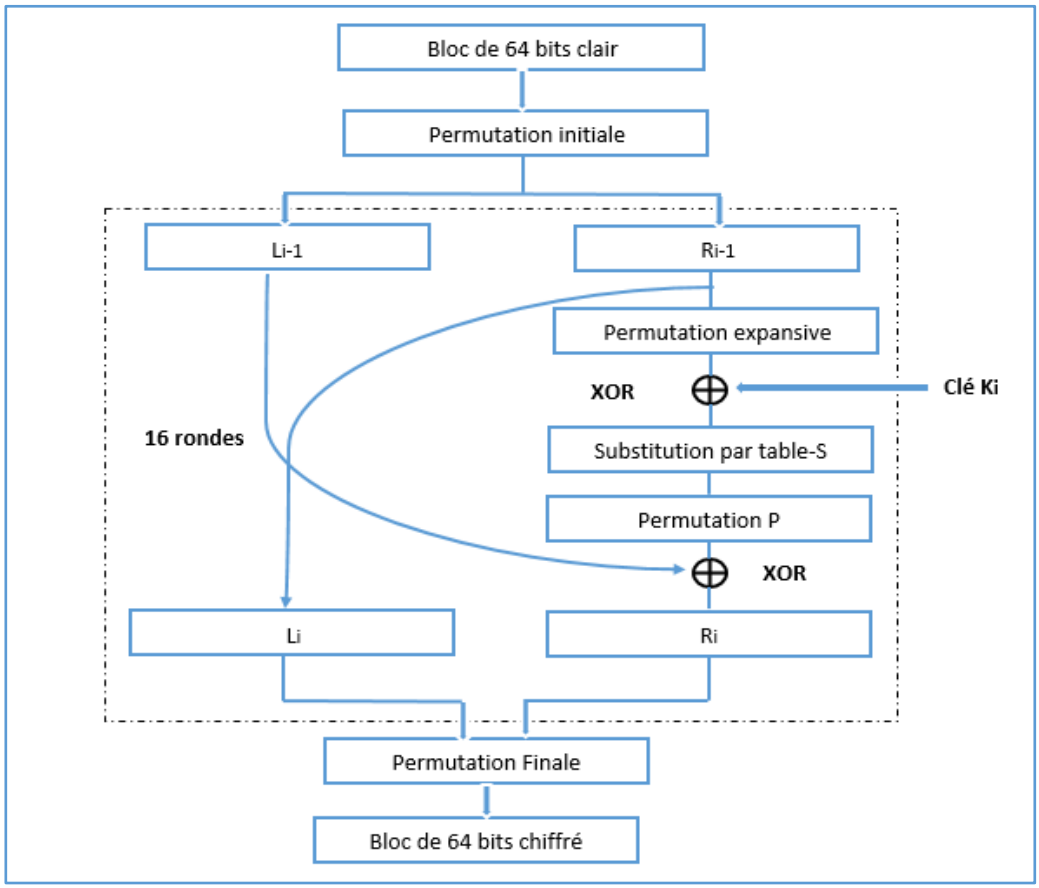


Figure 18: Algorithme principale de D.E.S [25]

4.3 Le RSA (Rivest - Shamir – Adleman)

Le RSA est le plus célèbre et le plus répondu des algorithmes asymétriques. Il Fut conçu par Ron Rivest, Adi Shamir et Len Alderman, en 1977. Cet algorithme est basé sur la factorisation des nombres premiers.

Grâce à sa théorie, le RSA sert aussi bien à effectuer le chiffrement des données de taille réduite mais permet également d'assurer le service d'authentification [22].

Principe

Il est basé sur la création d'une clé publique qui est diffusée, utilisée pour chiffrer le message et d'une clé privée gardée secrète utilisée pour déchiffrer le message [26] :

1. **Génération de clés**
   - Choisir deux nombres premiers p et q, aléatoirement, p≠q.
   - Calculer n = p*q et ϕ(n)=(p-1) *(q-1).
   - e un entier choisi aléatoirement tel que 1 < e < ϕ (n) et pgcd(e, ϕ(n)) =1 (e et ϕ(n) sont premier entre eux.
   - Calculer d tel que e*d mod ϕ(n)=1.
   - d est calculé en utilisant l'algorithme d'Euclide étendu , et ceci revient à résoudre une équation diophantienne.
   - (e,n) est la clé publique et (d,n) est la clé privée.
2. **Chiffrement**
   - L'entier m est le message à chiffrer tel que 1<m<n , et c est le chiffré, calculé comme suit : c= me mod n.
   - Ce cryptogramme sera envoyé au récepteur concerné.
3. **Déchiffrement**

   Le récepteur à son niveau déchiffre ce cryptogramme à l'aide de sa clé privée d, comme suit : m = cd mod n.

Exemple : Soit p=23, q=19, e=13.

Donc n=p*q=23*19=437 et ϕ(n)=22*18=396.

e * d mod ϕ (n)=1 -> e * d - k * ϕ(n)=1 -> d=61 (résolution d'équation diophantienne.

Soit le message à chiffrer m= 309.

Le chiffré c=me mod n=30913 mod 437 =245.

Pour déchiffrer ce cryptogramme m=24561 mod 437 =309.

4.4 DSA (Digital signature Algorithm)

Le DSA est un algorithme de signature numérique standardisé par le NIST aux Etats-Unis, du temps ou le RSA était encore breveté. Une révision mineure a été publié en 1996 « FIPS 186-1 » 'Fideral Information Processing standard' et le standard a été amélioré en 2002.
Le DSA ne peut être utilisé pour chiffrer des messages ou transmettre des clés [27].

5. Chiffrement par attributs

L'inconvénient d'un schéma de chiffrement comme RSA ou DES est la difficulté de mettre en place un contrôle d'accès avec une granularité fine pour le partage des données, en particulier dans le cas où nous ne connaissons pas l'identité des utilisateurs au préalable. Une solution viable à ces problématiques est donnée par le chiffrement par attributs ou « Attribute Based Encryption » (ABE), qui incorpore un processus de génération des clés de chiffrement et de déchiffrement et la notion de politique d'accès basée sur des attributs. Ce schéma offre simultanément des fonctionnalités de chiffrement et de contrôle d'accès.
Le chiffrement par attributs ABE a été introduit en 2005 par Sahai et Waters comme une évolution du chiffrement base sur les identités floues, lui-même étant une amélioration du chiffrement base sur l'identité. C'est un schéma de chiffrement a clé publique (asymétrique) du type un-a-plusieurs, c'est-à-dire qu'on chiffre avec une seule clé et on a la possibilité de générer plusieurs clés pour déchiffrer. Un avantage évident de cette technique est que chaque utilisateur a une clé dédiée, en cas de révocation d'une clé, il n'est pas nécessaire de refaire le chiffrement de données.
En plus de sécuriser la transmission et le stockage des données, ABE fournit un contrôle d'accès à forte granularité, une gestion évolutive des clés et une distribution flexible des données. Il permet de chiffrer les données et d'assurer le partage sur la base d'attributs descriptifs, sans aucune connaissance préalable de l'identité des destinataires. Seules les entités avec des attributs qui satisfont une politique d'accès aux données peuvent déchiffrer un texte [23].

Une politique d'accès est généralement exprimée sous forme d'un arbre, dans lequel les feuilles représentent des attributs, et les nœuds internes représentent des opérateurs booléens « AND » qui prend la valeur 2of 2 et « OR » qui prend la valeur 1of2.

Exemple

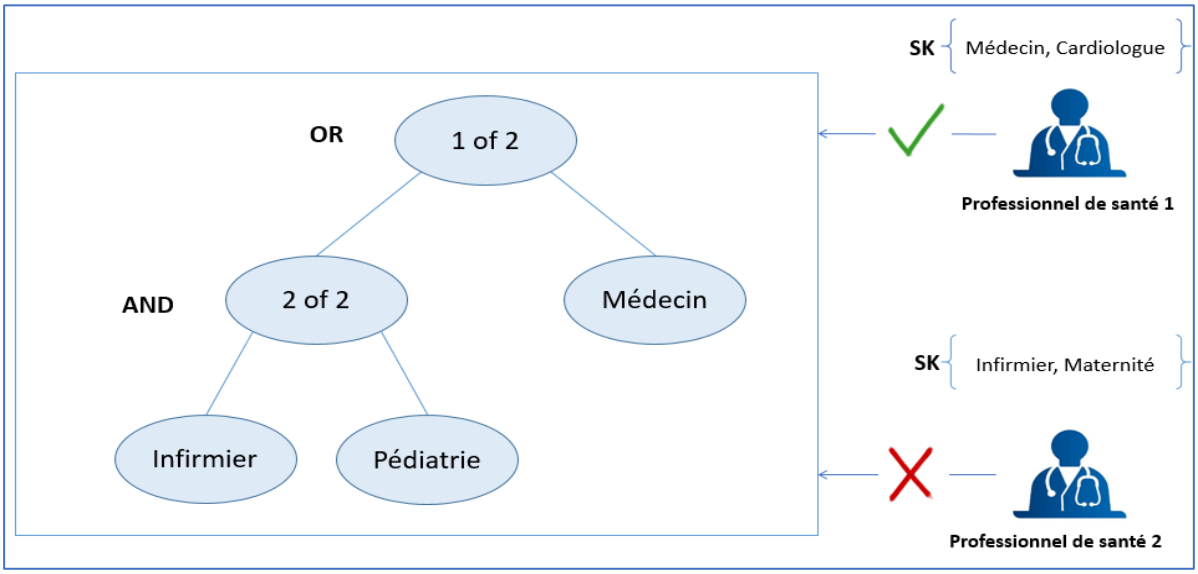


Figure 19: Exemple de la structure d'accès ABE

Dans l'exemple de la figure 19 nous avons supposons :

- Les attributs {Médecin, Cardiologue, Infirmier, Maternité}
- **Professionnel de santé 1** a une clé pour les attributs {Médecin, Cardiologue}
- **Professionnel de santé 2** a une clé pour l'attribut {Infirmier, Maternité}

Si une donnée est chiffrée par la politique (Infirmier et Pédiatrie ou Médecin) :

- **Professionnel de santé 1** pourra le déchiffrer.
- **Professionnel de santé 2** ne pourra pas le déchiffrer.

5.1 Approches ABE

Les deux principales variantes d'ABE sont :

- Key-Policy Attribute Based Encryption (KP-ABE).
- Ciphertext-Policy Attribute Based Encryption (CP-ABE).

5.1.1 KP-ABE

Il a été développée par Goyal *et al.* en 2006. Pour KP-ABE, la politique d'accès est intégrée dans la clé secrète, en d'autres termes, on décide pour chaque utilisateur quels sont les objets auxquels il aura accès. On attache à chaque texte chiffré un ensemble d'attributs.

Une clé secrète donnée, avec une politique d'accès donnée, ne peut déchiffrer que le texte chiffré ayant les attributs qui satisfont sa politique d'accès [23].

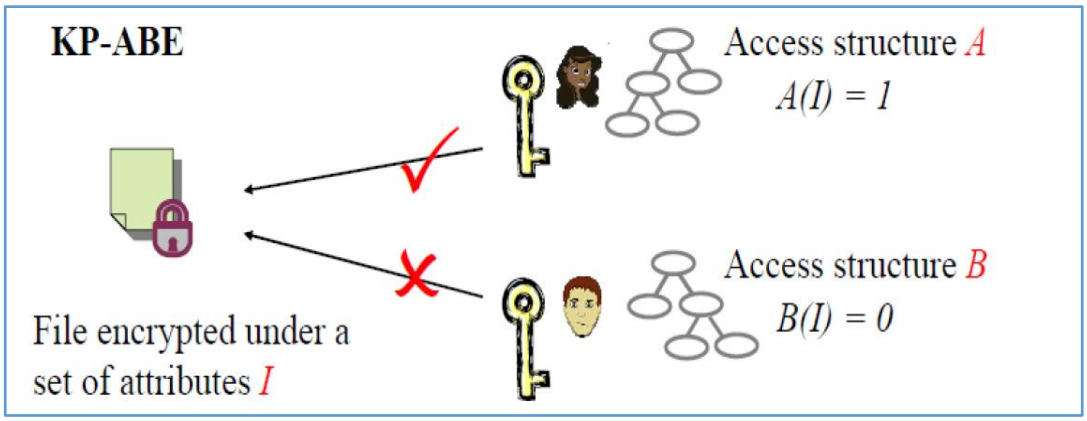


Figure 20: Chiffrement KP-ABE [28]

5.1.2 CP-ABE

Proposée pour la première fois par Béthencourt *et al.* en 2007, dans laquelle la politique d'accès est intégrée dans le texte chiffré et les clés secrètes sont générées avec un ensemble d'attributs décrivant l'utilisateur légitime qui pourra déchiffrer ce texte. Seul les clés secrètes avec un ensemble d'attributs qui satisfait la politique d'accès peuvent récupérer le texte en clair [23].

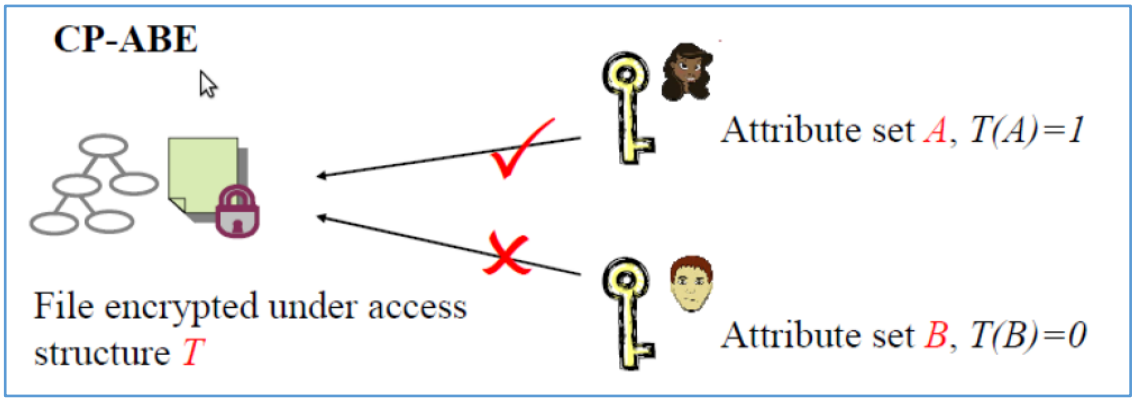


Figure 21: Chiffrement CP-ABE [28]

5.2 Algorithme ABE

Le chiffrement ABE est défini comme un quadruple de quatre algorithme [23]:

- Configuration (*Setup*).
- Chiffrement *(Enc).*
- Générations des clés (*KeyGen).*
- Déchiffrement (*Dec).*

La principale différence entre le KP-ABE et le CP-ABE est que dans CP-ABE, la politique d'accès est incluse dans le texte chiffré et les attributs sont inclus dans la clé de déchiffrement, alors que dans KP-ABE, c'est exactement l'inverse.

Algorithme	Description des entrées/sorties
Setup	• *Entrée :* Un paramètre de sécurité *l*. • *Sortie :* Une clé publique de chiffrement *Pk* et une clé secrète principale *MSk* qui servira à générer les clés secrètes de déchiffrement.
Enc	• Entrée : Message à chiffrer *m*, la clé publique *Pk* et un ensemble d'attributs *ai* dans le cas de KP-ABE ou une politique d'accès *P* dans le cas de CP-ABE. • Sortie : Le message chiffré *c*.
KeyGen	• Entrée : La clé secrète principale *MSk* et un ensemble d'attributs *ai* dans le cas de CP-ABE ou une politique d'accès *P* dans le cas de KP-ABE. • Sortie : une clé de déchiffrement secrète *Sk*, liée à un ensemble d'attributs *ai* dans le cas de CP-ABE ou à une politique d'accès *P* dans le cas de KP-ABE.
Dec	• Entrée : Le message chiffré *c* et une clé de déchiffrement *Sk*. • Sortie : Si l'ensemble d'attributs *ai* satisfont la politique *P* alors sortie *m* sinon $\perp$

Tableau 6: Tableau détaillant de l'algorithme ABE [23]

6. Conclusion

Dans ce chapitre nous avons étudié les différentes techniques de la cryptographie ensuite nous avons donné quelques exemples d'algorithmes de chiffrement de données et à la fin nous avons détaillé la technique de chiffrement par attributs.

Le cryptage est apparu afin de renforcer la sécurité. Les fonctionnalités de chiffrement ABE sont intéressantes pour une solution qui assure la protection de la vie privée des patients.

Chapitre 4 Conception de la Couche de Sécurité pour le SGDP

1. Introduction

Dans ce chapitre, nous proposons des mécanismes informatiques à implémenter formant la couche de sécurité pour le système de gestion du dossier médical électronique.

Nous commençons par décrire l'architecture générale de la couche de sécurité.

Ensuite, nous détaillons la conception de notre application web.

2. Description de la solution

Pour garantir la confidentialité et l'intégrité des données dans les systèmes de gestion des dossiers patients électroniques SGDP, l'accès au système et aux données médicales doit être contrôler selon des critiques qui respecte la déontologie médicale et les droits de malade.

Le modèle de contrôle d'accès permet la gestion des accès des professionnels de santé, et les techniques de chiffrement/ déchiffrement peuvent renforcer la confidentialité afin de sécuriser la donnée (donnée médical sensible), le transfert et le partage de cette dernière.

Nous allons proposer une solution qui combine ces mécanismes (contrôle d'accès et chiffrement /déchiffrement), Cette combinaison représente la couche de sécurité pour notre SGDP.

A partir de notre étude (chapitre 2 et chapitre 3), Nous allons utiliser les deux approches : le contrôle d'accès à base d'organisation OrBAC et le chiffrement par attribut CP-ABE. Pour les attributs, nous allons utiliser le principe d'OrBAC dans l'hôpital (Rôle , Contexte).

Notre système contient 2 éléments principaux (Figure 22) :

- Autorité des attributs : c'est un administrateur qui est chargé d'initialiser tous les paramètres du système.
- Professionnels de santé : sont les employés de l'hôpital, ils peuvent être des médecins ou bien des infirmiers.

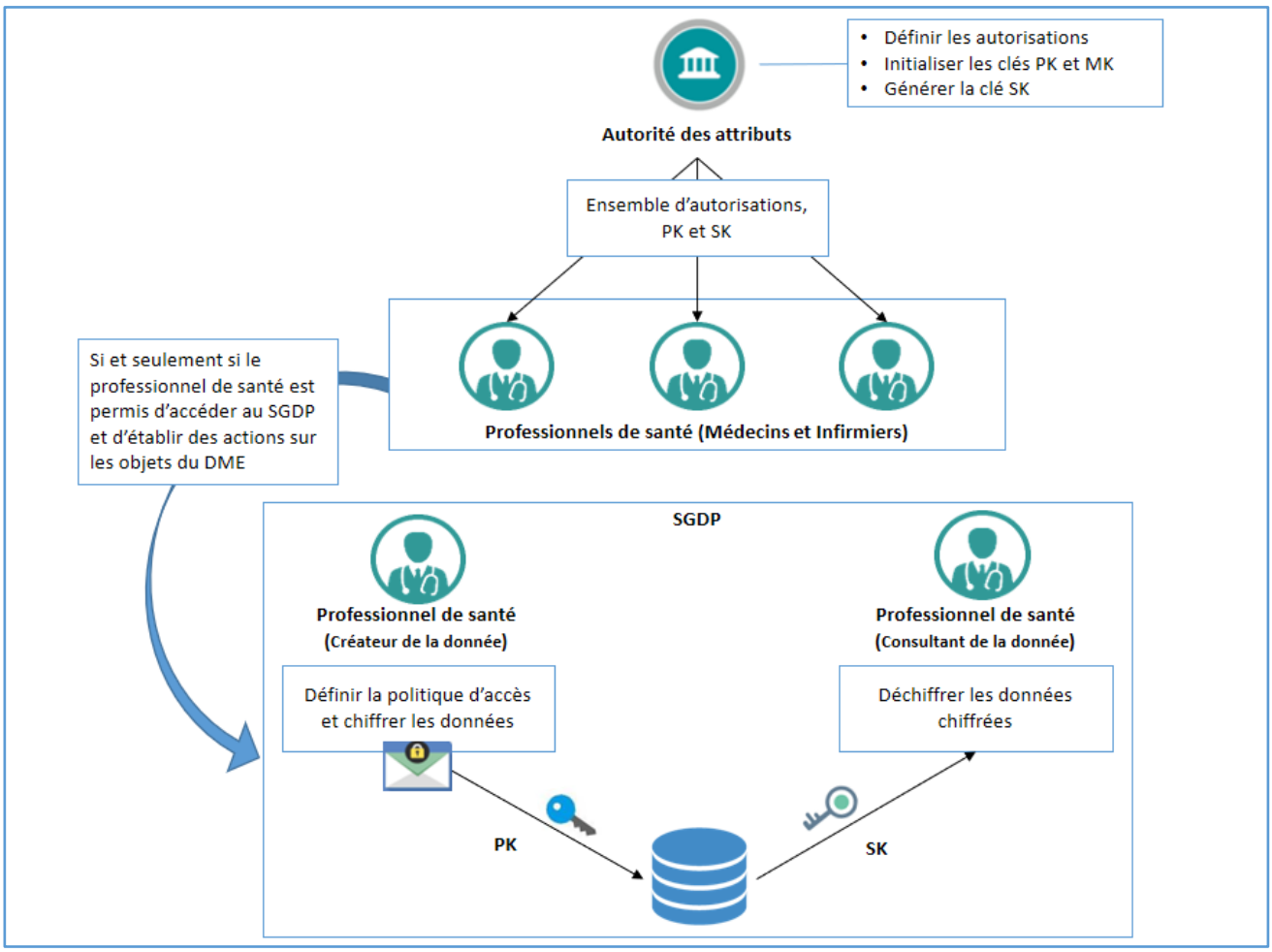


Figure 22: Architecture générale du système

La couche de sécurité de notre SGDP est décrite dans les sections suivantes :

2.1 Le contrôle d'accès basé sur l'organisation

Ce modèle, est centré sur le concept d'organisation, dans notre cas nous avons une seule organisation qui est l'hôpital.

En effet, tous les autres concepts (*Rôle, Vue, Activité, Contexte)* que nous avons définis et qui permettent de spécifier une politique de sécurité dépendent de l'hôpital.

Les tableaux suivants décrit les entités avec leurs attributs ainsi que les relations entre eux :

Entités	Attributs		
Rôle	Admin	Médecin	Infirmier
Contexte	Normal	Urgence	Covid-19

Tableau 7: Les entités Rôle et Contexte

Relation	Entités	Attributs		
Utilise	**Vue**	ORBAC	DME	ABE
	Objet	Autorisation Employé Utilisateur	Consultation Examen radiologique Examen biologique Rendez vous Ordonnance Patient	Clé principale Clé publique Clé secrète

Tableau 8: La relation Utilise

Relation	Entités	Attributs			
Considère	**Activité**	Consulter	Ajouter	Modifier	Supprimer
	Action	Show index	Create	Update	Destroy

Tableau 9: La relation Considère

Relation	Entités	Attributs
Permission	**Rôle**	(Admin, Consulter, ORBAC, Normal) (Admin, Ajouter, ORBAC, Normal) (Admin, Modifier, ORBAC, Normal) (Admin, Supprimer, ORBAC, Normal)
	Activité	(Admin, Ajouter, ABE, Normal) (Médecin, Consulter, DME, Normal) (Médecin, Consulter, DME, Urgence) (Médecin, Consulter, DME, Covid-19)
	Vue	(Médecin, Ajouter, DME, Normal) (Médecin, Ajouter, DME, Urgence) (Médecin, Ajouter, DME, Covid-19) (Infirmier, Consulter, DME, Normal)
	Contexte	(Infirmier, Consulter, DME, Urgence) (Infirmier, Consulter, DME, Covid-19) (Infirmier, Ajouter, DME, Urgence) (Infirmier, Ajouter, DME, Covid-19)

Tableau 10: La relation Permission

Les autres Concepts (Règles) d'ORBAC seront créés par l'autorité des attributs :

- La relation Habilite (Utilisateur, Rôle) : Est vérifiée lors de la création d'un utilisateur en lui attribuer un rôle.
- La relation Définit (Sujet, Action, Objet, Contexte) : Est vérifiée lors de la création d'une nouvelle règle d'autorisation pour un utilisateur.

Les autorisations concrètes Est_permis seront automatiquement vérifiées si les concepts précédents sont vérifiés. (Tous ce qui est non permit est interdit).

2.2 Chiffrement des données CP-ABE

Dans notre système chaque employé utilisateur a ses propres attributs qui dépend d'ORBAC, l'univers d'attribut est défini comme étant : {*Médecin*, *Infirmier* (Son Rôle), *Urgence*, *Covid-19* (Contexte où il a des autorisation), *Chirurgie*, *Cardiologie*, *Maternité*, *Pédiatrie*, *Radiologie* (Service où il travaille)}.

L'autorité des attributs initialise pour chaque employé les clés publique et principales et génère la clé secrète qui est une combinaison d'un ensemble d'attributs.

CP-ABE est implémenté pour permettre au créateur de données (professionnel de santé) de définir une politique d'accès pour le chiffrement de ses données ; la politique est sous forme d'un ensemble d'attributs. Seuls les utilisateurs qui satisfont la politique d'accès ont l'autorisation qui lui permettre de déchiffrer ces données.

3. Etude conceptuelle

3.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un diagramme UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

Il permet d'analyser et d'organiser les besoins des utilisateurs par rapport au système.

Dans notre cas nous avons 2 types d'utilisateurs :

- Professionnel de santé : Des personnes qui jouent des différents rôles dans l'organisation (hôpital) et accèdent aux dossiers des patients.
- Admin : responsable de la gestion des attributs (utilisateurs, rôles, autorisations, …) des utilisateurs et des clés.

Ces acteurs peuvent établir différentes fonctionnalités comme il est illustré dans le diagramme de cas d'utilisation suivant :

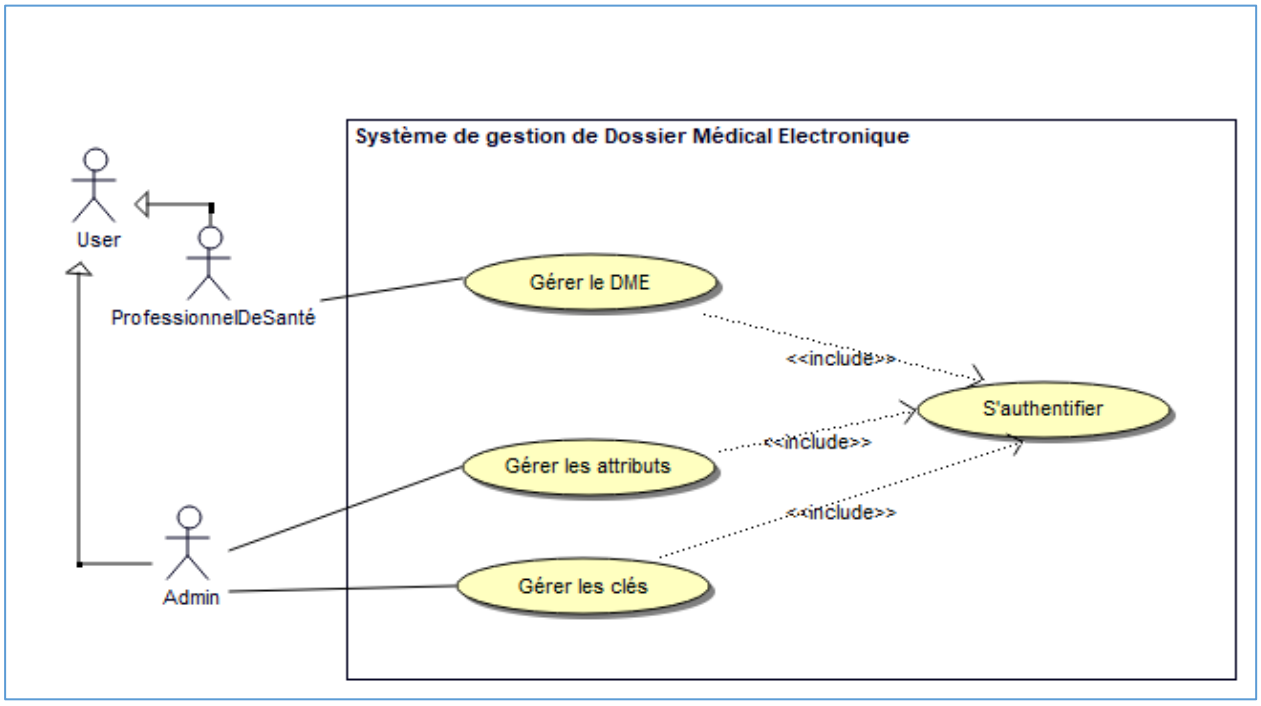


Figure 23: Diagramme de cas d'utilisation globale

Acteurs	Cas d'utilisation	Description
Tous les utilisateurs	S'authentifier	L'authentification est obligatoire pour pouvoir accéder au système.
Les professionnels de santé	Gérer le DME	Les médecins et les infirmiers peuvent rechercher, ajouter, modifier les données des patients.
Admin	Gérer les attributs	L'administrateur est chargé d'établir les rôles et les permissions aux utilisateurs.
	Gérer les clés	Génération des clés de chiffrement et de déchiffrement.

Tableau 11: Description des cas d'utilisation du diagramme globale

Maintenant nous allons détaillés les fonctionnalités suivantes :

- Gérer le DME
- Gérer les clés
- Gérer les attributs

3.1.1 Gérer le DME

La gestion de DME englobe toutes les fonctionnalités qui touchent les données des patients.

Les professionnels de santé tel que le médecin et l'infirmier sont les acteurs qui affecte des actions sur le DME.

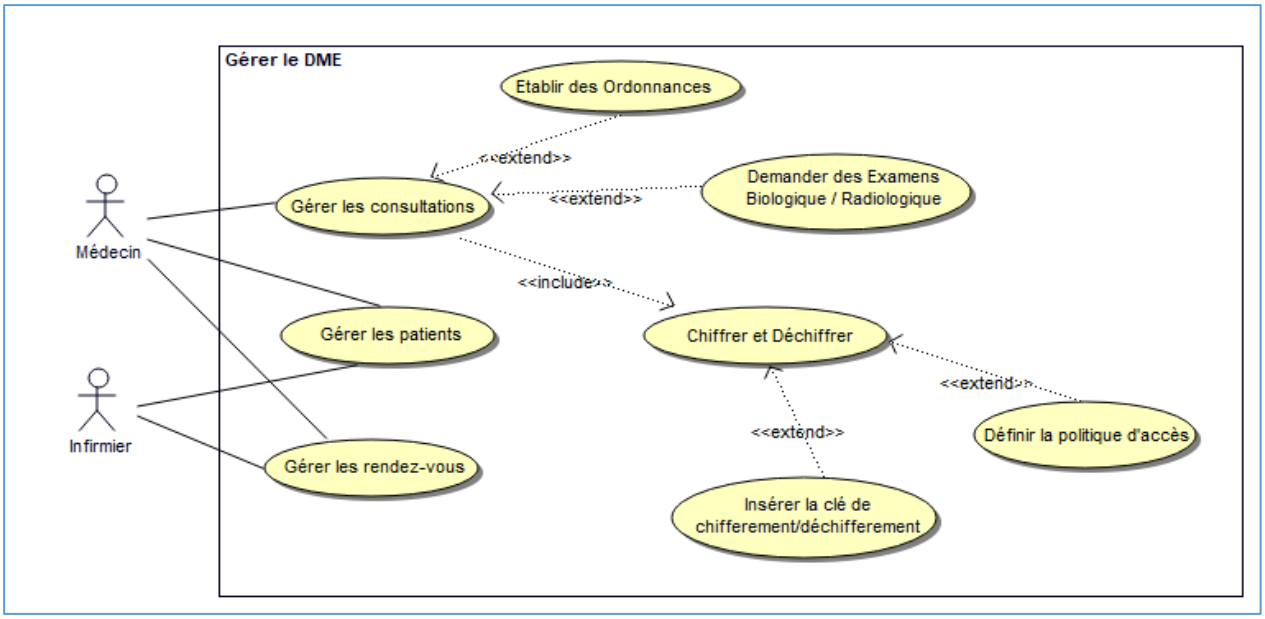


Figure 24: Diagramme de cas d'utilisation "Gérer le DME"

Acteurs	Cas d'utilisation	Description	
Médecin	Gérer les consultations	• Ajouter une consultation pour un patient et établir une ordonnance ou demander des examens radiologique ou biologique si c'est nécessaire ; • Consulter la liste des consultations.	**Lors de l'utilisation de DME par les professionnels de santé, peuvent :** **- Déchiffrer les données à consulter ou modifier en insérant la clé de déchiffrement,** **- Chiffrer les données ajoutés en insérant la clé publique et en définissant la politique d'accès**
Médecin & l'infirmier	Gérer les patients	• Ajouter un patient, • Consulter la liste des patients, • Consulter les détails d'un patient.	
	Gérer les rendez-vous	• Ajouter un rendez-vous, • Consulter la liste des rendez-vous.	

Tableau 12: Description des cas d'utilisation 'Gérer le DME'.

3.1.2 Gérer les attributs

C'est l'administrateur qui est chargé de gérer l'ensemble des attributs : employés, utilisateurs et règle d'autorisations.

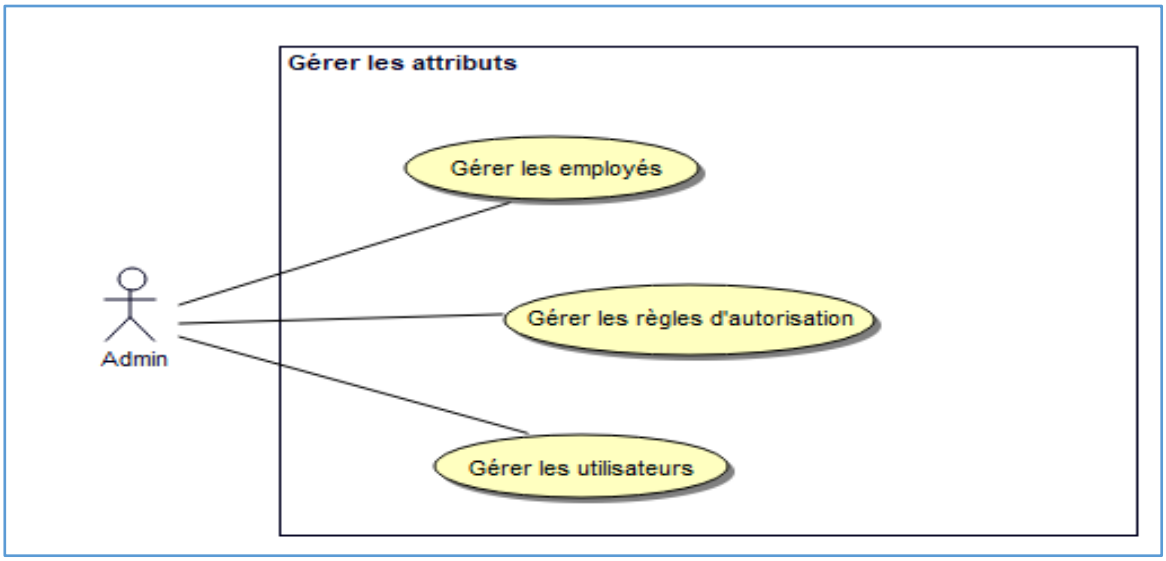


Figure 25: Diagramme de cas d'utilisation "Gérer les attributs"

Acteurs	Cas d'utilisation	Description
Admin	Gérer les règles d'autorisations	• Ajouter/Supprimer une autorisation, • Consulter la liste des autorisations.
	Gérer les employés	• Ajouter/Modifier/Supprimer un employé, • Consulter la liste des employés.
	Gérer les utilisateurs	• Ajouter un utilisateur en lui donnant un username, password et un rôle, • Consulter la liste des utilisateurs, • Supprimer un utilisateur.

Tableau 13: Description des cas d'utilisation " Gérer les attributs".

3.1.3 Gérer les clés

L'administrateur joue le rôle d'autorité des attributs; c'est lui qui définit les clés de chiffrement PK et les clés de déchiffrement SK.

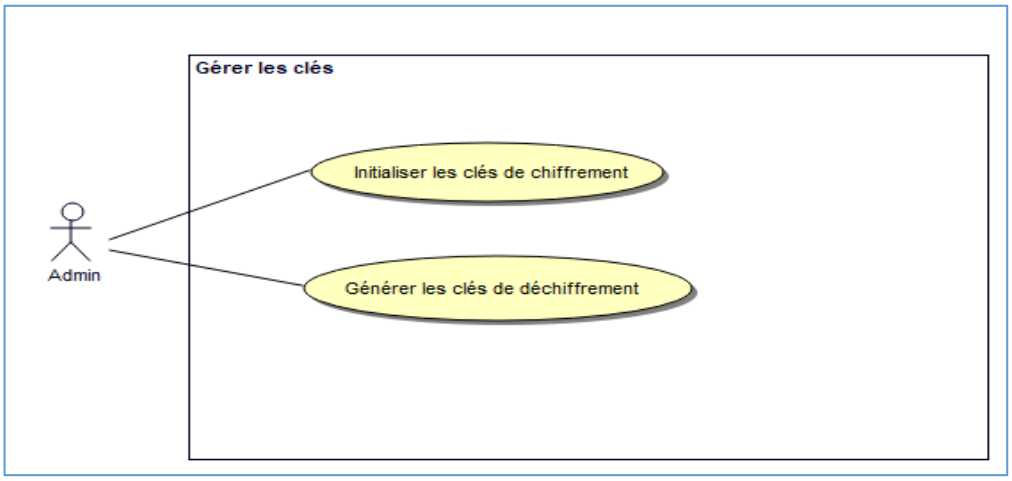


Figure 26: Diagramme de cas d'utilisation "Gérer les clés"

Acteurs	Cas d'utilisation	Description
Admin	Initialiser les clés de chiffrement	Définir les deux clés : – La clé public PK de chiffrement – La clé secrète principale MK pour la génération des clés secrète de déchiffrement
	Génération des clés de déchiffrement	Générer la clé secrète SK de déchiffrement liée à l'ensemble d'attributs dont l'employé a.

Tableau 14: Description des cas d'utilisation "Gérer les clés".

3.2 Diagrammes de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

Nous illustrons les diagrammes de séquence suivants :

- Authentification
- Ajouter un nouvel utilisateur
- Gestion des règles d'autorisations
- Génération des clés
- Chiffrement
- Déchiffrement

3.2.1 Authentification

Chaque utilisateur (professionnel de santé ou bien l'admin) doit s'authentifier en utilisant son email et mot de passe pour pouvoir accéder à son espace où il est permis de réaliser les différentes autorisations (des actions sur des objets).

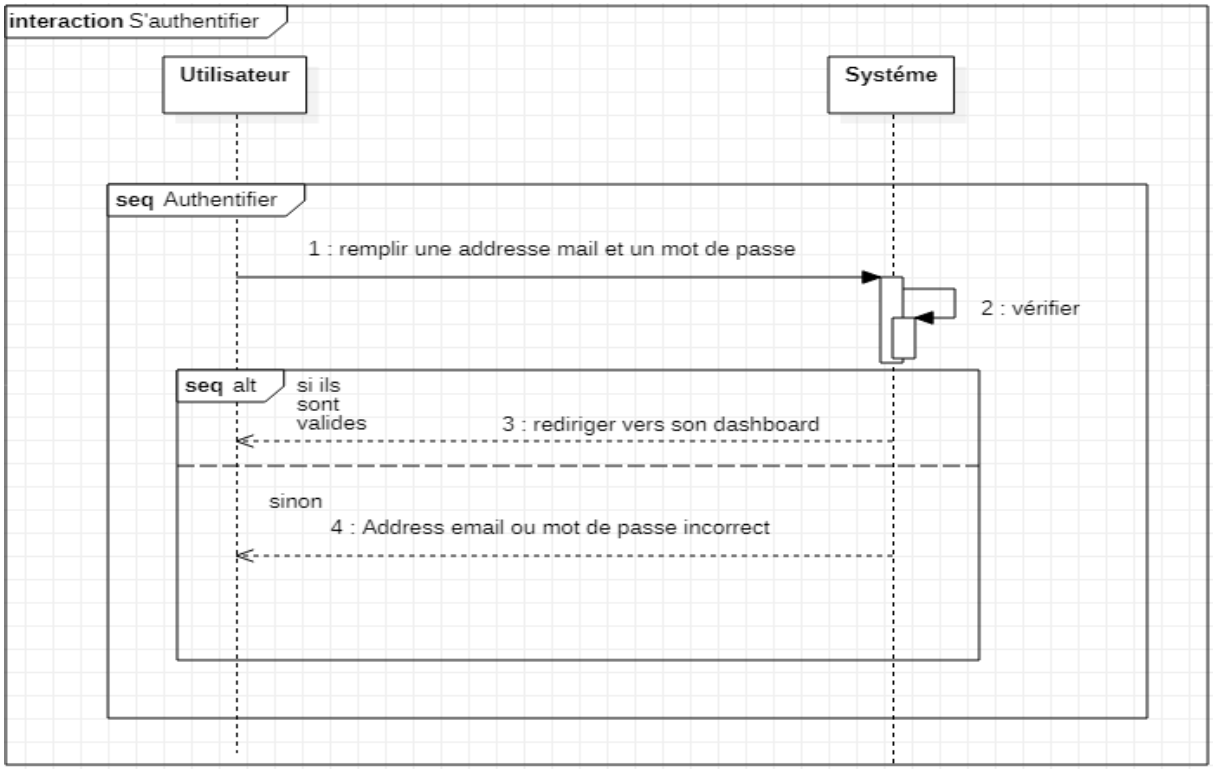


Figure 27: Diagramme de séquence "Authentification"

3.2.2 Ajouter un nouvel utilisateur

L'administrateur est le seul qui peut créer un espace pour n'importe quel employé afin de lui attribuer son rôle pour qu'il puisse accéder au système.

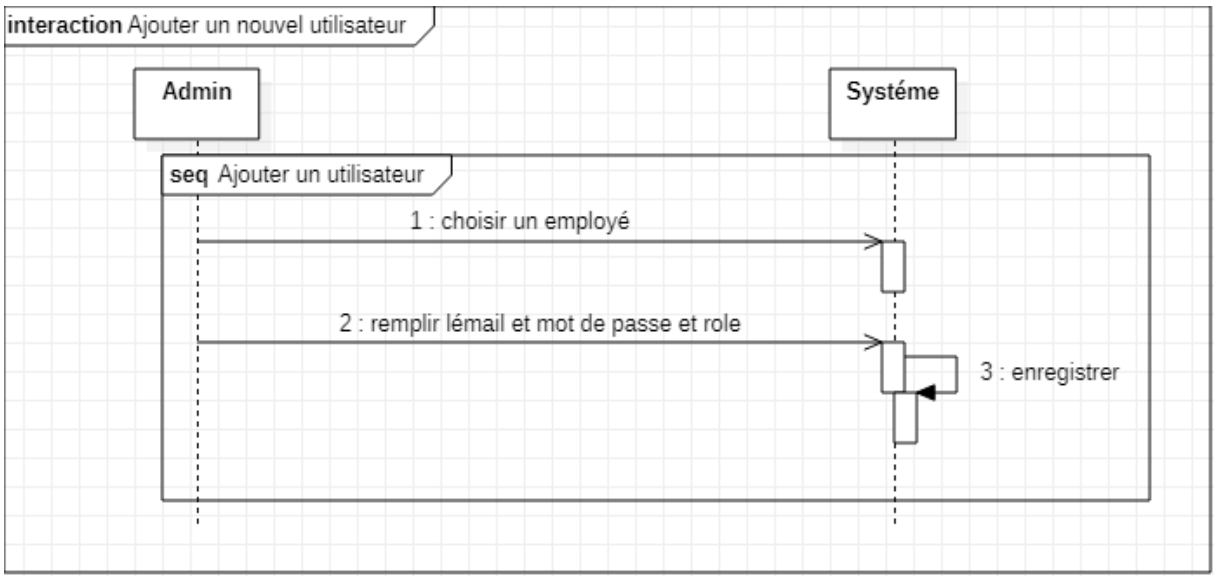


Figure 28: Diagramme de séquence "Ajouter un utilisateur"

3.2.3 Ajouter une autorisation

La gestion des attributs est chargée par l'administrateur donc c'est le seul qui peut Ajouter une autorisation pour un employé qui est déjà un utilisateur dans le système.

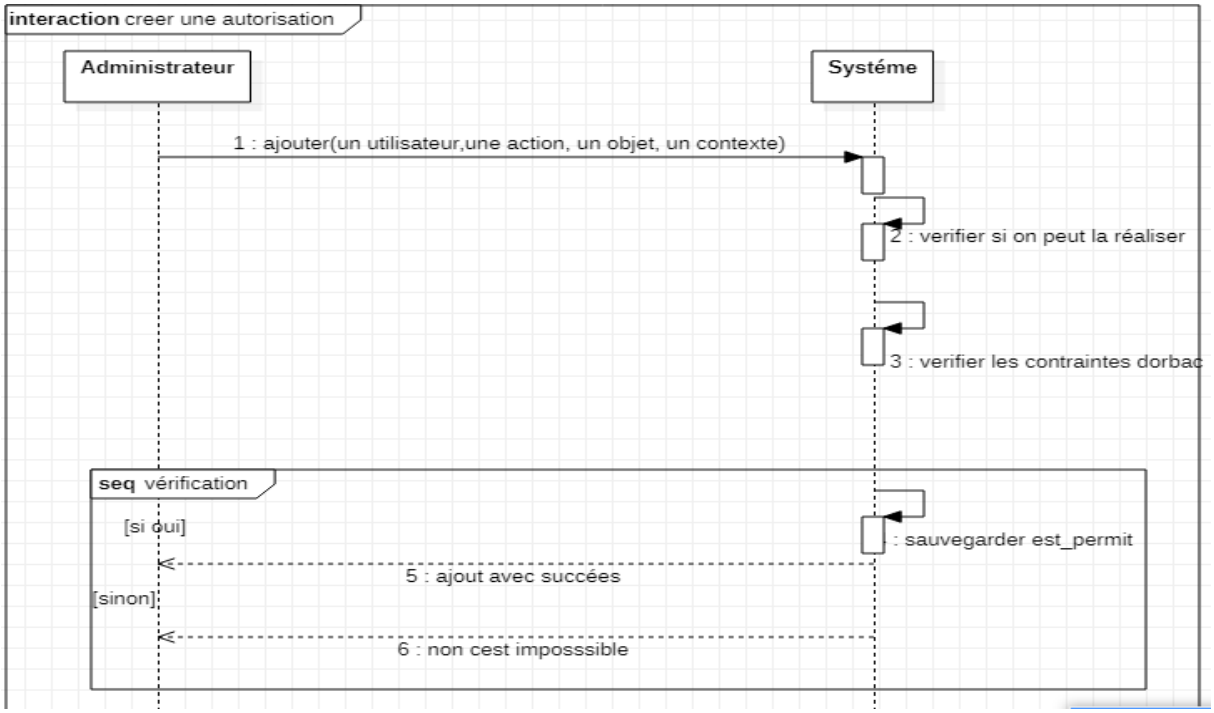


Figure 29: Diagramme de séquence "Ajouter une autorisation"

3.2.4 La génération des clés

Il faut d'abord initialiser la clé publique et principale pour le chiffrement puis générer la clé secrète de déchiffrement comme illustrer dans les figures ci-dessous.

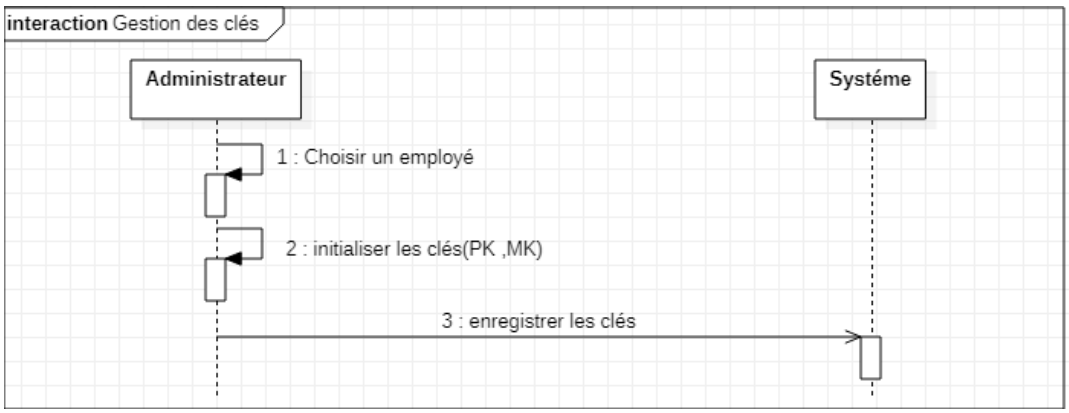


Figure 30: Diagramme de séquence "Initialiser les clés Principale MK et publique PK"

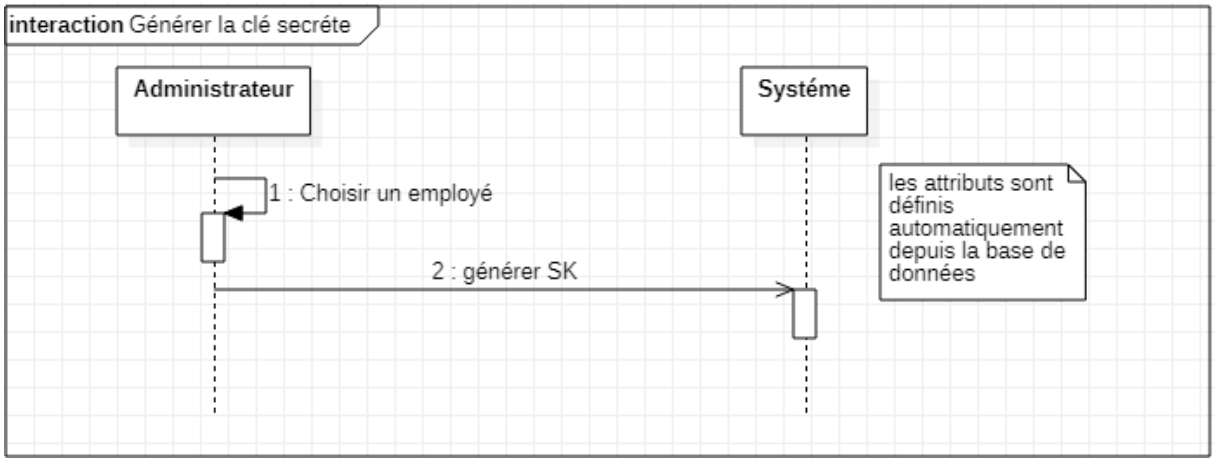


Figure 31: Diagramme de séquence "Générer la clé secrète SK"

3.2.5 Chiffrement

Le médecin créateur de la donnée peut crypter sa donnée, et pour cela il faut qu'il définir la politique d'accès à cette dernière. Et après chaque cryptage des autorisations seront ajoutés aux utilisateurs qui ont un ensemble d'attribut (clé secrète SK) inclus dans la politique choisie par le créateur. Voici le détail dans la figure ci-dessous :

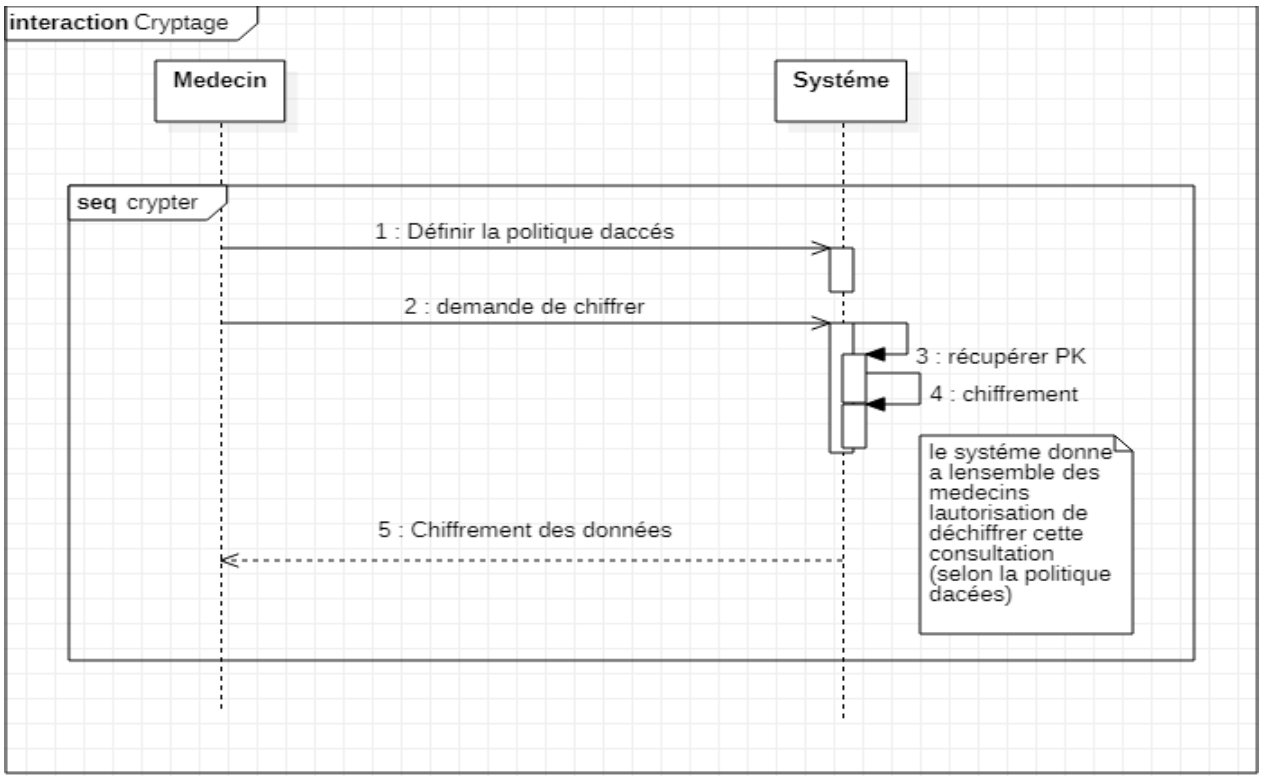


Figure 32: Diagramme de séquence "Cryptage"

3.2.6 Déchiffrement

Cette étape permet aux utilisateurs autorisés de décrypter et consulter les données en clair.

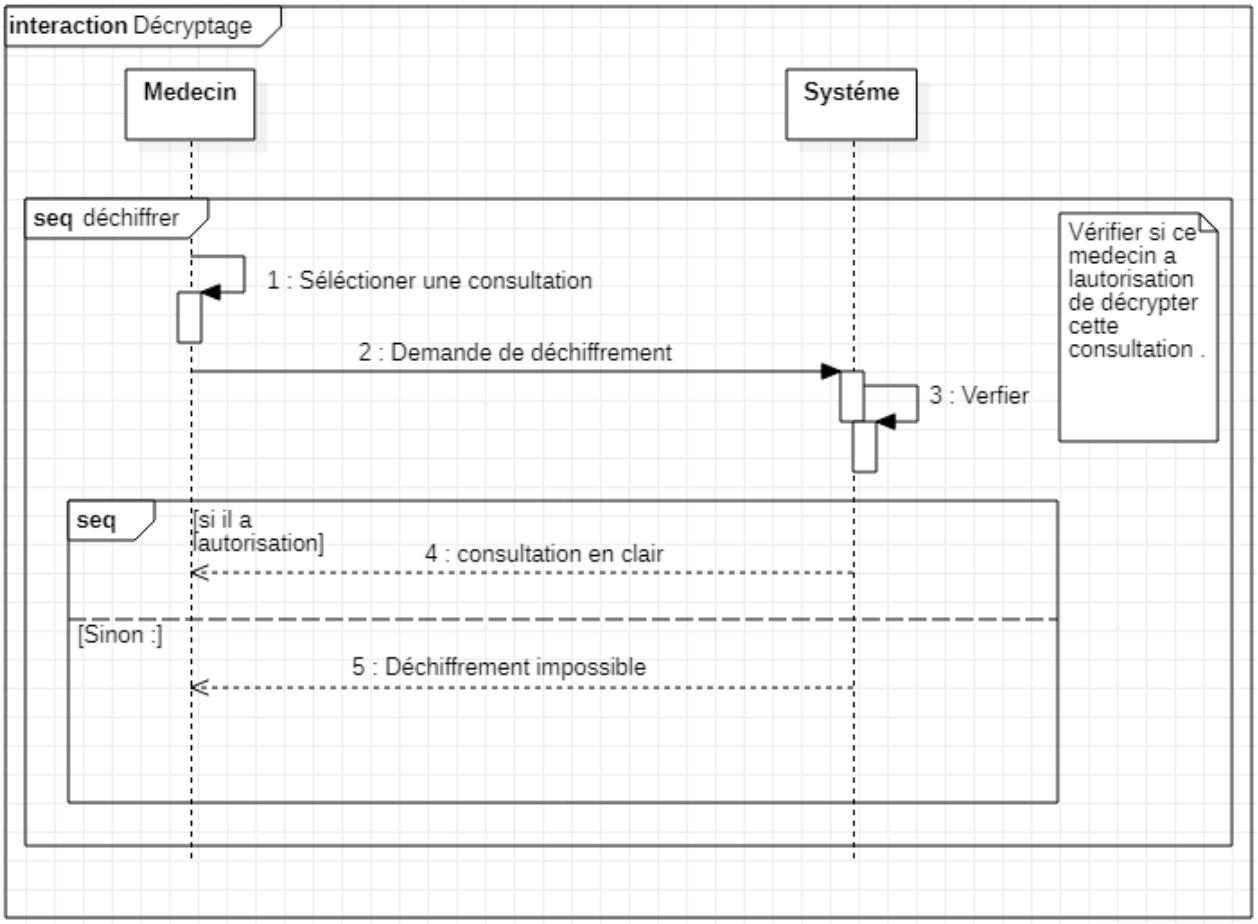


Figure 33: Diagramme de séquence "Décryptage"

4. Conclusion

Dans cette partie, nous avons modélisé l'approche utilisé dans la couche de sécurité pour notre système de gestion de dossier patient électronique.

L'approche se repose sur le contrôle d'accès à base d'organisation ORBAC et le chiffrement par attributs CP-ABE.

Nous passons dans la partie suivante à l'implémentation et mise en œuvre de cette approche.

Chapitre 5 Réalisation

1. Introduction

Ce chapitre a pour objectif majeur la présentation de la dernière partie de ce travail, C'est la phase qui traite la partie réalisation et mise en œuvre de la couche de sécurité pour le système de gestion de dossier patient électronique qui contient les différents composants décrits au niveau des chapitres précédents.

Dans un premier temps, on présente l'environnement de développement. Ensuite, on décrit le travail réalisé en détaillant quelques fonctionnalités réalisées par des captures d'écrans des principales interfaces graphiques.

2. Environnement de développement

Dans cette section on présente l'environnement de développement qui se réfère aux outils, langage de programmation et aux applications qu'on a installés et utilisés.

2.1 Les langages de programmation

Pour l'implémentation de notre SGDP avec la couche de sécurité proposée, nous avons utilisé le langage PHP et les technologies de web : HTML, CSS et Java Script.

2.2 Laravel Framework

Laravel est un Framework PHP open source basé sur le pattern MVC (Model, View, Controller) crée par Taylor Otwel. Publié en 2011.

Ce Framework a connu un grand succès et est aujourd'hui parmi les plus populaires et les plus utilisés.

Laravel inclut un nombre important de spécificités et outils qui facilitent le développement d'applications web.

Voici une brève explication des plus intéressants :

- **Composer** est un outil de gestion de dépendances. Il permet de contrôler l'installation et les mise à jour des dépendances que l'on peut gérer par le fichier composer.json. C'est un fichier texte ou l'on inscrit toutes les dépendances dont on a besoin.
- **Artisan** est un script PHP qu'il est possible d'exécuter en ligne de commande. Il s'occupe des migrations, affiche les routes, permet de vider le cache ou peut également pré-remplir la base de données.
- Les **migrations** fonctionnent comme un système de contrôle de version pour base de données. Il s'agit en quelque sorte d'un schéma, écrit en PHP, qui permet de générer la base de données à partir d'une commande Artisan. La base de données est ainsi rendue portable et peut facilement être partagée d'un environnement à l'autre.
- Le **MVC** est un design pattern qui a pour but de séparer l'application en trois parties : Model-View-Controller. Le contrôleur gère la partie logique, la vue s'occupe de l'affichage et le model interagit avec la base de données en passant par Eloquent ORM.
- **Eloquent** est un ORM, autrement dit Object-Relational-Mapping, qui mappe nos objets PHP aux tables de la base de données. Ainsi l'interrogation et la manipulation sont simplifiées.

- Le **routing** permet d'associer une URL à une méthode particulière du contrôleur
- Un outil de **validation** puissant qui permet de valider les entrées d'une requête http.
- **Blade** est un moteur de Template qui aide à structurer les views (vues) de l'application. En plus de cela, il permet aussi de créer de façon uniforme divers éléments HTML comme par exemple les formulaires.
- Les **middlewares** permettent de filtrer les requêtes HTTP. Par exemple, ils permettent de gérer la langue de l'application ou encore de vérifier si l'utilisateur est authentifié ou non. Ils peuvent être utilisés à chaque chargement de page ou seulement à certains moments précis.

2.3 Bootstrap

Bootstrap est un Framework Web open-source gratuit et ouvert pour la conception de sites Web et d'applications Web. Il contient des modèles de conception basés sur HTML et CSS pour la typographie, les formulaires, les boutons, la navigation et d'autres composants d'interface, ainsi que des extensions JavaScript optionnelles. Contrairement à de nombreux Framework Web, il se préoccupe uniquement du développement frontal.

2.4 MySQL SGDB

MySQL est le deuxième système de gestion de base de données le plus populaire. Il s'agit d'une base de données relationnelle open-source.

Elle est également développée par Oracle Corporation, anciennement par MySQL AB et Sun Microsystems. La première version a été distribuée en 1995. MySQL est implémenté en C et C++.

Il supporte également une grande variété de langages, dont PHP

2.5 WampsSaerver

Est une plate-forme de développement web sous Windows pour des applications web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL.

3. Description de l'implémentation de la couche de sécurité

L'utilisation du Framework Laravel facilite la modélisation et l'implémentation des relations et des entités d'OrBAC ainsi que les fonctions de chiffrement et de déchiffrement de CP-ABE en appliquant l'architecture MVC comme indique la figure 34

- Le contrôleur est responsable de la logique de contrôle de l'application, il sert à gérer les demandes des utilisateurs et à récupérer des données, en tirant parti des modèles,
- Les modèles servent à interagir avec la base de données et à récupérer les informations des objets,
- Les vues pour afficher les pages demandées.

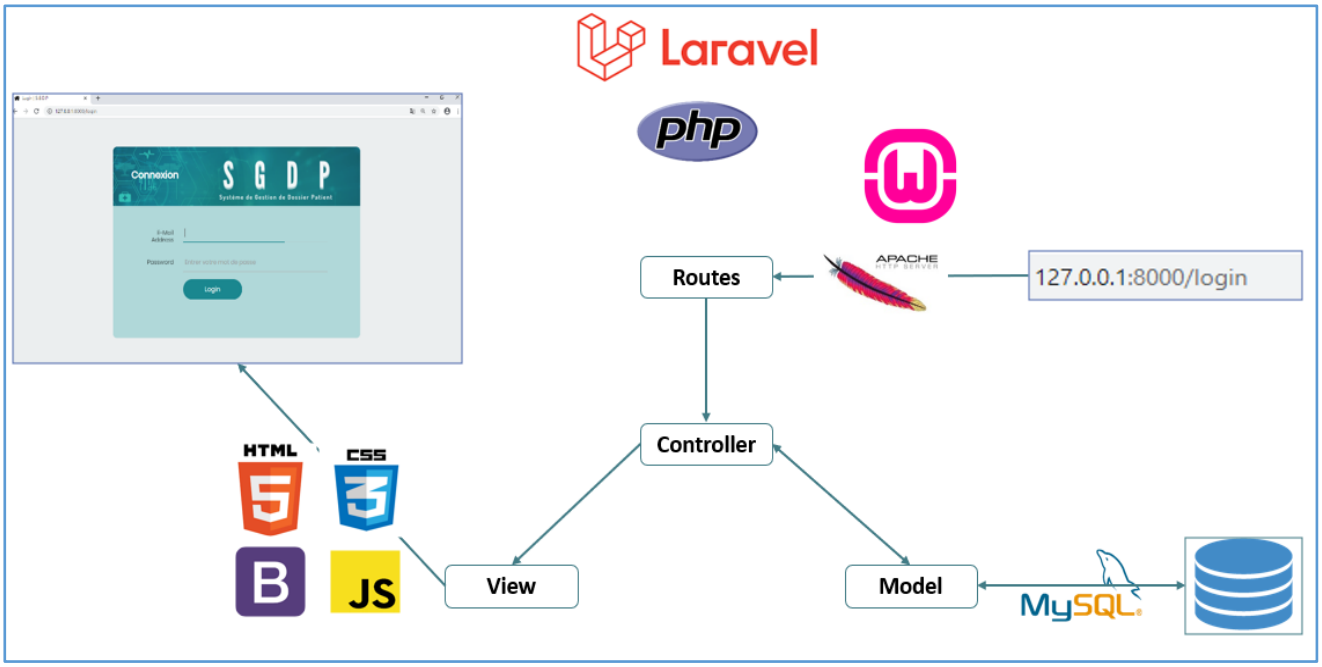
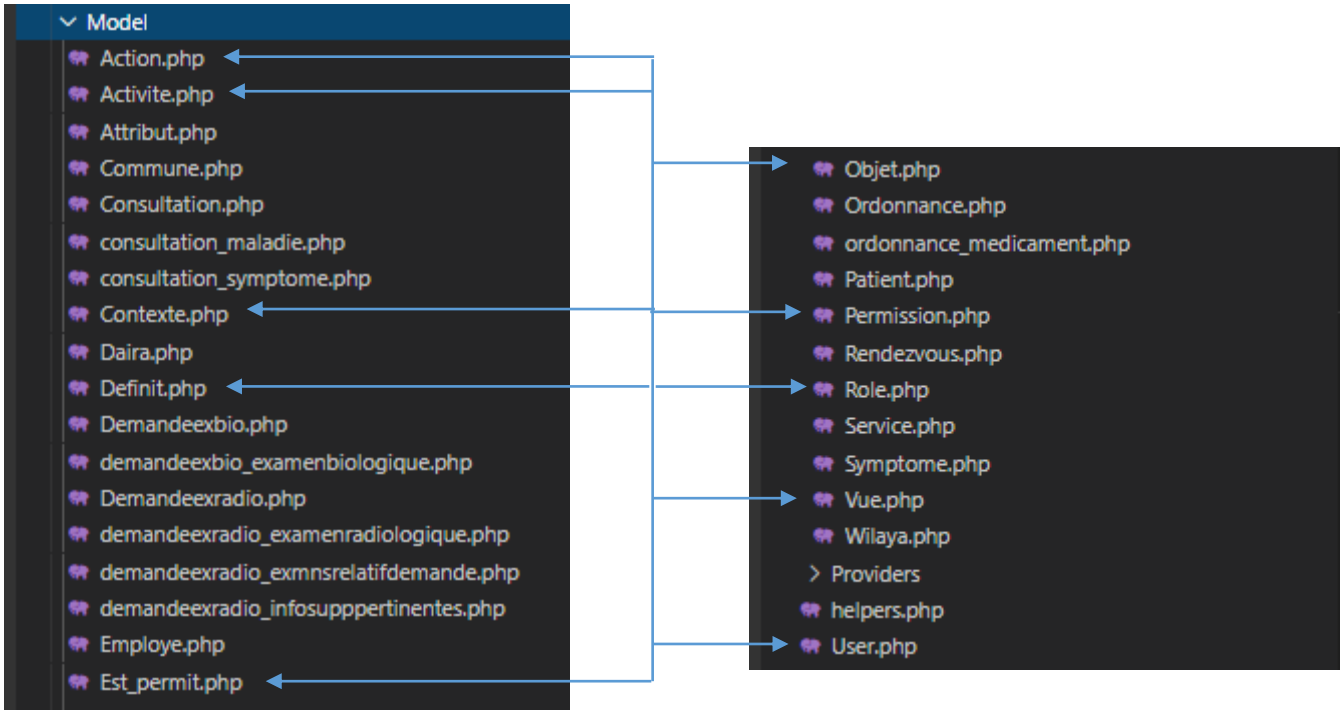


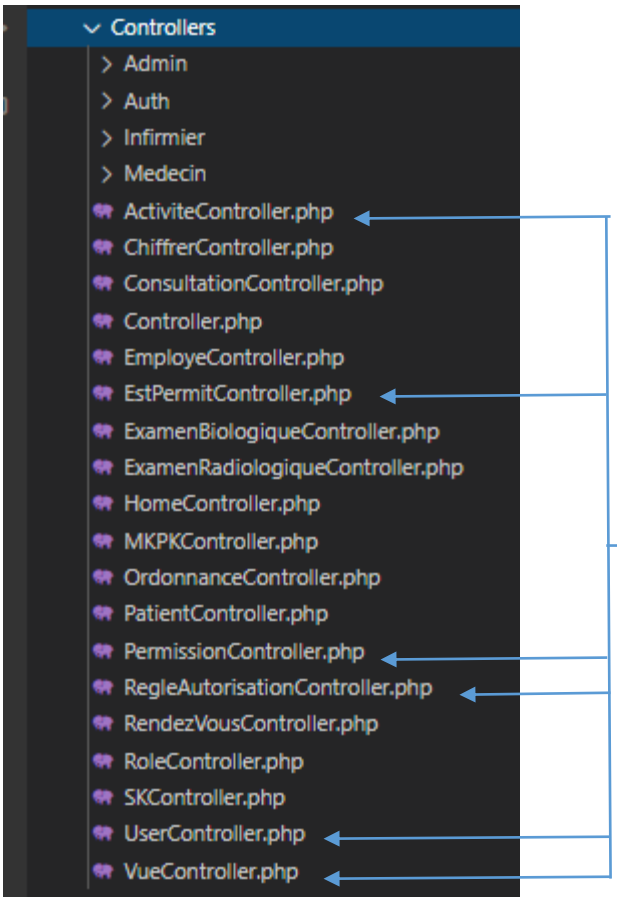
Figure 34: L'architecture MVC

3.1 OrBAC

- Pour chaque entité (*Sujet*, *Rôle*, *Objet*, *Vue*, *Action*, *Activité*, *Contexte)* et chaque relation (*Permission*, *Est_permis*, *Définit)*, nous avons implémenté un modèle :



- Et pour chaque relation on a implémenté un contrôleur :



- ActiviteController représente :
  La relation *Considère* (action, activité).
- EstPermitController :
  La relation *Est_permis* (user, action, objet).
- PermissionController :
  La relation *Permission (rôle, activité, vue, contexte).*
- Règled'autorisationController :
  La relation *Définit (user, action, objet).*
- VueController :
  La relation *Utilise (objet, vue).*
- UserController :

Dans ces contrôleurs nous avons fait appel à des fonctions qui se trouve dans un Helper, les fonctions permettent de contrôler et assurer le bien fonctionnement d'OrBAC.

- La fonction **maj** est exécutée après chaque ajout d'une nouvelle règle d'autorisation. Elle vérifier si la relation Définit *(user, action, objet)* est correcte et aussi si les autres relations (Habilite, Considère, Utilise) existent déjà. Si tout est vérifié une permission Est_Permit sera ajoutée dans la tables est_permits.

```
function maj($id){  //id de définit
    $definit = Definit::FindOrFail($id);   //on a la relation definit(user,action,objet,contexte )
                    //on va vérifier tous les realtions précedentes  pour confirmer la relation est_permit
    $id_user = $definit->user->id;
    $action = $definit->action->nom;
    $objet = $definit->objet->nom;
    $nom = $objet.' '.$action;
    $s= $objet.'?'.$action;
    $slug =  Str::replaceArray('?',['.'], $s);
    $id_role = $definit->user->role->id;    //la relation habilite est vérifiée
    $id_activite = $definit->action->activite->id;   //la relation considère est vérifiée
    $id_vue = $definit->objet->vue->id;     //la relation utilise est vérifiée
    $id_contexte = $definit->contexte->id;    //le contexte est vérifiée
    $est_permit = DB::table('permissions')   ///pour vérifier la relation permission
    ->where('role_id', '=', $id_role)
    ->where('activite_id', '=', $id_activite)
    ->where('vue_id', '=', $id_vue)
    ->where('contexte_id', '=', $id_contexte)
    ->first();
    if (is_null($est_permit)) {  // la relation permission(role,activité,vue,contexte ) n'est pas vérifiée
        echo 'you can t do this';
    } else {   //la relation permission est vérifiée donc on va remplir la bd est_permits
        if(is_null(DB::table('est_permits')  //n'existe pas déjà
                        ->where('user_id', '=', $id_user)
                        ->where('nom', '=', $nom)
                        ->where('slug', '=', $slug)
                        ->first() ) )
            {   DB::table('est_permits')->insert([
                    'user_id' => $id_user,
                    'nom' => $nom,
                    'slug' => $slug,
                ]);    } } }
```

Figure 35: La fonction maj

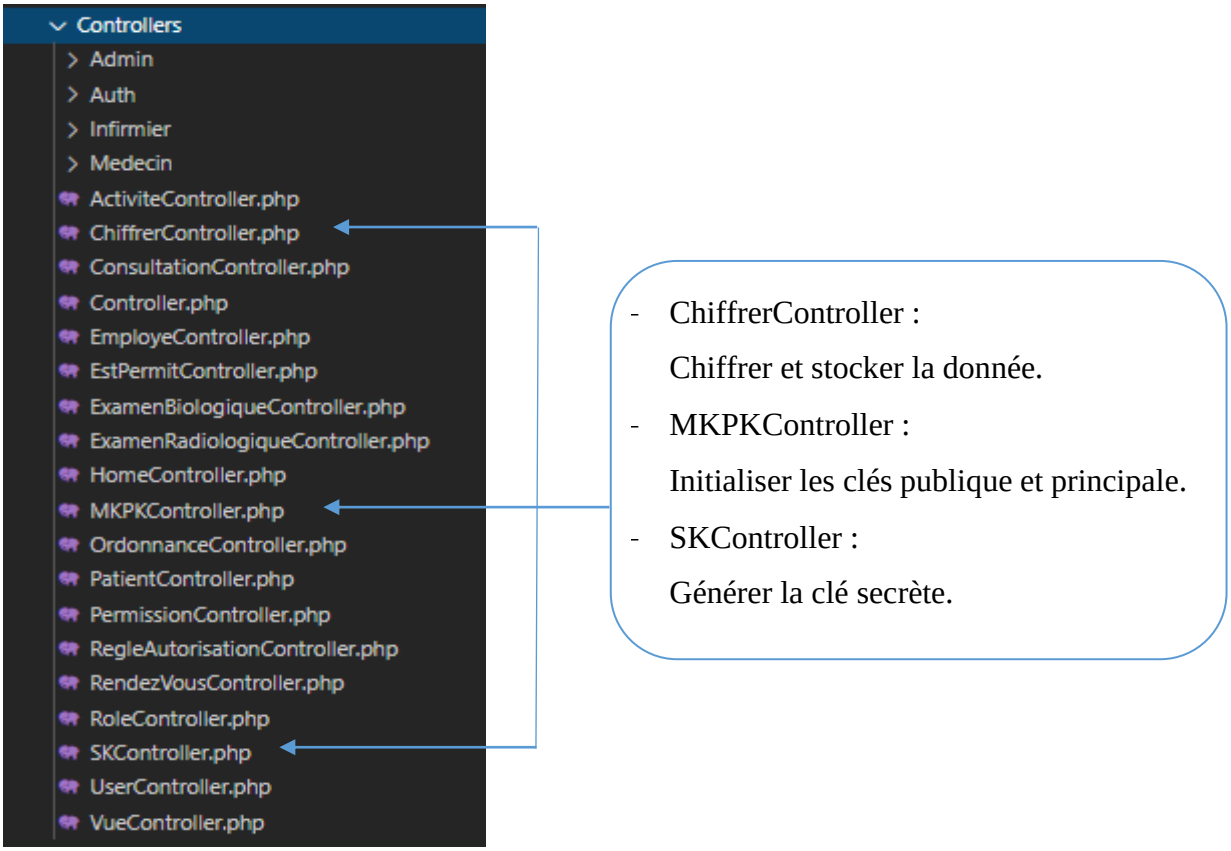
- La fonction **CanDo** vérifie si l'utilisateur avait une permission qui lui permet de faire une action sur un objet.

```
function CanDo($slug){
    $id_user = Auth::user()->id;
    if (is_null( DB::table("est_permits")
                    ->where('user_id', '=', $id_user)
                    ->where('slug', '=', $slug)
                    ->first() ) ){
                        abort(401, 'This action is unauthorized.');
                        return false;
    }else { return true;
} }
```

Figure 36: La fonction CanDo

3.2 CP-ABE

Pour la construction de chiffrement CP-ABE, nous avons implémenté des contrôleurs pour la génération des clés et le chiffrement.



Dans ces contrôleurs nous avons faire appel à des fonctions qui se trouvent dans le Helper.

- La fonction **IsOwner** : vérifie si l'utilisateur (professionnel de santé) est le créateur de la donnée ; signifie que c'est lui qui l'a créé.

```php
function IsOwnerConsultation($id){ //id of object
    $id_user = Auth::user()->id;
    $user = User::FindOrFail($id_user);
    $id_employe = $user->employe_id;
    $consultation = Consultation::FindOrFail($id);
    if ($consultation->employe_id == $id_employe){
        return true; //echo " user is owner ";
    }else{
        return false; //echo " user isn t owner";
}}
```

Figure 37: La fonction IsOwner

- La fonction **CanDecrypt** : vérifie si l 'utilisateur avait l'autorisation de décrypter.

```
function CanDecrypt($id){
    $nom ='consultation<decrypter>'.$id;
    $id_user = Auth::user()->id;
    if (is_null( DB::table("est_permits")
                    ->where('user_id', '=', $id_user)
                    ->where('nom', '=', $nom)
                    ->first() )
    ){  return false;}
    else {    return true;    }
}
```

Figure 38: La fonction CanDecrypt

- La fonction **generateKey** : initialise une clé aléatoire.

```
function generateKey(){
    $KeyLength =8;
    $str = "12afeljhrtsbvn98765)(HGQFEMLLDH/";
    $randstr = substr(str_shuffle($str), 0,$KeyLength);
    return $randstr;
}
```

Figure 39: La fonction generateKey

- La fonction **genereSK** : génère la clé secrète.

```
function genereSK($id){  //id employe
    $employe = Employe::FindOrFail($id);
    $mk = $employe->mk;   $service = $employe->service->nom;
    $user = DB::table("users")
                ->where('employe_id', '=', $id)
                ->first();
    $role = DB::table("roles")
                ->where('id', '=', $user->role_id)
                ->first();
    $role =$role->name;  $contextes="";
    $definits = DB::table("definits")
        ->where('user_id', '=', $user->id)
        ->get();
    foreach($definits as $definit){
        $contexte= DB::table("contextes")
                    ->where('id', '=', $definit->contexte_id)
                    ->first();
        $contexte =$contexte->nom;
        if(! Str::contains($contextes,$contexte)){
            if($contextes==""){  $contextes= $contextes.$contexte;}
            else{$contextes= $contextes.",".$contexte;}
        }
    }
    $attributs = array($role.",".$service.",".$contextes);
    $attributs = json_encode($attributs,JSON_UNESCAPED_UNICODE);
    $sk =$mk.$attributs;  $employe->sk = $sk;
    $employe->save();
```

Figure 40: La fonction genereSK

4. Présentation de l'application

Notre Application commence d'abord par l'authentification des utilisateurs.

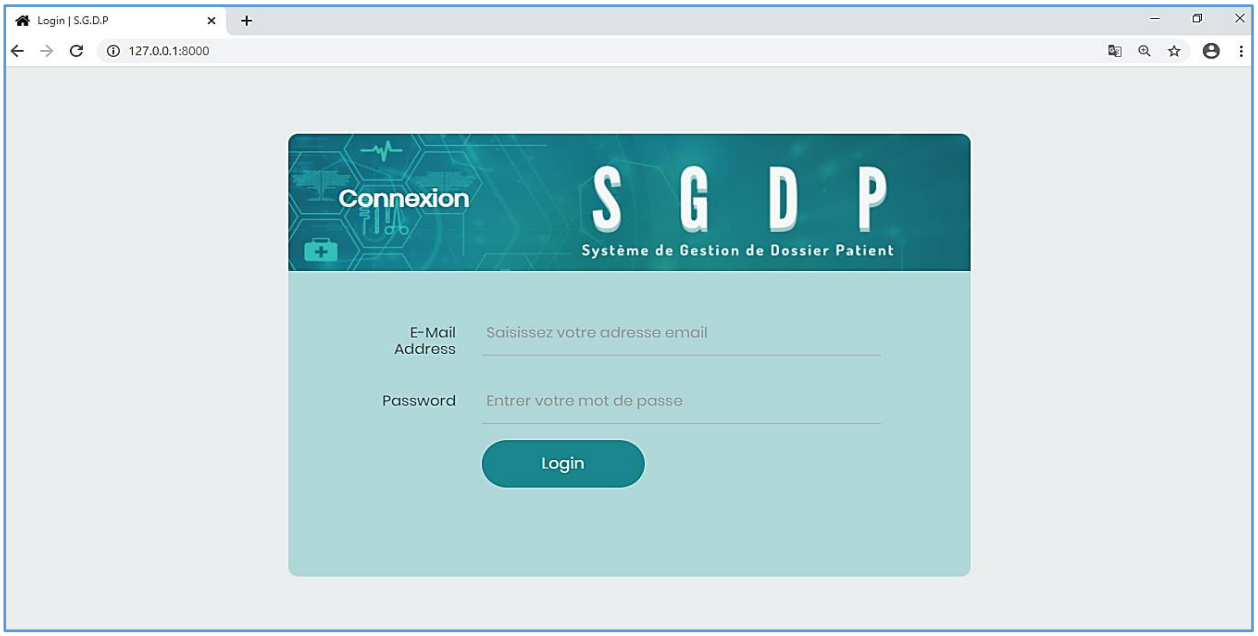


Figure 41: Interface d'authentification

L'authentification nous dirige vers deux espaces à l'aide d'un middleware qui vérifier le rôle de l'authentifiant

Nous pouvons dire que l'application est composée de deux parties :

- L'espace d'administrateur
- L'espace des professionnels de santé

4.1 Espace Admin

4.1.1 Afficher les paramètres d'OrBAC

Les paramètres d'OrBAC (rôle, vue, activité, permissions) sont prédéfinis

- **Vue :** représente la relation Utilise (objet, vue)
- **Activité :** représente la relation Considère (action, activité)
- **Permission :** représente la relation Permission (rôle, activité, vue, contexte)

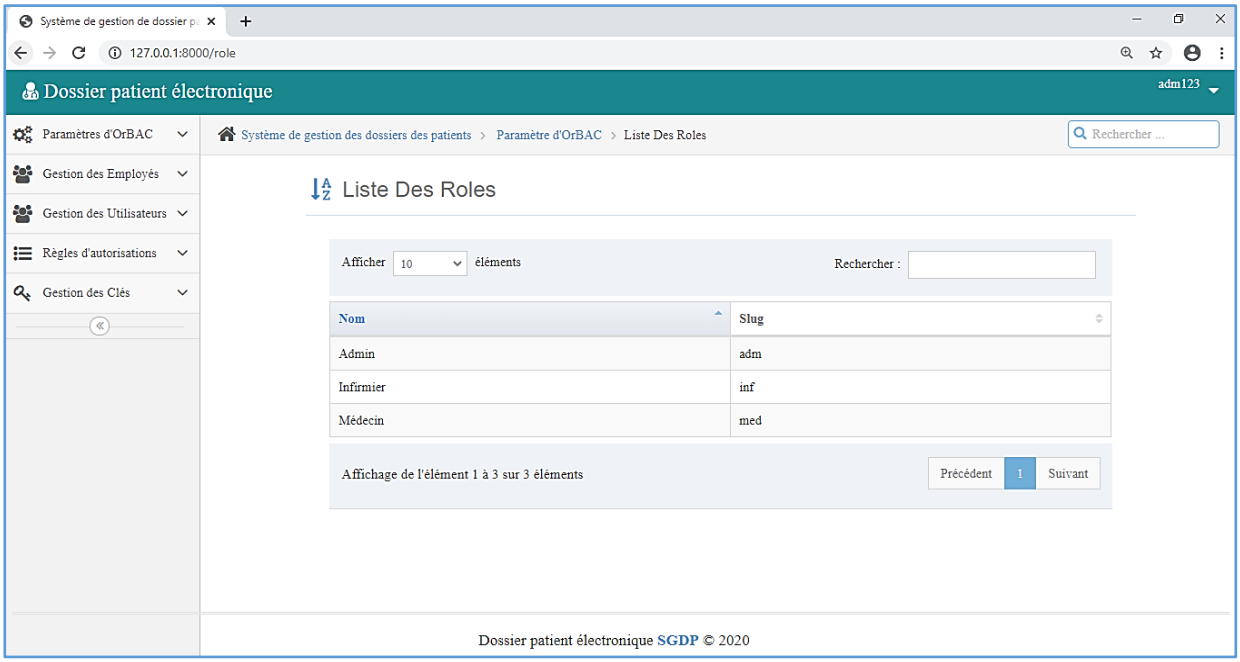


Figure 42: Liste des rôles

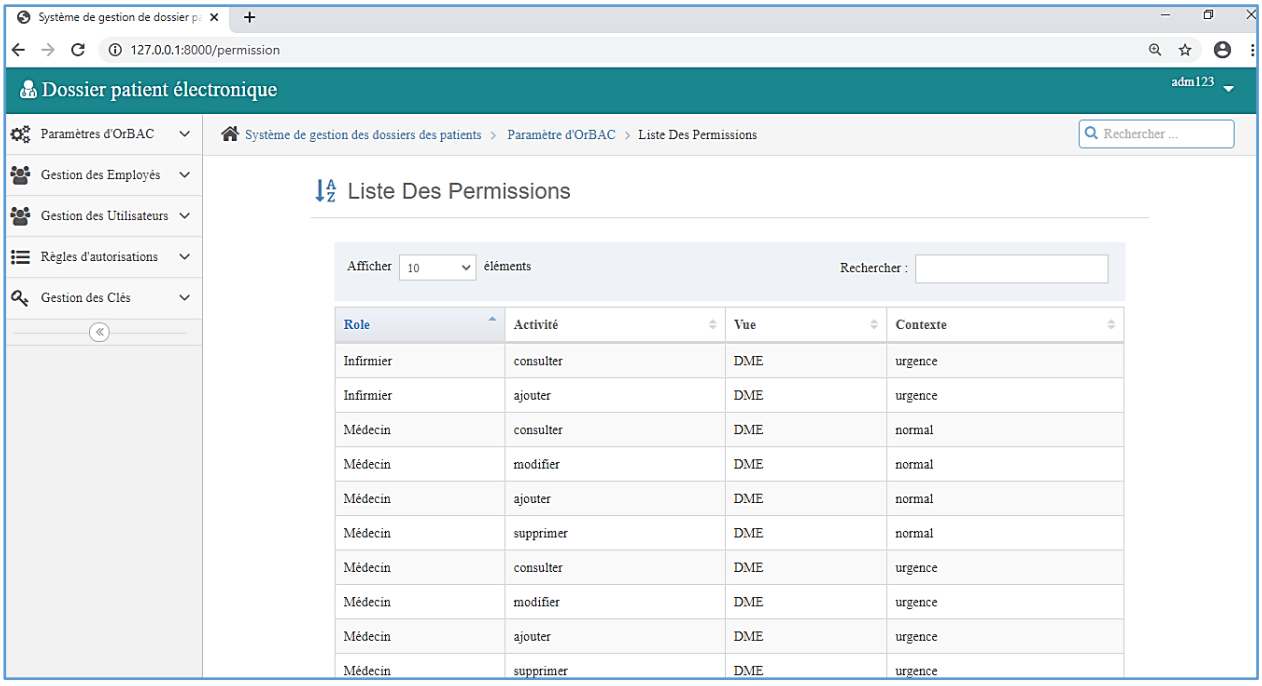


Figure 43: Liste des permissions

4.1.2 Gestion des utilisateurs et employés

Lors de la création d'un espace utilisateur pour un employé déjà ajouté dans system la relation d'OrBAC Habilite sera créée ; Habilite (utilisateur, rôle).

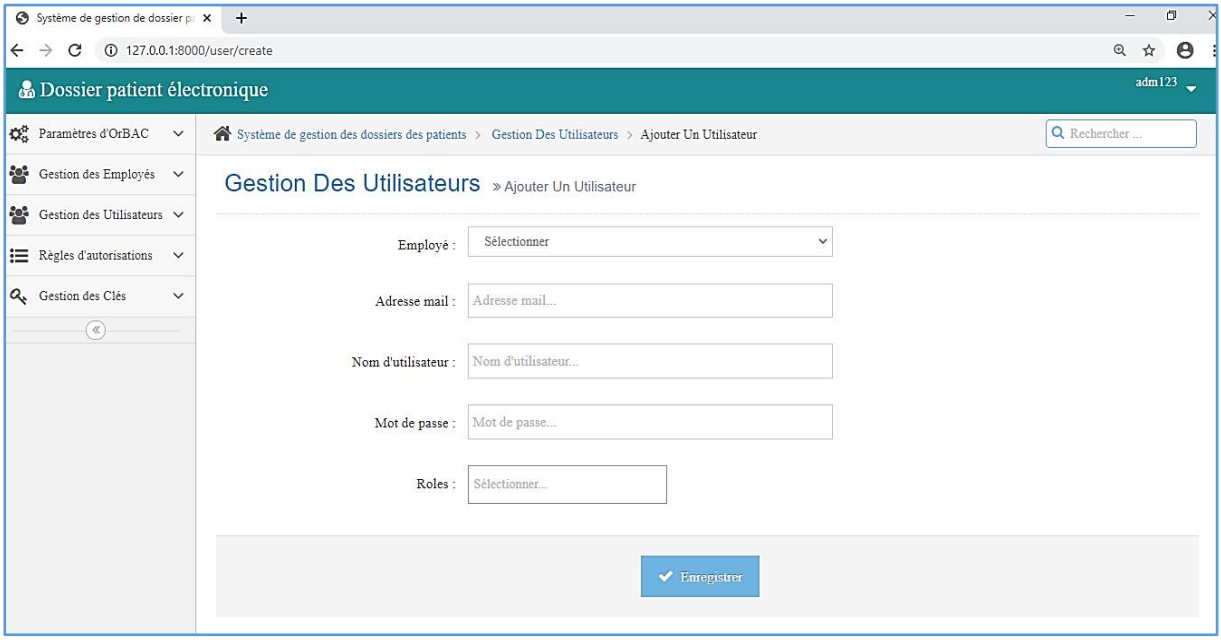


Figure 44: Ajouter un utilisateur

4.1.3 Règles d'autorisations

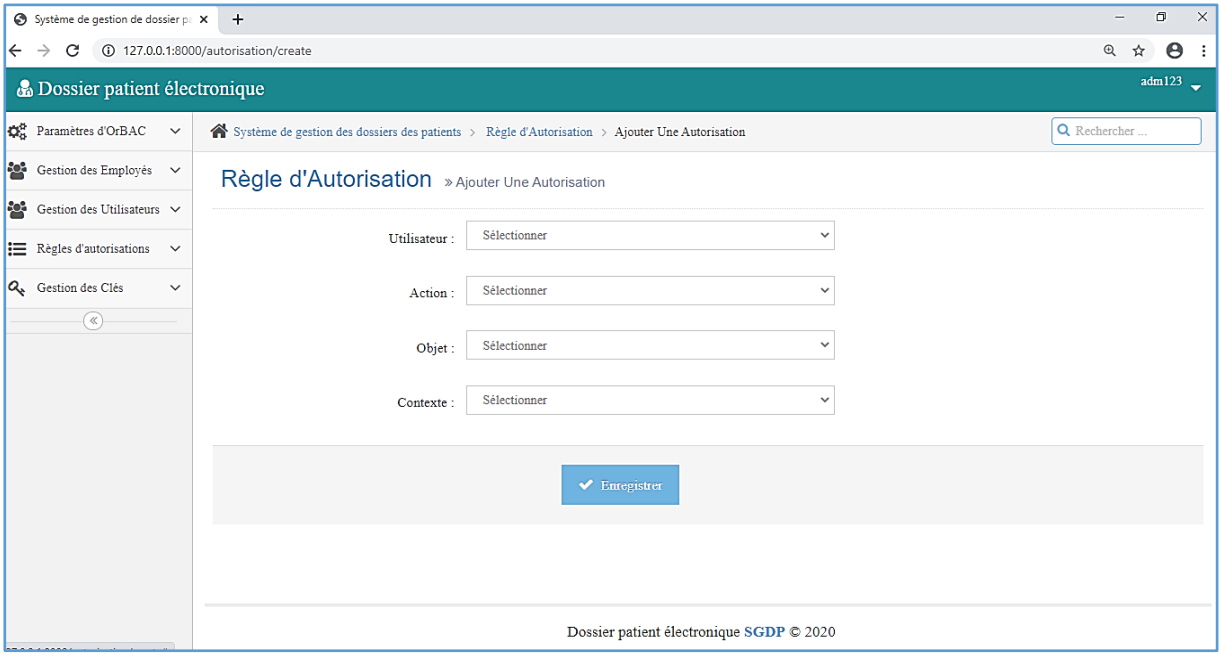


Figure 45: Ajouter une règle d'autorisation

4.1.4 La gestion des clés de CP-ABE

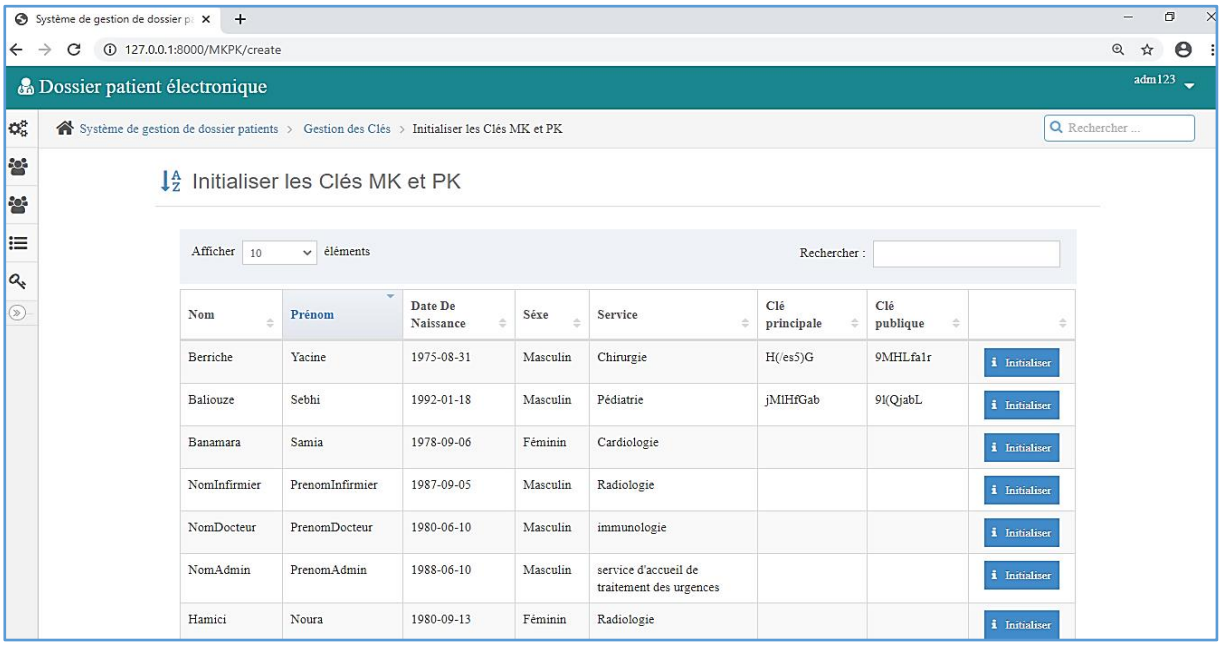


Figure 46: Initialiser les clés Publique PK et Principale MK

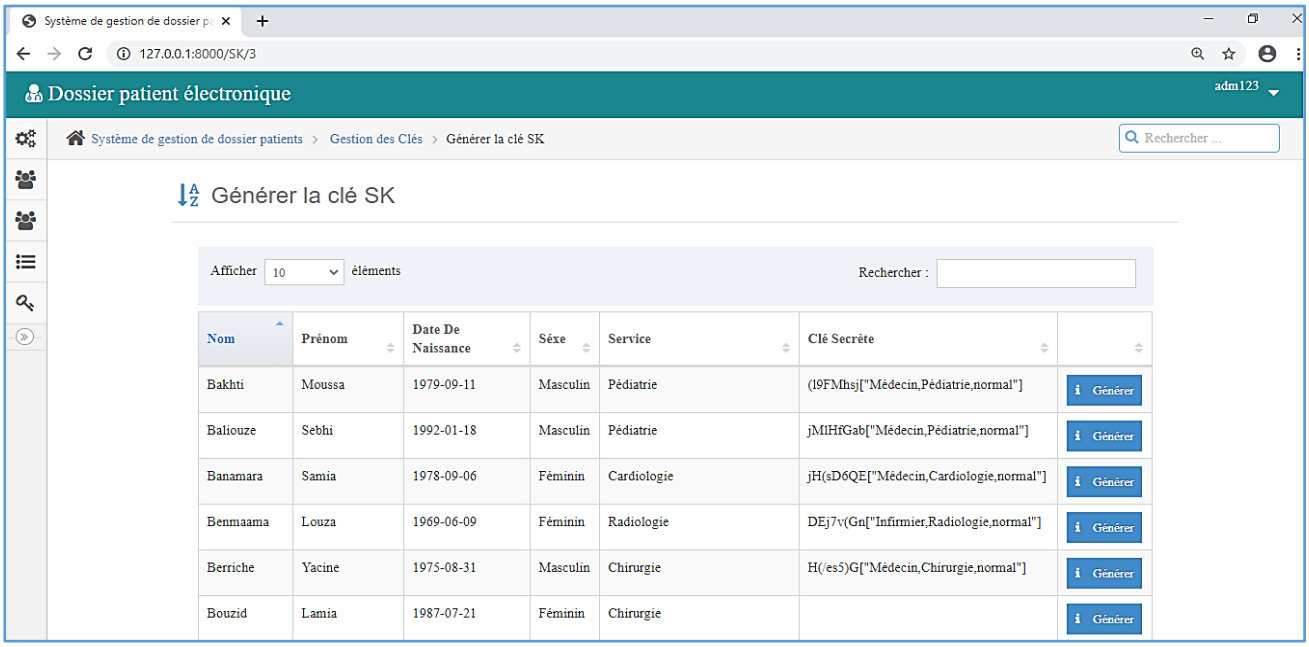


Figure 47: Générer la clé secrète SK

4.2 Espace professionnels de la santé

Cet espace regroupe Gestion des patients, Gestion des consultations et la Gestion des rendez-vous.

- Avant chaque action sur n'importe quel objet on vérifie d'abord si l'utilisateur authentifié (l'employé) est permis de la faire (La fonction CanDo).

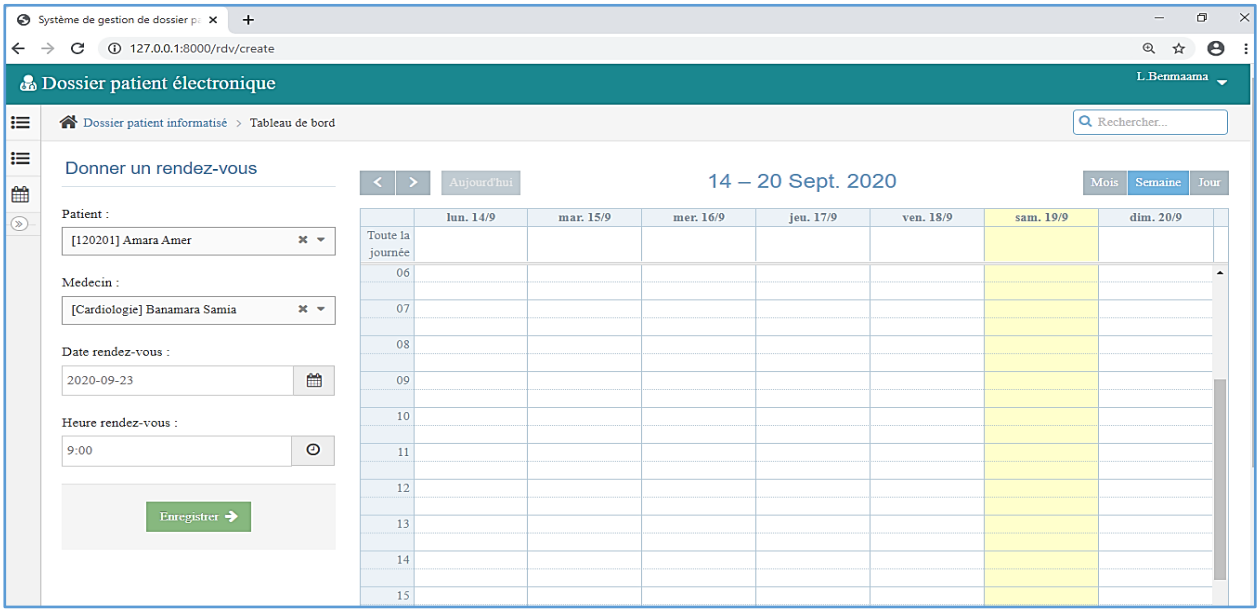


Figure 48: Requête Ajouter un Rdv autorisée

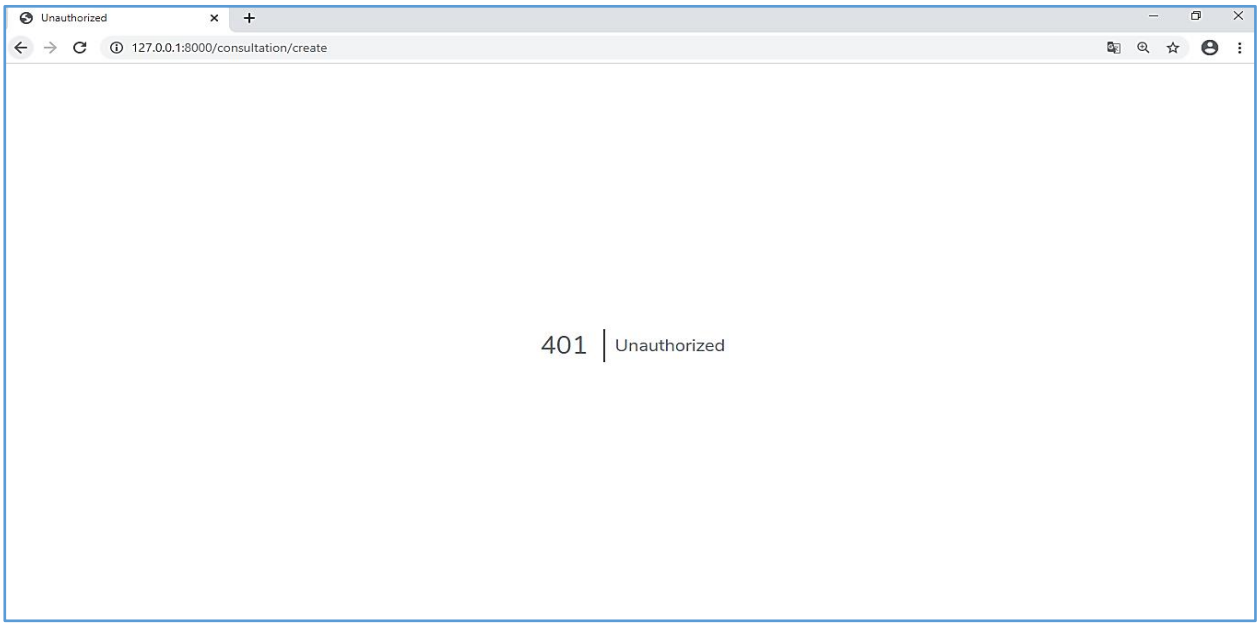


Figure 49: Requête non autorisée

- Après l'ajout d'une nouvelle consultation le créateur de cette donnée (le créateur) a le choix de la crypter.

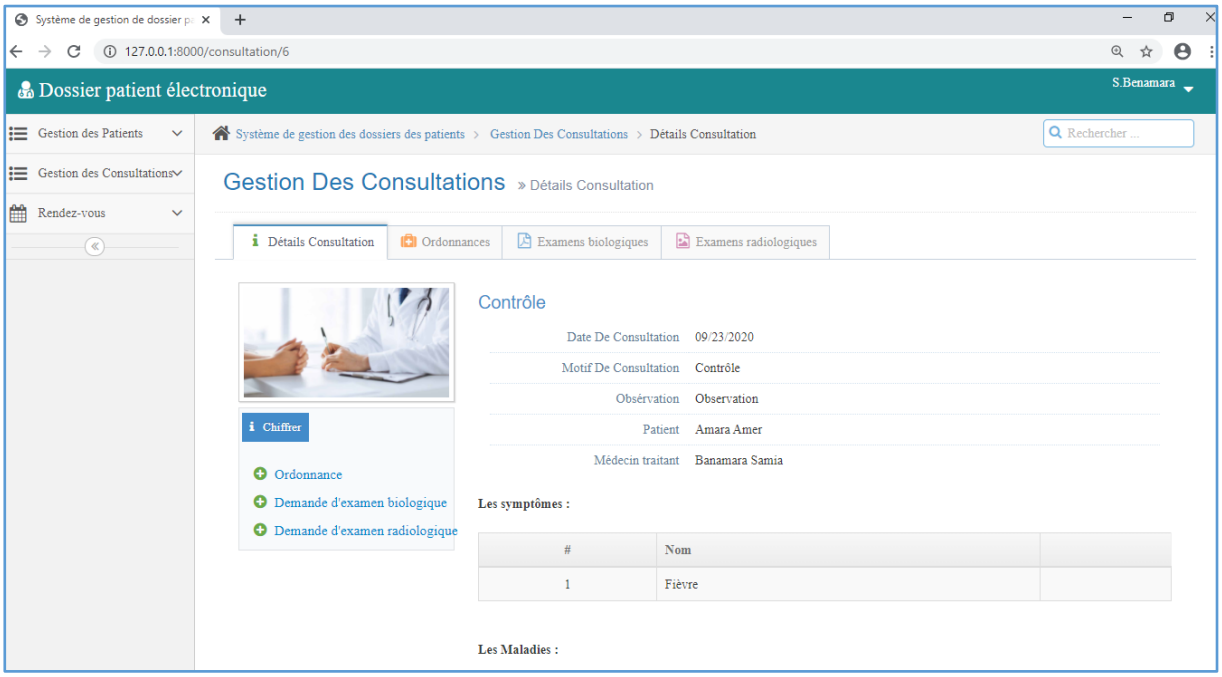


Figure 50: Détails d'une consultation

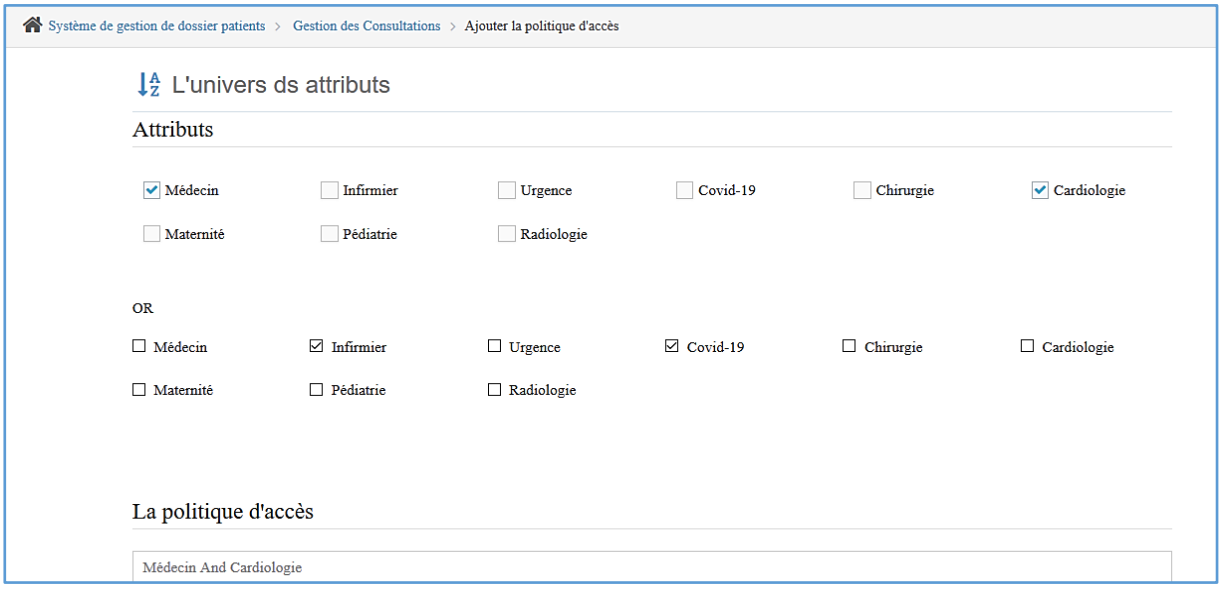


Figure 51: Définir la politique d'accès

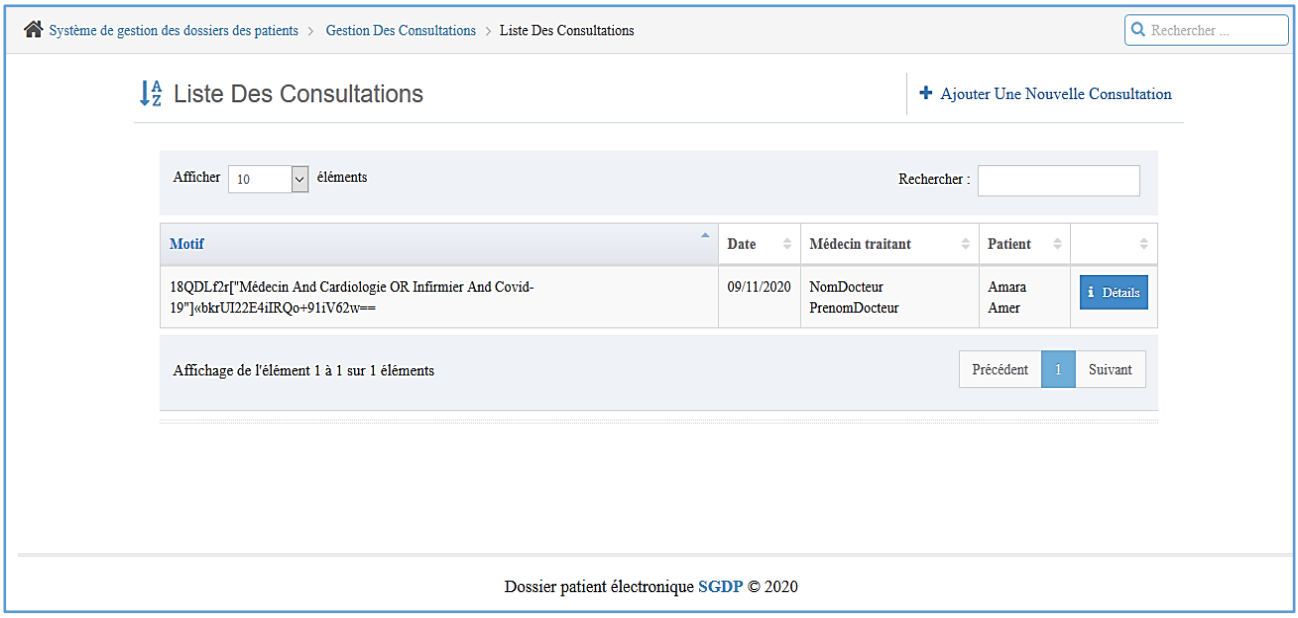


Figure 52: La consultation chiffrée

- Si un autre professionnel de santé veut afficher le détail d'une consultation. les données seront décryptées et affichées si et seulement si l'ensemble d'attributs dans sa clé SK inclus dans la politique définit par le créateur de la donnée.

  ➢ L'utilisateur n'a pas l'autorisation de décrypter

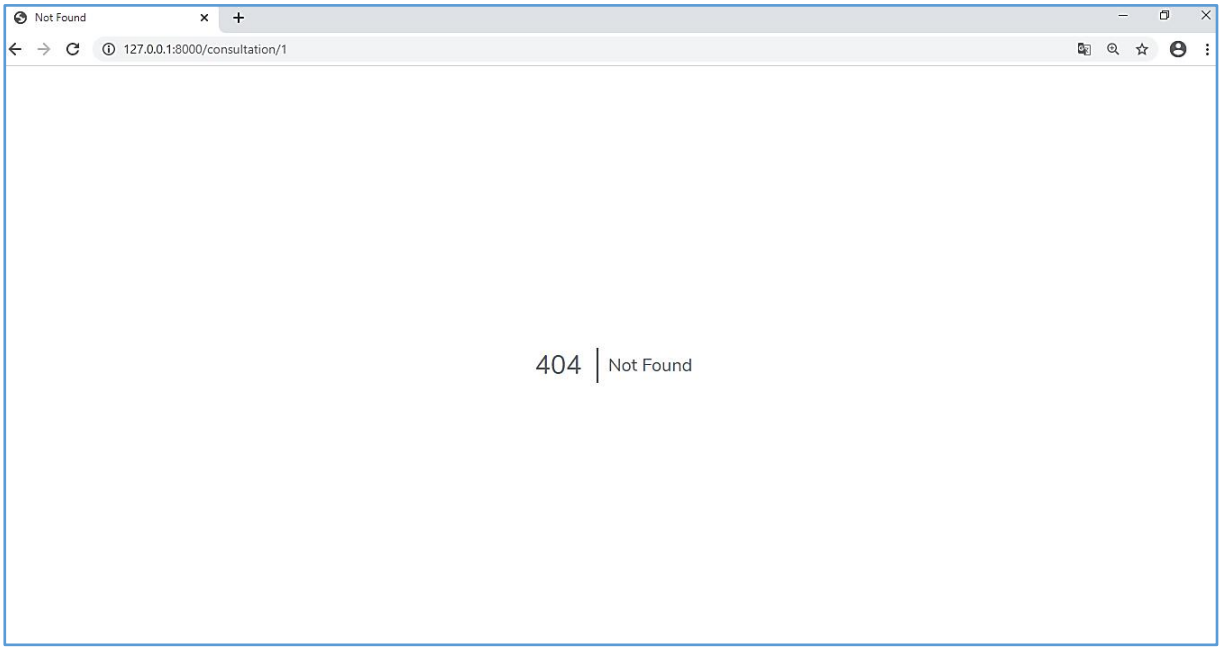


Figure 53: Consultation non décryptée

- L’utilisateur a l’autorisation de décrypter

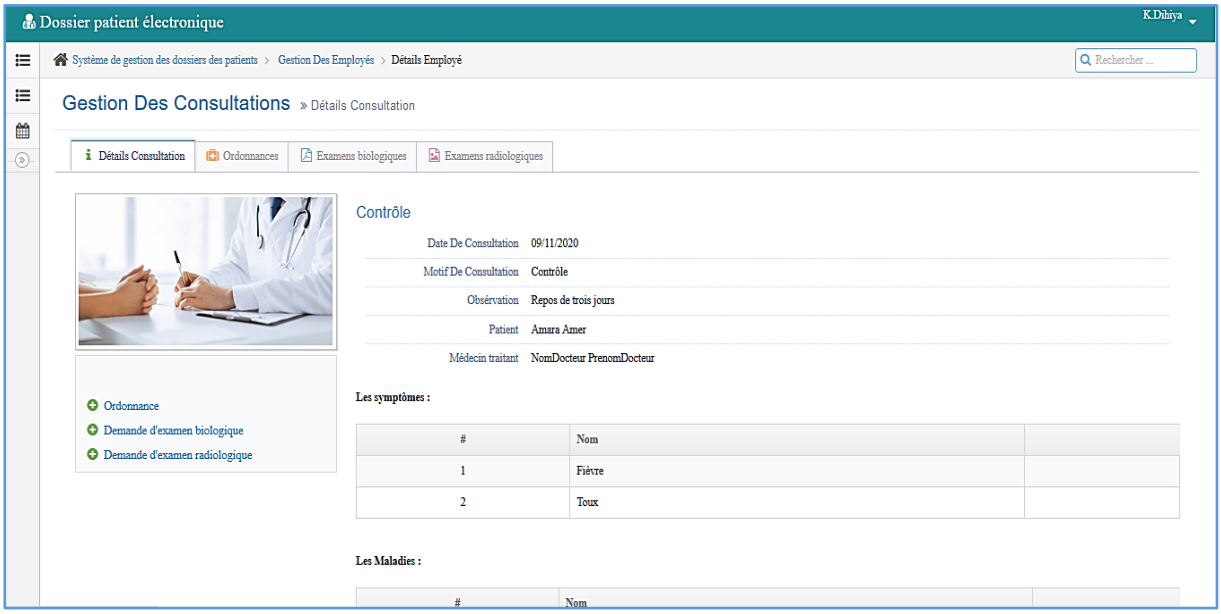


Figure 54: Consultation décryptée

5. Conclusion

Ce chapitre est le dernier de ce mémoire, dans lequel nous avons spécifié les différents outils de développement utilisés dans le travail.

Nous avons présenté le processus d’implémentation et les résultats obtenus par la mise en œuvre de la couche de sécurité de notre système tout en respectant la conception élaborée.

Pour finaliser cette partie on a ajouté des captures d’écran de notre application.

Conclusion Générale

L'objectif visé à travers ce travail est de s'intéresser à la problématique de la sécurité des systèmes de gestion des dossiers médicales électronique afin de pouvoir partager les données médicales sensibles entre les professionnels de santé autorisés tout en préservant la confidentialité, la déontologie médicale et la vie privée des patients.

Pour répondre à cette problématique, nous avons fait une étude bibliographique que nous avons répartie sur trois chapitres.

Dans le chapitre 1, nous avons présenté des généralités sur le dossier médicale électronique.

Ensuite, dans le chapitre 2 nous avons détaillé les différents modèles de contrôle d'accès et enfin les méthodes de la cryptographie dans le chapitre 3.

Après, nous avons proposé une solution de la couche de sécurité qui combine les deux approches de sécurité ; La première approche concerne le contrôle d'accès basé sur l'organisation (ORBAC) et la deuxième approche concerne le chiffrement CP-ABE.

Dans cette couche de sécurité les clés privées des utilisateurs sont spécifiées par un ensemble d'attributs à partir d'ORBAC (rôle, contexte, ...) et les données chiffrées ne peuvent être déchiffrés que par les employés autorisés dont leurs clés privées satisfaites la politique d'accès de chiffrement.

D'autre part, ces deux approches assurent la protection des données du patient car elles permettent aux utilisateurs autorisés l'accès et le déchiffrement de ces dernières et interdire tous utilisateurs non autorisés. Cette solution fournit une couche de sécurité puissante.

Pour mettre en œuvre notre solution, nous avons implémenté les deux approches dans une application web (SGDP Système de gestion de dossier patient) en utilisant le Framework Laravel (langage PHP).

Notre application permet :

- A l'administrateur de gérer les clés de chiffrement, les clés de déchiffrement et les autorisations des utilisateurs,
- A l'utilisateur l'accès au système par l'authentification avec une adresse mail et un mot de passe,
- Au professionnel de santé de créer, rechercher, consulter des consultations.
- Au professionnel de santé autorisé de déchiffrer des consultations. Ces dernières sont chiffrées avant d'être stockées dans la base de données.

Par ailleurs, notre solution de soulève un certain nombre de questions ouvertes intéressantes telles que :

- Sécurité de la politique d'accès : La politique d'accès est sauvegardée en clair, elle doit être chiffrée elle aussi pour empêcher un utilisateur malveillant de l'utiliser et déchiffrer d'autres consultations.
- Si un patient aller chez un médecin qui n'est pas autorisé de déchiffrer ses données alors il faut créer un contexte pour cette problématique.
- Si un virus ou une attaque ou un problème qui empêche le serveur alors l'historique des dossiers médicaux est disparu, alors il faut sauvegarder une copie des données sensibles dans un autre serveur.
- Il faut penser à partager ces dossiers médicaux électroniques entre les différentes organisations au niveau national car si un patient va faire un accident loin de son organisation mère, il faut savoir qu'es ce qu'il a comme maladie chronique par exemple … etc. Alors, ces données peuvent être récupérées à partir de l'organisation mère.

Enfin, les problèmes de sécurité demeurent toujours des problèmes ouverts en conséquence beaucoup de pistes restent à explorer.

Bibliographie

[1] Gnoumou, M. *Mise en place du système d'information hospitalier la clinique SANDOF (*Université Polytechnique Bobo-Dioulasso), p13.

[2] Guiral, E. *Les systèmes d'informations Hospitalier : Histoire, Enjeux et Difficultés Rencontres, Devenir et Lien avec la Médecine de Ville* (Thèse de Doctorat, Université Toulouse III Paul Sabatier), pp. 14-26.

[3] Franchi-Godin, J. *Le dossier Patient Informatisé : Enjeux et Conséquences pour le Personnel* (Mémoire Master, Université de Lille 2), pp23-24.

[4] Guendoul, S., & Manane, S. (2017). *Le Système d'Information Hospitalier (SIH) et le Dossier Médical Informatisé (DMI) Cas du CHU de TIZI-OUZOU* (Thèse de Doctorat, Université Mouloud Mammeri), pp. 28-54.

[5] Miroud, M. (2016). *La Sécurité dans les Systèmes E-Santé* (Mémoire de Magister, Université des Sciences et de la Technologie d'Oran), p18.

[6] Benahmed, H., & Lalmi, M. *Réingénierie du dossier électronique du patient « Approche données semi-structurées* (Thèse de Doctorat), pp. 17-18.

[7] Debiane, N., & Zegmali, F. *Développement d'un modèle pour le contrôle d'accès au dossier médical personnel (Etude de cas : CHU Algérien)* (Mémoire de Master, Université Abou Bakr Belkaid Tlemcen), pp. 14-15.

[8] CERISTNEWS *Bulletin d'information trimestriel* (Huitième numéro Mars 2012) ISSN-2170-0656, pp.11-14.

[9] Didier, G. (2002). Sécurité informatique : Risques, stratégies et solutions (Edipro). Introduction, pp.17-18.

[10] (Juin 2011). *Le Dossier Médical Personnel et la sécurité* (Fiche pratique), pp. 2-8.

[11] *Critères d'évaluation de la sécurité des systèmes informatiques* (ITSEC). (s. d.). Office des publications officielles des Communautés européennes, p20.

[12] Baïna, A. (2009). *Contrôle d'accès pour les grandes infrastructures critiques. Application au réseau d'énergie électrique* (Thèse de Doctorat), pp. 44-47.

[13] Uttha, W. (2016). *Etude des politiques de sécurité pour les applications distribuées: le problème des dépendances transitives: modélisation, vérification et mise en œuvre* (Thèse de Doctorat, Université d'Aix-Marseille), pp. 24-27.

[14] Cheaito, M. (2012). *Un cadre de spécification et de déploiement de politiques d'autorisation* (Thèse de Doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier), pp.26-28

[15] Kumar, A., Karnik, N., & Chafle, G. (2002). Context sensitivity in Role-Based access control. *ACM SIGOPS Operating Systems Review*, *36*(3), pp. 53-66.

[16] David, F., & Richard, K. (1992). Role-Based access controls. In *Proceedings of 15th NIST-NCSC National Computer Security Conference* (Vol. 563). Baltimore, Maryland: NIST-NCSC.

[17] Kalam, A. A. E., Baida, R. E., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., ... & Trouessin, G. (2003, Juin). Organization based access control. In *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks* (pp. 120-131).

[18] Medjdoub, S. (2005). *Modèle de contrôle d'accès pour XML:" Application à la protection des données personnelles"* (Thèse de Doctorat).

[19] Devigne, J. (2013). *Protocoles de re-chiffrement pour le stockage de données* (Thèse de Doctorat), p14.

[20] Dumont, R. (2009). Cryptographie et Sécurité informatique. Université de Liège, *2010*, pp. 8-76.

[21] Hadji, F. (2018). *Conception et réalisation d'un système de cryptage pour les images médicales* (Thèse de Doctorat, Université Mohamed Boudiaf de M'Sila).

[22] Said, A., & Kahina, A. (2008). *Cryptographie et sécurité des réseaux implémentation de L'AES sous Matlab* (Thèse de Doctorat, Université Mouloud Mammeri), pp. 10-46.

[23] Yahia, Y. O. (2019). *Proposition d'un modèle de sécurité pour la protection de données personnelles dans les systèmes basés sur l'internet des objets* (Thèse de Doctorat), p43-48.

[24] Ameur, F., Zerrouki, F. (2018). *Conception et réalisation d'un système hybride pour la compression et la sécurisation des documents* (Mémoires de Master, Université blida 1), pp. 23-24.

[25] Aimeur, A. (2017). *Conception et implémentation d'un système hybride pour la sécurité de données: application aux images numériques* (Thèse de Doctorat, Université Mohamed Boudiaf de M'Sila), pp. 10-15.

[26] Attaf, N., & Cherfa, H. (2012). *Etude sur l'Applicabilité de la Cryptographie Asymétrique aux Réseaux de Capteurs sans Fil*. (Mémoire de Master, Université de Bejaïa), pp. 27-28.

[27] Digital Signature Algorithm. Consulté 28 mai 2020, à l'adresse https://fr.wikipedia.org/wiki/Digital_Signature_Algorithm

[28] Kumar, P., & Aluvalu, R. *Key Policy Attribute Based Encryption (KP-ABE): A Review*. International Journal of Innovative and Emerging Research in Engineering, Volume 2, Issue 2, 2015. p50.