République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb Blida 1

Faculté des Sciences

Département : Informatique

Mémoire De Fin D'études

En Vue De L'obtention Du Diplôme De Master

Option

Sécurité des systèmes d'information

Thème

Réalisation d'un modèle de contrôle d'accès basé sur OrBAC et KP-ABE dans le Cloud.

Réalisé par :

ROUIZI Nafissa

SAADI Sarah

Proposé par : Mme. GHEBGHOUB Yasmina

Mme. OUKID Saliha	**Présidente**
Mme. AROUSSI Sana	**Examinatrice**
Mme. GHEBGHOUB Yasmina	**Encadreur**

Promotion : 2022-2023

Remerciement

Nous tenons à remercier notre encadreur, Mme. GHEBGHOUB pour tous les efforts qu'elle a consentis tout au long de l'élaboration de ce travail, ses encouragements, ses précieux conseils et la confiance qu'elle nous a toujours témoignée nous ont été de grande aide.

Nos remerciements vont également aux membres de jury, Mme. OUKID et Mme.AROUSSI d'avoir consacré leur temps pour évaluer notre travail et d'avoir accepté de participer à ce jury. Nous remercions tous les enseignants qui ont contribués à notre formation.

Nous tenons à exprimer notre profonde reconnaissance envers nos familles et nos proches pour leur soutien inconditionnel et leurs encouragements constants. Leur amour et leur compréhension ont été une source de motivation essentielle.

Enfin, nous souhaitons exprimer notre gratitude envers toutes les personnes qui ont contribué de quelque manière que ce soit, directement ou indirectement, à ce travail.

Nous sommes reconnaissantes envers tous ceux qui ont participé à cette aventure et nous les remercions du fond du cœur.

Résumé

De nos jours, les organisations dépendent de plus en plus du Cloud pour gérer leurs données. Cependant, la sécurité reste un défi majeur, les informations sensibles stockées dans le Cloud nécessitent des mesures de protection adéquates pour prévenir les violations de sécurité et les accès non autorisés. Dans ce contexte, nous proposons un système qui combine les deux approches de sécurité : KP-ABE (chiffrement basé sur les attributs avec une politique de clés) et OrBAC (Contrôle d'accès basé sur l'organisation).

Notre système vise à résoudre certains des problèmes de sécurité liés au Cloud en offrant un contrôle d'accès granulaire aux données stockées. On combine les avantages de chiffrement basé sur les attributs (ABE) et du contrôle d'accès basé sur l'organisation (OrBAC) pour offrir un système de sécurité efficace et flexible dans le Cloud. En organisant les attributs selon le modèle OrBAC, nous permettons une gestion fine des autorisations d'accès aux données. L'ABE assure la confidentialité des données stockées, tandis qu'OrBAC permet une expression claire et indépendante de la politique de sécurité.

Notre système propose une solution robuste et flexible pour la sécurité dans le Clouden intégrant ces deux approches. Il offre une protection renforcée des données sensibles, tout en offrant une gestion d’accès adaptée aux besoins spécifiques de chaque organisation. Cette approche aide les organisations à gagner la confiance dans l’utilisation du cloud pour stocker et gérer leurs données, en offrant une solution de sécurité avancée et adaptée aux exigences actuelles.

Mots clés : Cloud, KP-ABE, OrBAC, Chiffrement, Contrôled’accès, Chiffrement basé sur les attributs(ABE), Contrôle d'accès basé sur l’organisation, Autorisations, Confidentialité, politique de sécurité.

Abstract

Today, organizations increasingly rely on the cloud to manage their data, but security remains a major challenge. Sensitive information stored in the cloud requires adequate safeguards to prevent security breaches and unauthorized access. In this context, we propose a system that combines the two security approaches: KP-ABE (key policy attribute-based encryption) and OrBAC (Organization-based Access Control).

Our system aims to address some of the cloud security issues by providing granular access control to stored data. The benefits of attribute-based encryption (ABE) and organization-based access control (OrBAC) are combined to provide an efficient and flexible security system in the cloud. By organizing the attributes according to the OrBAC model, we allow fine management of data access permissions. ABE ensures the confidentiality of the stored data, while OrBAC allows a clear and independent expression of the security policy.

Our system offers a robust and flexible solution for cloud security by integrating both approaches. It offers enhanced protection of sensitive data, while offering access management adapted to the specific needs of each organization. This approach helps organizations gain confidence in the use of the cloud to store and manage their data, providing an advanced security solution tailored to current requirements.

Keywords: Cloud, KP-ABE, OrBAC, Encryption, Access control, Attribute-based encryption (ABE), Enterprise-based access control Permissions, Privacy, and Security policy.

ملخص

اليوم، تعتمد المنظمات بشكل متزايد على السحابة لإدارة بياناتها، لكن الأمن لا يزال يمثل تحديًا كبيرًا. تتطلب المعلومات الحساسة المخزنة في السحابة ضمانات كافية لمنع الانتهاكات الأمنية والوصول غير المصرح به. في هذا السياق، نقترح نظامًا يجمع بين نهجي الأمان: KP-ABE (التشفير القائم على السمة مع سياسة رئيسية) وOrBAC (التحكم في الوصول القائم على المنظمة).

يهدف نظامنا إلى معالجة بعض مشكلات أمان السحابة من خلال توفير تحكم دقيق في الوصول إلى البيانات المخزنة. يتم الجمع بين فوائد التشفير القائم على السمات (ABE) والتحكم في الوصول القائم على المنظمة(OrBAC) لتوفير نظام أمان فعال ومرن في السحابة. من خلال تنظيم السمات وفقًا لنموذج OrBAC، نسمح بالإدارة الدقيقة لأذونات الوصول إلى البيانات. تضمن ABE سرية البيانات المخزنة، بينما تسمح OrBAC بالتعبير الواضح والمستقل عن السياسة الأمنية.

يوفر نظامنا حلاً قويًا ومرنًا لأمن السحابة من خلال دمج كلا النهجين. وهو يوفر حماية معززة للبيانات الحساسة، ويوفر في الوقت نفسه إدارة للوصول تتكيف مع الاحتياجات المحددة لكل منظمة. يساعد هذا النهج المؤسسات على اكتساب الثقة في استخدام السحابة لتخزين بياناتها وإدارتها، مما يوفر حلاً أمنيًا متقدمًا مصممًا وفقًا للمتطلبات الحالية.

الكلمات الرئيسية: السحابة، KP-ABE، OrBAC، التشفير، التحكم في الوصول، التشفير القائم على السمات (ABE)، أذونات الوصول،السياسة الأمنية.

Table des matières

Table des figures

Table des tableaux

Liste des abréviations

ABE	Attribut Based Encryption
AES	Advanced Encryption Standard
CIA	Cenfidentiality Integrity Availability
CT	Cipher Text
DAC	Discetionary Access Control
DDOS	Distributed Denial of Service
DES	Data Encryption Standard
ECC	Elliptic Curve Cryptography
Hcp-abe	Hidden Ciphertext Policy Attribute Based Encryption
IaaS	Infrastructure as a service
IPFS	InterPlanetary File System
MAC	Mandatory Access Control
MSK	Master Key
MVC	Model-View-Controller
NIST	National Institute of Standards and Technology
OrBAC	Organization-Based Access ControlControl
PaaS	Platform as a service
PK	Public Key
RBAC	Role-Based Access Control
RSA	Rivest-Shamir-Adleman
SaaS	Software sa a Service
SK	Secret Key
UML	Unified Modeling Language

Introduction générale

Le cloud computing est devenu un concept majeur faisant référence à l'utilisation de la mémoire et de la puissance de traitement des ordinateurs et des serveurs situés dans le monde entier et connectés par un réseau, comme Internet. Le cloud, cependant, introduit un certain nombre de problèmes de sécurité, y compris la confidentialité, l'intégrité et la disponibilité. Les nouvelles menaces à la sécurité, comme l'accès non autorisé et les attaques par déni de service, sont présentées par leur nature distribuée.

Certains des problèmes de sécurité liés au cloud peuvent être résolus à l'aide de technologies de contrôle d'accès et de cryptage. Lorsque les données sont chiffrées, seules les parties ayant la clé de déchiffrement peuvent y déchiffrer. Cela rend difficile pour les attaquants d'accéder à ces données sensibles, même s'ils parviennent à violer le système de cloud. Par contre, les systèmes de contrôle d'accès limitent l'accès aux ressources du cloud grâce a des politiques prédéfinies. Cela garantit que les données et les ressources sensibles dans le cloud ne peuvent être accessibles que par les parties autorisées.

Le contrôle d'accès est un aspect essentiel de la sécurité du cloud en raison des risques associés au stockage de données dans le cloud. Cependant les modèles traditionnels de contrôle d'accès pourraient ne pas être suffisants pour relever les défis de cloud. La question qui se pose est : comment la combinaison d'OrBAC et KP-ABE peut être utiliséepour résoudre les défis de sécurité dans les environnements de cloud.

Notre objectif est d'élaborer un modèle de contrôle d'accès puissant basé sur KP-ABE et OrBAC, garantissant un accès à grain fin aux données sensibles stockées dans un Cloud et renforçant la confidentialité et l'intégrité de ces données en utilisant une combinaison d'un modèle de contrôle d'accès et d'un algorithme de chiffrement basé sur une structure d'attributs en donnant la possibilité à l'utilisateur d'exprimer qu'il peut accéder à son fichier et enfin, d'augmenter la confiance entre les clients et fournisseurs du cloud en chiffrant les données.

Ce mémoire est organisé comme ceci:

- Dans le premier chapitre, nous allons examiner l'utilisation des systèmes de contrôle d'accès et de cryptage et leur rôle crucial dans la sécurité du cloud. Nous allons ensuite

discuter du cloud, en spécifiant ses caractéristiques, ses niveaux et ses modèles de déploiement.

- Le deuxième chapitre est consacré à la présentation de notre solution et à la modélisation de notre travail.
- Le dernier chapitre est consacré à la mise en œuvre de notre solution, en utilisant différents outils et technologies pour atteindre notre objectif.

Nous concluons notre mémoire par une conclusion générale de notre travail et des perspectives d'amélioration de la solution proposée.

Chapitre I :

La sécurité des données dans le Cloud.

I.1-Introduction

Le cloud est devenu de plus en plus populaire ces dernières années en raison de ses nombreux avantages tels que la rentabilité, l'évolutivité et la flexibilité. Cependant, malgré sa popularité, le cloud est toujours confronté à un certain nombre de défis de sécurité. Les données confidentielles stockées dans le cloud sont vulnérables. Il est donc important que les organisations adoptent des mesures de sécurité efficaces, comme le contrôle d'accès et le chiffrement, pour protéger leurs ressources et leurs données dans le cloud.

Dans ce chapitre, nous abordons trois aspects clés. Nous commençons par introduire les systèmes de contrôle d'accès, qui sont essentiels pour protéger et limiter l'accès aux ressources cloud. Ensuite, nous discutons du rôle du chiffrement dans la protection des données qui sont partagées et stockées. Enfin, nous décrivons le concept du cloud computing, en nous concentrant sur les problèmes de sécurité uniques et les solutions associées à cette technologie.

I.2-La sécurité informatique

La sécurité informatique, qui est la protection accordée à une information automatisée, vise à préserver l'intégrité, la disponibilité, et la confidentialité des ressources du système d'information contre les menaces accidentelles ou intentionnelles. Elle assure que les ressources matérielles et/ou logicielles d'un parc informatique sont uniquement utilisées dans le cadre prévu et par des personnes autorisées [1].

La sécurité informatique comprend trois piliers principaux, bien connus sous l'acronyme CIA(figure1): la confidentialité, l'intégrité et la disponibilité [2].

- La confidentialité implique la dissimulation de renseignements sensibles à des parties non autorisées. Trois mécanismes permettant d'assurer la confidentialité sont la cryptographie, le contrôle de l'accès et l'autorisation.
- Le deuxième pilier, l'intégrité, vise à empêcher toute altération non autorisée en premier lieu ou à détecter une telle altération après qu'elle s'est produite.
- Le troisième pilier, la disponibilité, fait référence aux biens qu'un système et ses données devraient être mis à la disposition des parties en temps opportun.

Il y a d'autres concepts importants comme l'authenticité, qui est la propriété des données et des transactions qui sont authentiques, et la non-répudiation, qui est l'assurance qu'une partie ne peut pas refuser une transaction, une déclaration ou une signature [2].

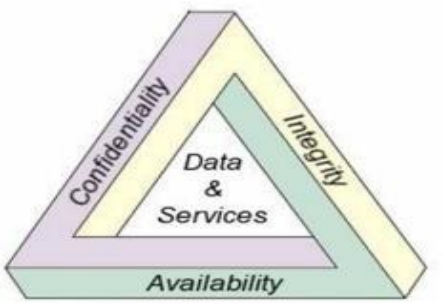


Figure 1: CIA [3].

I.3-Les systèmes de contrôle d'accès

Le contrôle d'accès est essentiel pour sécuriser les ressources informatiques et déterminer qui peut y accéder et quelles actions ils peuvent effectuer. Il prévient les accès non autorisés et protège la confidentialité et l'intégrité des informations.

- Le contrôle d'accès peut prendre plusieurs formes. En plus de déterminer si un utilisateur a le droit d'utiliser une ressource, le système de contrôle d'accès peut limiter quand et comment les ressources peuvent être utilisées.
- Le contrôle d'accès est essentiel pour préserver la confidentialité et l'intégrité des informations.
- La nécessité du contrôle d'accès pour préserver la disponibilité est moins évidente, mais il a clairement un rôle important : un attaquant qui gagne un accès non autorisé à un système est susceptible de le perturber.

I.3.2-Modèles de contrôles d'accès

Dans cette section, nous abordons divers types de modèles de contrôle d'accès, chacun ayant ses propres points forts, ses caractéristiques distinctives et ses limites. Ces modèles sont utilisés pour réglementer l'accès aux systèmes, aux données et aux ressources, en assurant l'intégrité et la confidentialité des données sensibles.

I.3.2.1-Le contrôle d’accès discrétionnaire (DAC)

Dans le modèle DAC (Discretionary Access Control), la politique de contrôle d'accès est basée sur le concept de sujet, d'action et d'objet. Les sujets représentent les entités actives du système, généralement les utilisateurs, tandis que les objets représentent les entités passives ou les données. Les actions définissent les accès directs que les sujets peuvent effectuer sur les objets. Les modèles DAC sont qualifiés de discrétionnaires, car les permissions font référence directement à un utilisateur particulier [4]. Considérant que chaque sujet peut détenir un droit de possession sur un objet, le propriétaire de l'objet peut alors accorder des droits sur son objet à d'autres sujets. Il en résulte cependant un problème de perte de confidentialité de l’information [4].

I.3.2.2-Le contrôle d’accès mandataire (MAC)

Contrairement aux modèles DAC, les modèles MAC (Mandatory Access Control) relèvent de la catégorie des modèles de contrôle de flux, car la seule façon de garantir pleinement la sécurité à plusieurs niveaux est de contrôler tous les flux d'information possibles. Dans les modèles MAC, un niveau de sécurité est attribué à chaque sujet (niveau d'autorisation) et à chaque objet (niveau de classification) [4]. La politique de sécurité est obligatoire, c'est-à-dire qu'elle est contraignante pour tous les utilisateurs et ne peut pas être modifiée. Dans ce type de modèle, deux propriétés de contrôle d'accès doivent être appliquées :

- "Pas de lecture ascendante" : ce qui signifie qu'un sujet ne peut pas lire un objet classifié à un niveau de confidentialité supérieur à l'autorisation du sujet.
- "Pas d'écriture descendante" : ce qui signifie qu'un sujet ne peut pas écrire dans un objet classifié à un niveau de confidentialité inférieur à l'autorisation du sujet.

Ainsi, le problème de perte de confidentialité décrit précédemment ne peut pas se poser. Mais il a des limites en termes de complexité à mettre en œuvre et à gérer. Ils nécessitent souvent une politique de sécurité bien définie, une configuration soignée et une administration continue pour assurer des contrôles d’accès adéquats [4].

I.3.2.3-Le contrôle d'accès basé sur les rôles (RBAC)

Le contrôle d'accès basé sur les rôles RBAC (Role Based Access Control) est un modèle de contrôle d'accès dans lequel les décisions d'accès sont basées sur les rôles auxquels l'utilisateur est attaché. Contrairement aux modèles de contrôle d'accès DAC ou MAC, dans le RBAC la politique de contrôle d'accès ne s'applique pas directement aux utilisateurs et 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. Un rôle découle généralement de la structure d'une entreprise. Les utilisateurs exerçant des fonctions similaires peuvent être regroupés sous le même rôle. Un rôle, déterminé par une autorité centrale, associé à un sujet des autorisations d'accès sur un ensemble d'objets [5].

La modification des contrôles d'accès n'est pas nécessaire chaque fois qu'une personne rejoint ou quitte une organisation. Lorsqu'un nouveau sujet est créé dans le système d'information, il suffit d'affecter des rôles au sujet pour que ce sujet puisse accéder au système d'information conformément aux permissions accordées à cet ensemble de rôles [5].

I.3.2.4-Contrôle d'accès basé sur l'organisation(OrBAC)

Les modèles cités (DAC, MAC, RBAC) ne permettent de modéliser que des politiques de sécurité qui se limitent à des permissions statiques. Ils ne permettent pas la possibilité d'exprimer des règles contextuelles [6].

Le contrôle d'accès basé sur l'organisation (OrBAC Organization Based Access Control) est un modèle qui permet de spécifier des politiques de sécurité contextuelle relative aux permissions, interdictions, obligations et recommandations. OrBAC utilise la notion de hiérarchie de rôle, c'est-à-dire un mécanisme d'héritage de permission à travers la hiérarchie de rôle. Ceci peut être applicable dans le cas d'une organisation ayant des sous-organisations, les organisations se succèdent de père en fils [6].

Le but principal d'OrBAC est de permettre de définir une politique de sécurité indépendamment de son implémentation. Il utilise deux niveaux d'abstraction pour exprimer une politique de sécurité [6] :

Le niveau abstrait : Ce niveau est appelé aussi niveau organisationnel, il est représenté par l'utilisation de trois métas entités : rôle, activité et vue qui spécifient les politiques de sécurité.

Le niveau concret : comprend les sujets, les actions et les objets. C'est à ce niveau que les politiques concrètes précisent qu'un certain sujet a la permission de réaliser une certaine action sur un certain objet.Ces politiques sont dérivées des politiques abstraites, c'est-à-dire en dérivant la relation abstraite qui autorise le rôle joué par le sujet de réaliser l'activité considérant l'action sur la vue utilisée par l'objet à sécuriser.

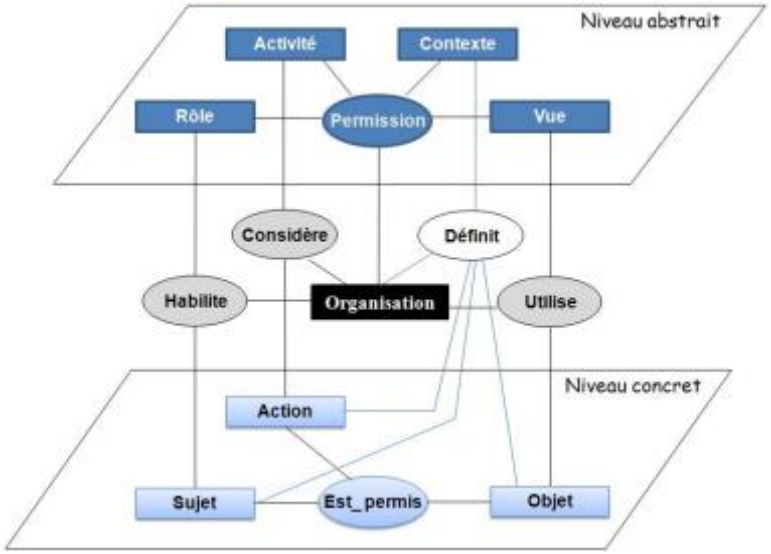


Figure 2:Le modèle OrBAC [7].

Pour définir les relations correspondantes aux permissions, On appelle ***Est_permis*** la relation entre un sujet, une action et un objet [7]:

Est_permis (Sujet, Action, Objet)

De telles règles de contrôle d'accès sont concrètes et sont similaires aux règles de contrôled'accès obtenues dans le modèle DAC par exemple. On appelle ***Permission*** la relation abstraite entre un rôle, une activité et une vue. L'organisation dans laquelle une permission est valide est aussi indiquée dans la relation [7]:

Permission (Organisation, Rôle, Activité, Vue, Contexte)

Cette relation signifie que l'organisation donne la permission à un rôle de réaliser une activité sur une vue. L'objectif dans OrBAC est de rédiger la politique de sécurité à l'aide de permissions abstraites. Les permissions concrètes sont alors dérivées des permissions abstraites. La règle de dérivation est la suivante [7]:

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)

AlorsEst_permis (Sujet, Action, Objet)

Figure 3:Définition d'une permission avec le formalisme OrBAC [7].

La relation ***Est_permis*** modélise les permissions concrètes entre les sujets, les objets et les actions.

I.3.3-Comparaison desdifférents modèles de contrôle d'accès

Les modelés CA	**DAC**	**MAC**	**RBAC**	**ORBAC**
Performance	Faible	Selon le niveau de sécurité	Élevé	Élevé
Réutilisabilité	Multi	Non mentionné	Multi	Multi
Attribution des rôles	Non mentionné	Attribution à un seul nœud	Multi	Multi
Défaillance ponctuelle	Échec d'autorisation	Moins	Moins	Moins
Tête de nœud	Moins	Moins	Moins	Varie
Dynamité	Faible	Faible	Faible	Élevé

Tableau 1: Comparaison des différents modèles traditionnels de contrôle d'accès [8].

ORBAC (Contrôle d'accès basé sur l'organisation) est un modèle de contrôle d'accès plus complexe que les modèles DAC et MAC en raison de sa structure à deux niveaux et de son utilisation d'attributs. Cette complexité supplémentaire permet une granularité plus fine dans la définition des autorisations.

ORBAC s'appuie sur les concepts de contrôle d'accès basé sur les rôles (RBAC) tout en introduisant une composante organisationnelle. Il offre des autorisations plus précises en intégrant des attributs précis, offrant une plus grande souplesse dans la gestion des autorisations. Avec ORBAC, le contrôle d'accès peut être adapté selon le rôle des utilisateurs au sein des organisations, ce qui permet des environnements de contrôle d'accès plus complexes impliquant plusieurs organisations et leurs relations hiérarchiques. Ce modèle améliore la gestion des autorisations et offre une approche globale pour le contrôle d'accès dans des contextes variés.

I.4-Cryptographie

À la fin du XXe siècle. Une riche théorie a émergé, permettant l'étude rigoureuse de la cryptographie comme une science. En outre, le domaine de la cryptographie englobe maintenant beaucoup plus que la communication secrète, y compris l'authentification des messages, les signatures numériques, les protocoles d'échange de clés secrètes, les protocoles d'authentification, les élections électroniques, et l'argent numérique. Sans tenter de fournir une définition parfaite de la cryptographie moderne, nous pourrions dire que c'est l'étude scientifique des techniques pour sécuriser l'information numérique, les transactions et les calculs distribués [9].

I.4.1-L'objectif de la cryptographie

L'un des objectifs fondamentaux de la cryptographie est d'aborder adéquatement les quatre domaines suivants tant en théorie qu'en pratique. La cryptographie consiste à prévenir et à détecter les activités malveillantes [9].

a) La confidentialité est un service utilisé pour garder le contenu de l'information de tous, sauf ceux qui sont autorisés à l'avoir. Le secret est un terme synonyme de confidentialité et de vie privée. Il existe de nombreuses approches pour assurer la confidentialité, allant de la protection physique aux algorithmes mathématiques qui rendent les données inintelligibles.

b) L'intégrité des données est un service qui traite de la modification non autorisée des données. Pour assurer l'intégrité des données, il faut avoir la capacité de détecter la manipulation des données par des parties non autorisées. La manipulation des données comprend l'insertion, la suppression et la substitution.

c) L'authentification est un service lié à l'identification. Cette fonction s'applique à la fois aux entités et à l'information elle-même. Les informations transmises sur un canal doivent être authentifiées en ce qui concerne l'origine, la date d'origine, le contenu des données et l'heure d'envoi.

d) La non-répudiation est un service qui empêche une entité de nier des engagements ou des actions antérieures. Lorsqu'un différend survient parce qu'une entité nie que certaines mesures ont été prises, un moyen de régler la situation est nécessaire.

I.4.2-Cryptographie à clé symétrique

La cryptographie à clé symétrique, également appelée cryptographie à clé secrète ou à clé partagée, est un mécanisme où l'expéditeur et le destinataire partagent une clé commune pour le chiffrement et le déchiffrement. Pour être efficace, la clé doit être partagée via une communication sécurisée. Si elle est compromise, le message chiffré peut être facilement déchiffré par un attaquant. Cette technique cryptographique offre un service rapide sans utiliser de nombreuses ressources. Divers algorithmes, tels que AES, DES, et 3DES, ont été développés pour mettre en œuvre la cryptographie à clé symétrique [10].

Standard de chiffrement avancé 'AES'

(Advanced Encryption Standard) est l'algorithme de chiffrement le plus polyvalent utilisé pour le chiffrement de données. Avec l'émergence de l'AES, de nombreux algorithmes de chiffrement tels que le DES et le 3DES ont été remplacés. De nombreuses banques utilisent encore aujourd'hui l'algorithme de chiffrement AES. AES est un chiffre symétrique par bloc qui utilise plusieurs rounds de chiffrement. AES dispose de plusieurs chiffres par bloc tels qu'AES-128, AES-192 et AES-256. AES effectue plusieurs rounds de chiffrement, par exemple 10, 12 et 14 rounds. Ainsi, le texte en clair d'entrée est converti en texte chiffré après l'exécution d'une séquence de rounds [11].

I.4.3-Cryptographie à clé asymétrique

La cryptographie à clé asymétrique est également connue sous le nom de cryptographie à clé publique. Dans cette technique, le mécanisme est un peu différent, l'émetteur utilise une clé publique du destinataire pour le chiffrement ce qui rend le message indéchiffrable sans la clé appropriée. Quand le message atteint son destinataire, celui-ci utilise sa clé privée pour déchiffrer le message [10]. Il existe différents algorithmes pour mettre en œuvre ce

mécanisme de chiffrement, tels que RSA, Diffie-Hellman, ECC et algorithme de signature numérique.

Chiffrement par attributs 'ABE '

Sahai et Waters ont proposé un nouveau type de cryptage basé sur l'identité qu'ils ont appelé le cryptage basé sur l'identité floue dans lequel sont considérées les identités comme un ensemble d'attributs descriptifs.

Dans ABE (Attribute Based Encryption), les données sont cryptées en utilisant des attributs et décryptées à l'aide de la clé secrète d'un utilisateur associé à une politique d'accès. L'utilisateur ne peut déchiffrer que lorsque les identifiants de l'utilisateur satisfont à la politique d'accès, et il ne fournit pas seulement le contrôle d'accès fin, mais fournit également la révocation, résistant à la collision [12]. Deux types d'ABE ont été proposés :

- **Le chiffrement basé sur des attributs avec une politique de clé (KP-ABE)**

Il a été développé 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 [13].

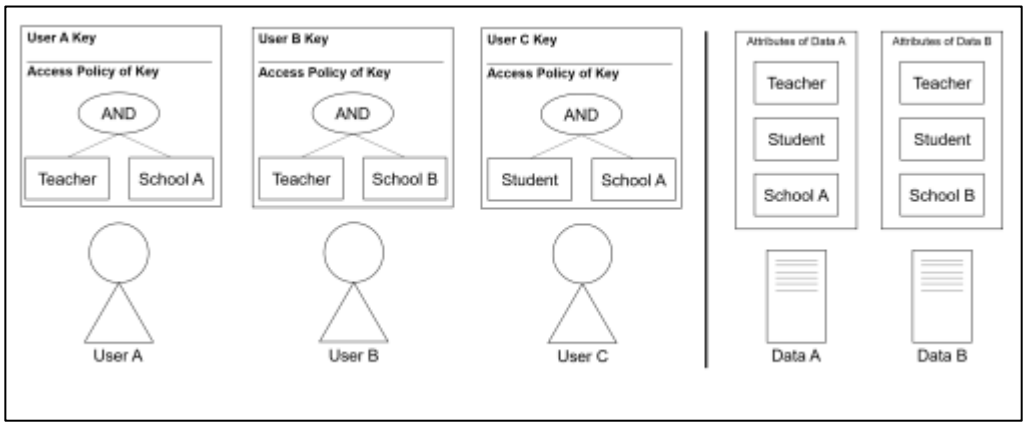


Figure 4:Exemple de KP-ABE [14].

Dans l'exemple de KP-ABE (Figure 3), **les données A** représentent les données internes de **l'école A** décrites par les attributs **Enseignant, Étudiant et École A**.

L'utilisateur A (l'enseignant de **l'école A**) dispose de la clé avec la politique d'accès : **("Enseignant" ET "École A")** et peut déchiffrer **les données A,** car les attributs des données A satisfont la politique d'accès.

L'utilisateur B, en tant qu'enseignant de **l'école B**, ne peut pas déchiffrer **les données A**. **L'utilisateur B** dispose de la clé avec la politique d'accès : **("Enseignant" ET "École B")**, mais les attributs des **données A** ne satisfont pas la politique d'accès.

- **Le chiffrement basé sur des attributs avec une politique de texte chiffré (CP-ABE)**

Proposé 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. Seules les clés secrètes avec un ensemble d'attributs qui satisfait la politique d'accès peuvent récupérer le texte en clair [15].

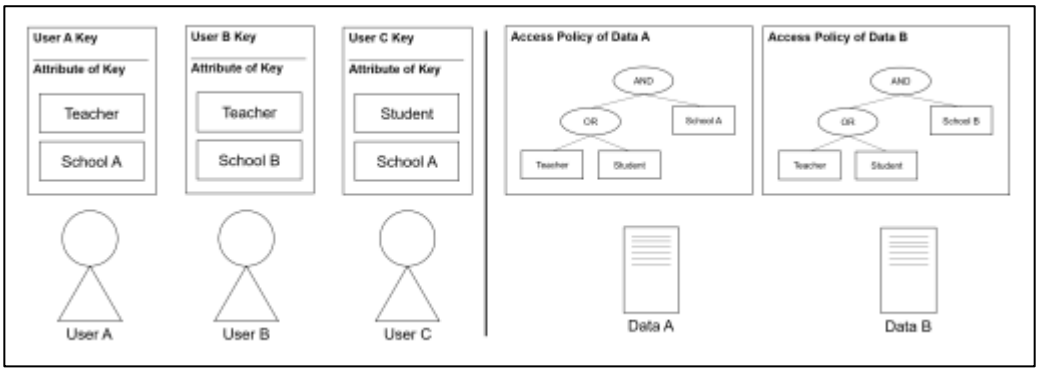


Figure 5:Exemple de CP-ABE [14].

Dans l'exemple de CP-ABE (Figure 4), **les données A** représentent les données internes de **l'école A**avec la politique d'accès : **("Enseignant" OU "Étudiant") ET "École A".**

L'utilisateur A (l'enseignant de **l'école A**) et **l'utilisateur C** (l'étudiant de **l'école A**) peuvent tous deux déchiffrer **les données A**.

L'utilisateur B, en tant qu'enseignant de **l'école B**, ne peut pas déchiffrer **les données A** mais peut déchiffrer **les données B**.

- **Politique d'accès :**

Soit U = {Attr1, Attr2,…, Attrn} l'univers des attributs. La politique d'accès A est une collection de sous-ensembles non vides de U (c'est-à-dire A $\subseteq 2^U \setminus \{\emptyset\}$). Nous appelons A une collection monotone lorsque :

∀B, C ∈ U, si B ∈ A et B ⊆ C, alors C ∈ A

La politique d'accès A est une collection monotone et contient les ensembles autorisés d'attributs [14]. Une façon naturelle de construire la politique d'accès A est d'utiliser une formule booléenne avec les opérateurs ET et OU. Cette formule prend l'ensemble d'attributs S en entrée et renvoie vrai lorsque S satisfait.

I.4.4-comparaison entre les deux approches d'ABE

CP-ABE et KP-ABE sont deux approches dérivées de l'ABE (Attribute-Based Encryption) qui utilisent le chiffrement basé sur les attributs pour protéger les données. La principale différence entre elles réside dans la manière dont les politiques d'accès sont définies.

Le tableau ci-dessous présente les différences entre KP-ABE (Key Policy Attribute-Based Encryption) et CP-ABE (Cipher Policy Attribute-Based Encryption) :

Paramètres	KP-ABE	CP-ABE
Contrôle d'accès précis	Faible, Élevé s'il existe une technique de ré-encodage	Réalisation moyenne d'un contrôle d'accès complexe
Efficacité	Moyenne	Moyenne, pas efficace pour un environnement d'entreprise moderne
Résistance aux collisions	Bien	Bien

Tableau 2:Comparaison entre KP-ABE et CP-ABE [16].

Le modèle KP-ABE peut être plus efficace en raison de sa capacité à gérer les autorisations d'accès basées sur les attributs. Il permet de définir des politiques d'accès granulaires en utilisant des attributs spécifiques, ce qui offre une plus grande flexibilité dans la gestion des autorisations dans un environnement cloud où les ressources sont partagées entre plusieurs utilisateurs. Cela permet de contrôler de manière plus précise l'accès aux données sensibles et de garantir la confidentialité des informations stockées dans le cloud. En revanche, le modèle CP-ABE peut ne pas être aussi efficace dans les environnements de cloud en raison de la complexité de gestion des attributs des utilisateurs.

I.4.5-Les travaux existants

Dans cette partie, nous allons présenter plusieurs travaux qui traitent de ce sujet. Nous allons mettre en évidence plusieurs articles qui explorent l'utilisation du cryptage basé sur les attributs, et des modèles de contrôle d'accès :

1. Schéma de cryptage basé sur les attributs avec structures d'accès non monotoniques

En 2007, Ostrovsky et al. [17]ont proposé un chiffrement basé sur les attributs avec une structure d'accès non monotonique. La formule d'accès de la structure d'accès dans la clé privée de l'utilisateur peut représenter n'importe quel type à travers des attributs,y compris des attributs négatifs.. Il est différent du précédent système de cryptage basé sur les attributs. Les schémas précédents sont comme le schéma KP-ABE, et la structure d'accès dans la clé privée de l'utilisateur à une formule d'accès monotone. Il n'y a pas d'attributs négatifs. À part cela, la structure d'accès de ce système est la même que celle du système KP-ABE. Il y a une formule booléenne telle qu'And, OR et des portes de seuil dans ces structures d'accès, mais il y a une formule booléenne, NOT dans la structure d'accès de ce schéma.

2. Schéma de cryptage basé sur les attributs hiérarchiques

En 2011, Wang et al. [18]ont proposé un système de chiffrement hiérarchique fondé sur les attributs composé d'un système de chiffrement hiérarchique fondé sur l'identité (HIBE) et d'un système de chiffrement fondé sur cp-abe. Ce schéma utilisait la propriété de génération hiérarchique de clés dans le schéma HIBE pour générer des clés. En outre, il a utilisé la forme normale disjonctive (DNF) pour exprimer la politique de contrôle d'accès, et la même autorité de domaine dans ce schéma a administré tous les attributs dans une seule clause conjonctive. Ce schéma comporte cinq rôles : le service de stockage en cloud,

le propriétaire des données, l'autorité racine, l'autorité du domaine et les utilisateurs de données.

3. Chiffrement basé sur les attributs de la politique de chiffrement avec révocation efficace

Liang et al. [19] ont proposé le CP-ABE avec la révocation des utilisateurs dans lequel ils ont utilisé les techniques de partage secret linéaire et d'arborescence binaire, et il a été prouvé que la sécurité sous modèle standard. Ici, chaque utilisateur affecté avec l'identifiant unique, il est donc très facile de révoquer l'utilisateur avec l'identifiant unique. La nouvelle clé est générée avec une liste de révocation qui est maintenue par le propriétaire des données.

4. Chiffrement basé sur des attributs avec une politique de texte chiffré caché

Zhang et al. [20] ont développé une nouvelle méthode de vérification pour vérifier si l'utilisateur possède les attributs appropriés qui correspondent à la politique d'accès partiellement masquée. Cependant, ils supposent que la correspondance est vraie avant d'obtenir le résultat de contrôle final, le système HCP-ABE est conçu pour fournir un contrôle d'accès précis et la préservation de la vie privée.

Comparaison des travaux existants :

- Contrôle d'accès granulaire

Dans le même groupe, le système a accordé le droit d'accès différent à l'utilisateur individuel. Les utilisateurs sont sur le même groupe, mais chaque utilisateur peut avoir le droit d'accès différent aux données. Même pour les utilisateurs d'un même groupe, leurs droits d'accès ne sont pas les mêmes.

- Confidentialité des données

Avant de télécharger des données sur le cloud, celles-ci étaient cryptées par le propriétaire des données. Par conséquent, les parties non autorisées, y compris le cloud, ne peuvent pas connaître les informations sur les données cryptées.

- Révocation de l'utilisateur

Si l'utilisateur quitte le système, le système peut révoquer directement son droit d'accès. L'utilisateur révocable ne peut pas accéder aux données stockées, car son droit d'accès a été révoqué.

- Politique cachée

Cela permet de protéger la confidentialité des politiques de sécurité tout en garantissant que seuls les utilisateurs autorisés peuvent accéder aux données.

critères	1	2	3	4
Niveau de granularité du contrôle d'accès	Oui	Oui	Oui	Oui
Confidentialité des données	Oui	Oui	Oui	Oui
Révocation des utilisateurs	Non	Moyenne	Oui	Oui
Politique cachée	Non	Non	Non	Oui

Tableau 1:Comparaison des travaux existants [21].

I.5-Le cloud computing

En général le concept de base de cloud computing est l'utilisation des ressources par l'intermédiaire d'internet sans que l'utilisateur connaisse la localisation de ces ressources. Selon le NIST [22],"le cloud computing est un modèle qui permet un accès réseau omniprésent, pratique et à la demande à un pool partagé de ressources informatiques configurables, par exemple : réseaux, serveurs, stockage, applications et services".

Étant donné que le cloud a simplement besoin d'une connexion Internet pour fonctionner, il est vraiment moins cher et plus simple que d'autres options, c'est pour cette raison que la plupart des entreprises le considèrent comme un choix supérieur.

Cloud computing se caractérise par cinq caractéristiques [8] :

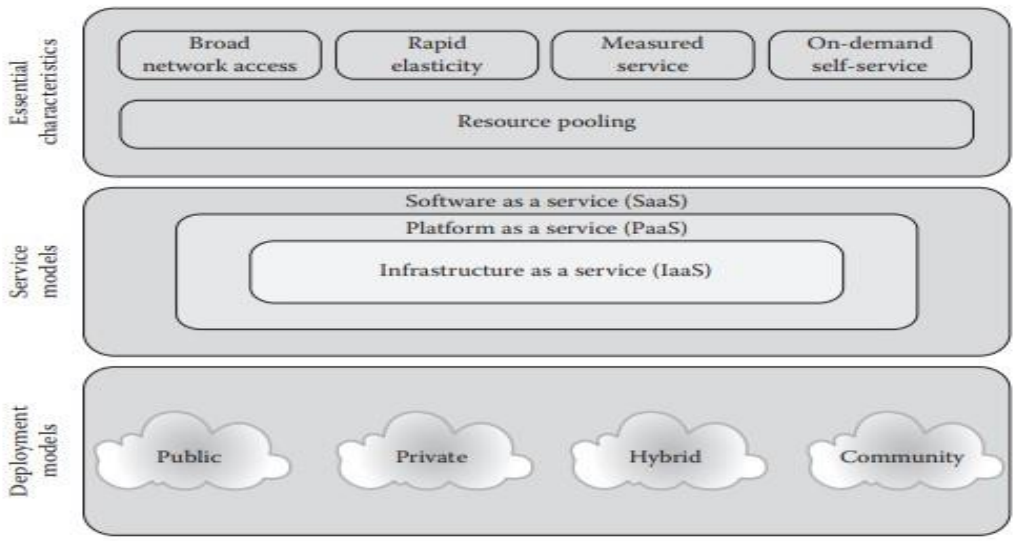


Figure 6:Les éléments de Cloud Computing [22].

- ✓ Large accès au réseau : Les services sont accessibles à partir de n'importe quel appareil connecté à internet.
- ✓ Élasticité rapide : Cela permet à l'utilisateur de déployer de nouvelles ressources ou d'ajuster les limitations de service de manière quasi instantanée.
- ✓ Service mesuré : Ce service assure le contrôle et l'optimisation des ressources grâce à la mesure et à la surveillance automatique.
- ✓ Libre-service : Les consommateurs de services sur demande peuvent utiliser les services Web pour accéder automatiquement aux ressources informatiques sur demande.
- ✓ Mise en commun des ressources : les clients peuvent partager une réserve de ressources informatiques avec d'autres clients, permettant ainsi une utilisation plus efficace.

Les services de cloud se divisent en trois couches et chacun présente des caractéristiques différentes. Selon le NIST ils sont: (1) Infrastructure as a Service, (2) Platform as a Service, et

(3) software as a Service. Cette architecture en couches où les services d'une couche supérieure peuvent être composés de services de la couche sous-jacente [23].

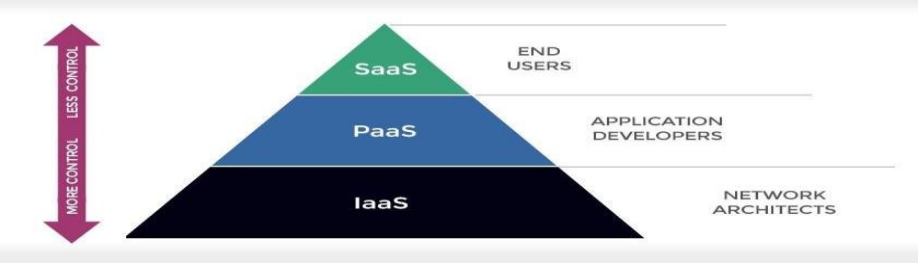


Figure 7:Représentation des différentes couches de cloud [19].

Les couches du cloud [8] :

- **Software as a Service (SaaS) :** Le consommateur peut héberger le logiciel sur l'infrastructure cloud et fournir des services logiciels à d'autres consommateurs sur la base d'un abonnement.
- **Platform as a Service (PaaS):** Le consommateur peut développer son application sur une plateforme configurée sur une infrastructure cloud. Le consommateur se débarrasse de l'achat et de la gestion des licences.
- **Infrastructure as a Service (IaaS) :** Le consommateur peut obtenir les infrastructures informatiques, de stockage et d'E/S nécessaires en peu de temps, avec une évolutivité et une élasticité.

Le cloud offre quatre modèles de déploiement [8]:

- **Le Cloud Public:** le cloud public offre un environnement multi-locataire. L'infrastructure cloud est partagée entre tous les abonnés. La sécurité est le défi majeur dans le cloud public. C'est le modèle de déploiement le moins cher parmi les quatre.
- **Le Cloud Privé :** Le cloud privé offre des ressources cloud exclusivement pour une seule organisation. La multi-locataire n'est pas autorisée. Il offre une sécurité plus élevée. C'est le modèle de déploiement le plus coûteux.

- **Le Cloud Hybride :** Ce modèle est développé en combinant les modèles de cloud public et privé. Ce modèle offre une meilleure sécurité que le cloud public et est plus économique que le cloud privé.
- **Le Cloud Communautaire :** Ce modèle est similaire au cloud public. Mais seules les organisations ayant une confiance mutuelle partagent les ressources cloud. La multi-locataire est autorisée uniquement entre les organisations de confiance mutuelle.

I.5.1-La sécurité dans le cloud

Pour assurer la satisfaction des mesures de sécurité dépendant des caractéristiques des différents scénarios, il n'existe pas de solution unique (pas même une définition unique du problème). Il y a plutôt différents aspects, avec des problèmes connexes, des défis et des contrôles de sécurité qui doivent être et qui peuvent trouver application dans différents scénarios [24].

On mentionne quelques-uns:

1. Protection des données au repos

Le premier problème de base est la protection des données stockées elles-mêmes. Avec les solutions actuelles, les utilisateurs ont généralement besoin de faire entièrement confiance aux fournisseurs de cloud. En fait, bien que les fournisseurs de services cloud appliquent des mesures de sécurité aux services qu'ils offrent, ces mesures leur permettent d'avoir un accès complet aux données. Lorsque la confidentialité des données doit être garantie même aux vis-à-vis du fournisseur, d'autres solutions doivent être envisagées. Les solutions pour protéger la confidentialité dans ce scénario nécessitent généralement de crypter les données avant de les diffuser aux fournisseurs de cloud [25].

2. Accès précis aux données

Le maintien de la confidentialité des données même par rapport aux fournisseurs qui les stockent ou les traitent implique, lorsque les données sont protégées par cryptage, que les fournisseurs ne peuvent pas déchiffrer les données pour l'exécution de la requête. Dans les applications où l'accès précis, généralement l'exécution des requêtes, doit être pris en charge, les requêtes doivent ensuite être évaluées sur les données chiffrées elles-mêmes. Il y a deux approches pour fournir cette capacité. La première approche consiste à effectuer des requêtes directement sur les données cryptées, lorsqu'une telle capacité est rendue possible par des techniques cryptographiques spécifiques (par exemple, le chiffrement homomorphe). La

deuxième approche consiste à joindre aux données chiffrées certaines métadonnées (index) qui sont ensuite utilisées pour la recherche et l'exécution des requêtes [25].

3. Accès sélectif aux données

Lorsque des données sont stockées dans le cloud, le problème se pose de savoir comment imposer des restrictions de contrôle d'accès. Par exemple, certains services de stockage en cloud (p. ex., Amazon S3 et Google Cloud Storage) prennent en charge la définition des listes de contrôle d'accès pour la réglementation de l'accès aux données. L'application de cette politique de contrôle d'accès est cependant déléguée au fournisseur de cloud. Dans de nombreux cas, cette solution n'est pas possible puisque la politique de contrôle d'accès peut être confidentielle et ne doit donc pas être divulguée au fournisseur. En outre, l'externalisation du contrôle d'accès au cloud nécessite une confiance totale dans les fournisseurs dédiés, car la protection des données serait entièrement entre leurs mains. Une approche prometteuse pour déléguer le contrôle d'accès au cloud tout en n'exigeant pas une confiance totale envers les fournisseurs repose sur la combinaison du contrôle d'accès et du cryptage, c'est-à-dire le cryptage des données avec différentes clés, en fonction des autorisations qui les détiennent [25].

I.6-Conclusion

Dans ce chapitre, nous avons exploré les différents modèles de protection et mécanismes de contrôle d'accès tels que DAC, MAC, RBAC et ORBAC. Ainsi que les différentes formes de chiffrement basées sur les attributs, telles que CP-ABE et KP-ABE, nous avons également discuté de l'importance de la sécurité et de la confidentialité des données dans le cloud computing. Ensuite, nous avons examiné les éléments, modèles et concepts du cloud computing. Malgré les avantages offerts par le cloud computing, ses aspects publics et ouverts soulèvent des préoccupations en matière de sécurité et de confidentialité. Les mécanismes de contrôle d'accès et la cryptographie jouent un rôle essentiel dans la protection des données dans cet environnement.

En conclusion, OrBAC et KP-ABE peuvent être utiles pour résoudre les problèmes de sécurité liés au contrôle d'accès. Grâce à leur souplesse et leur dynamisme, ils peuvent aider à atténuer certaines des limites des modèles de contrôle d'accès traditionnels dans les environnements cloud.

Chapitre II :

Conception Et Modélisation.

II.1-Introduction

Ce chapitre détaille les étapes clés de la conception et de la modélisation de notre système basée sur les modèles OrBAC (Organization based access control) et KP-ABE (Key-Policy Attribute-Based Encryption).

Nous commençons par examiner les travaux existants. Ensuite, nous présentons notre solution proposée. Après avoir établi le contexte de notre travail, nous passons à une étude conceptuelle détaillée. Dans cette section, nous utilisons le langage de modélisation unifié, UML, pour représenter notre système et ses interactions avec les utilisateurs.

Ce chapitre a pour but de donner une compréhension approfondie de la conception et de la modélisation de notre système.

II.2- Solution proposée

Pour préserver la confidentialité et l'intégrité des données dans le cloud, il est essentiel de mettre en œuvre un contrôle d'accès rigoureux, qui offre un moyen de réglementer le système et l'accès aux données. De plus, l'utilisation de techniques de cryptage renforce considérablement la sécurité des données, que ce soit lors de leur transfert, partage ou stockage. Notre approche se base sur 2 composants principaux :

- Le modèle OrBAC utilisé pour structurer et gérer précisément les politiques de contrôle d'accès selon les rôles et les règles propres à l'organisation. Cela permettra des autorisations d'accès granulaires adaptées aux besoins de sécurité.
- Le chiffrement par attribut KP-ABE qui nous permettra de chiffrer les données en fonction de leurs attributs descriptifs (attribut d'OrBAC). Cela signifie que seuls les utilisateurs possédant les attributs requis pourront déchiffrer les données.

En combinant ces deux approches, nous serons en mesure d'assurer un contrôle d'accès précis et une protection accrue des données dans le cloud. Notre solution vise à fournir une architecture sécurisée et robuste pour le partage, le stockage et la gestion des données sensibles dans des environnements cloud.

Pour une meilleure compréhension de notre système proposé, nous allons présenter deux schémas généraux. Le premier schéma explique l'intégration d'OrBAC avec KP-ABE, détaillant la relation et l'interaction entre ces deux composants clés de notre système. Cela est

essentiel pour comprendre comment les deux approches sont combinées pour former un contrôle d'accès puissant.

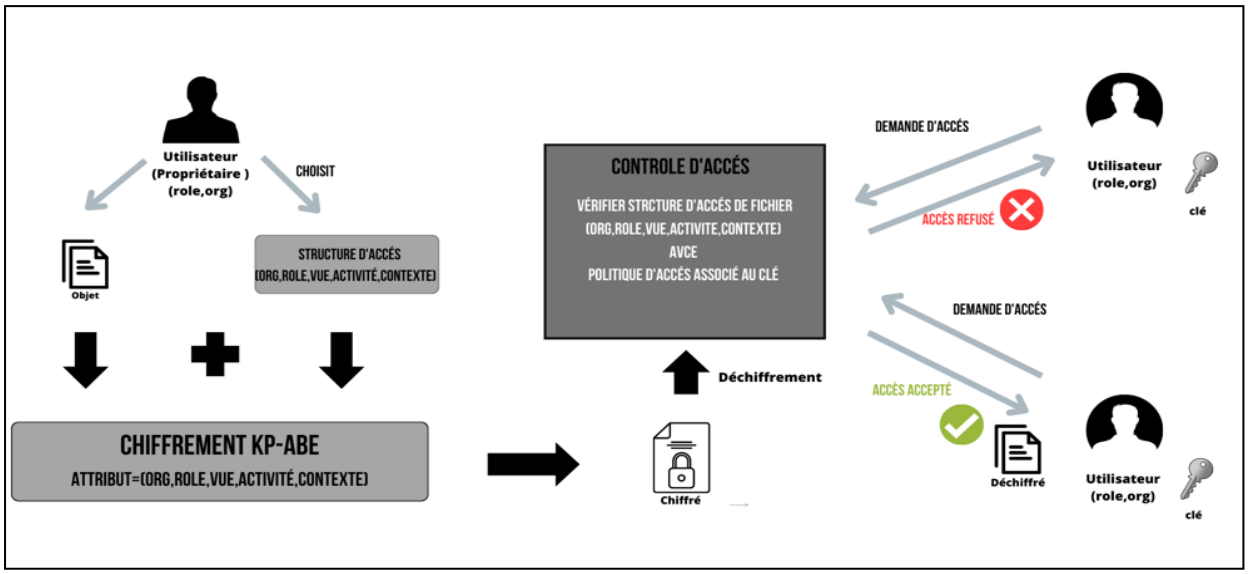


Figure 8:Intégration OrBAC avec KP-ABE.

Dans ce schéma, considérons un utilisateur qui possède un fichier spécifique, il est responsable de définir la structure d'accès qui est formule selon le système d'attributs d'OrBAC, ce fichier avec la structure d'accès entre dans la phase de chiffrement, dans cette phase le système chiffre le fichier en utilisant les attributs spécifies dans la structure d'accès.

Par la suite lorsque d'autres utilisateurs tentent d'accéder à ce fichier, ils sont soumis à un processus de vérifications, ce processus vérifier si leurs clés privées contiennent les attributs de fichier si une correspondance est trouvé le système accorde l'accès sinon l'accès est refusé.

Le deuxième schéma donne une vue globale de l'ensemble du système montrant comment les différents composants interagissent et travaillent ensemble pour fournir un contrôle d'accès sécurisé et efficace. Au cœur du système se trouve le processus de cryptage qui intègre les principes de l'OrBAC et du KP-ABE.

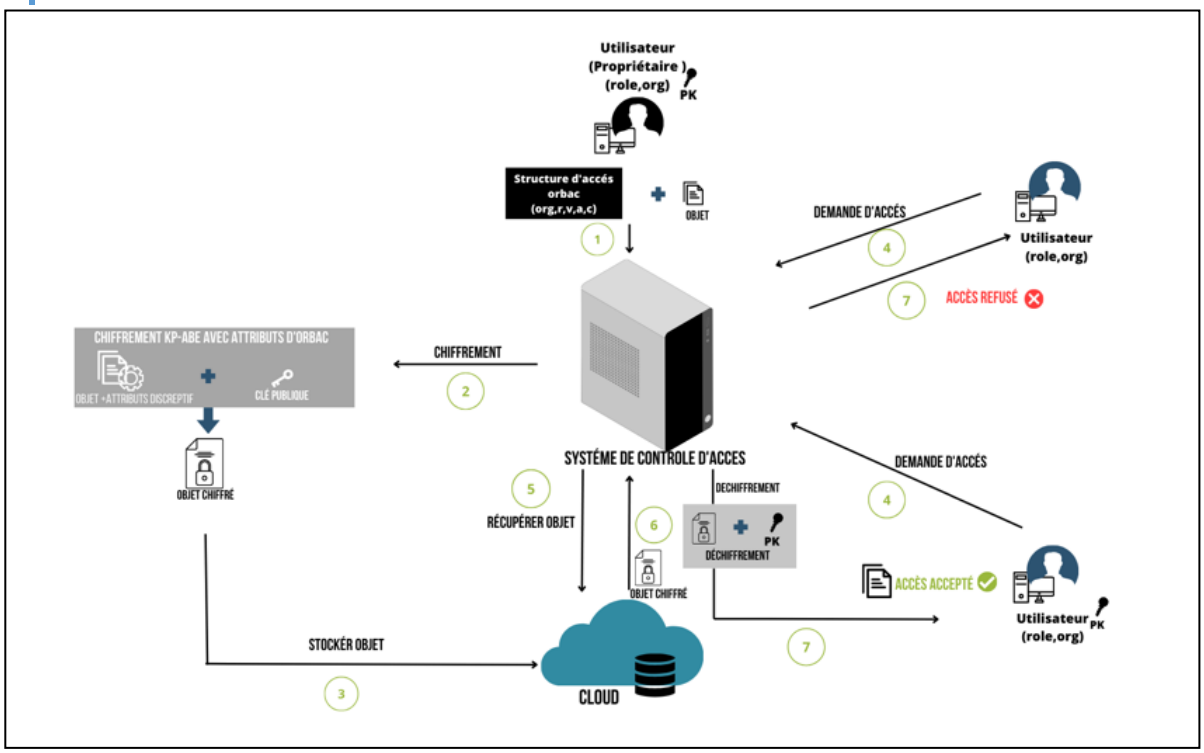


Figure 9:Schéma de système.

Les attributs du modèle OrBAC sont utilisés comme attributs dans le chiffrement KP-ABE. Cela implique l'utilisation des quatre algorithmes clés de KP-ABE:

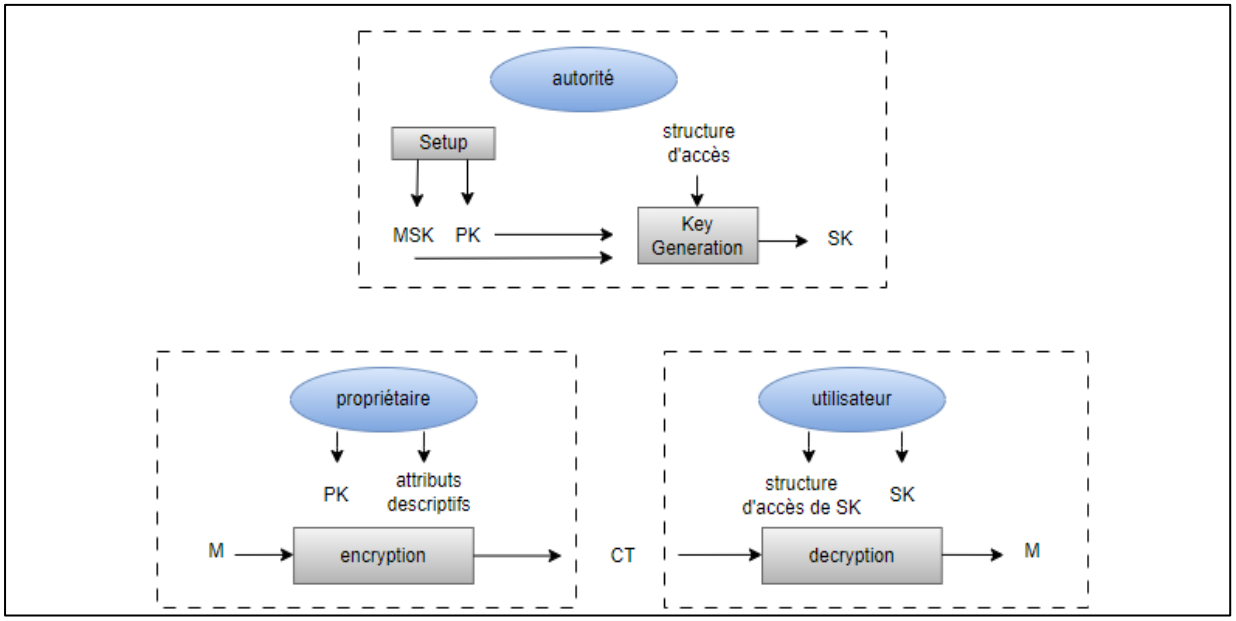


Figure 10:Les algorithmes de KP-ABE.

– Setup() : Il produit (PK, MSK), où PK désigne la clé publique et MSK la clé principale.

– KeyGen(PK,MSK,α) : L'algorithme de génération de clé prend comme entrée la structure d'accès d'un utilisateurαet la clé principale MSK et la clé publique PK. Il affiche la clé privée de l'utilisateur SK.

– Encryption(PK,M,β) : L'algorithme de chiffrement prend comme entrée un message M , l'ensemble d'attributs β et la clé de chiffrement PK. Il produit le chiffrement CT avec la politique d'accès β.

– Decryption(CT,SK) : L'algorithme de déchiffrement prend comme entrée un texte chiffré CT qui est supposé être chiffré sous l'ensemble d'attributsβet la clé privée SK pour la structure d'accèsα. Il affiche le message M si l'ensemble d'attribut β satisfait la structure d'accèsα,et erreur sinon.

II.3-Etude conceptuelle

Avant la phase d'implémentation de toute application, une étude et une conception complète du système sont cruciales. Nous avons choisi d'utiliser le langage de modélisation unifié (UML), un outil puissant pour représenter graphiquement et comprendre des systèmes complexes. Nos diagrammes UML incluent des diagrammes de cas d'utilisation des diagrammes de séquence et des diagrammes de classe, chacun offrant une perspective différente sur la structure et les interactions du système.

II.3.1-Diagramme de cas d'utilisation

Les cas d'utilisation contiennent des informations détaillées sur le système, les utilisateurs du système, les relations entre le système et les utilisateurs et le comportement requis du système [26].

II.3.1.1-Diagramme de cas d'utilisation relatif à l'utilisateur

La figure ci-dessous représente le diagramme de cas d'utilisation d'un utilisateur dans le système.

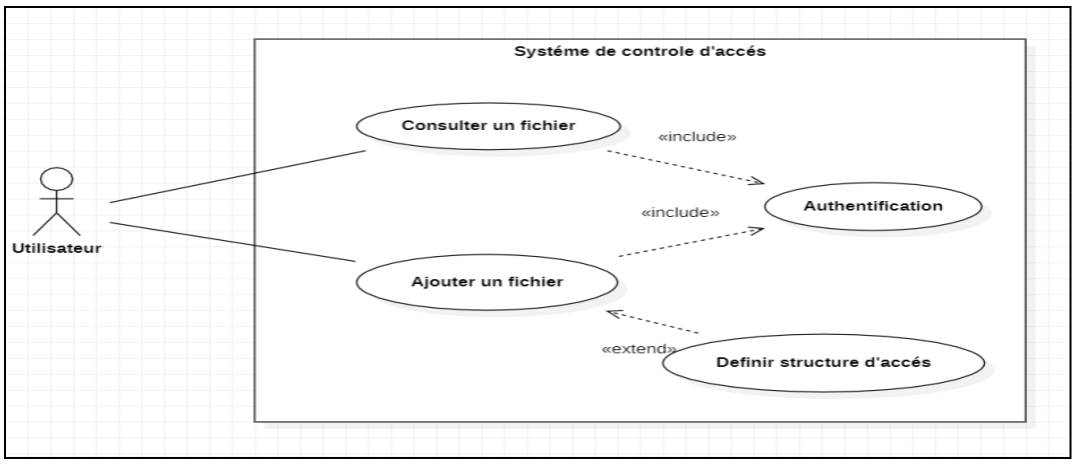


Figure 11:Cas d'utilisation de l'utilisateur.

Description de cas d'utilisation d'un utilisateur

Acteurs	Cas d'utilisation	Description
Utilisateur	• S'authentifier	L'authentification avec le nom d'utilisateur et le mot de passe est obligatoire
	• Ajouter un Fichier	L'utilisateur ajoute un fichier au système pour le stocker dans le cloud.
	• Consulter un Fichier	L'utilisateur accède à un fichier stocké dans le cloud pour le consulter ou le télécharger.
	• Définir structure d'accès	Avant de stocker son fichier, l'utilisateur définit une structure d'accès pour ce dernier.

Tableau 2:Description de cas d'utilisation de l'utilisateur.

II.3.1.2-Diagramme de cas d'utilisation relatif à l'administrateur

L'administrateur est le responsable de toute gestion dans le système, il assure le bon fonctionnement de l'application, son rôle regroupe plusieurs responsabilités.

La figure ci-dessous représente le diagramme de cas d'utilisation de l'administrateur de notre application

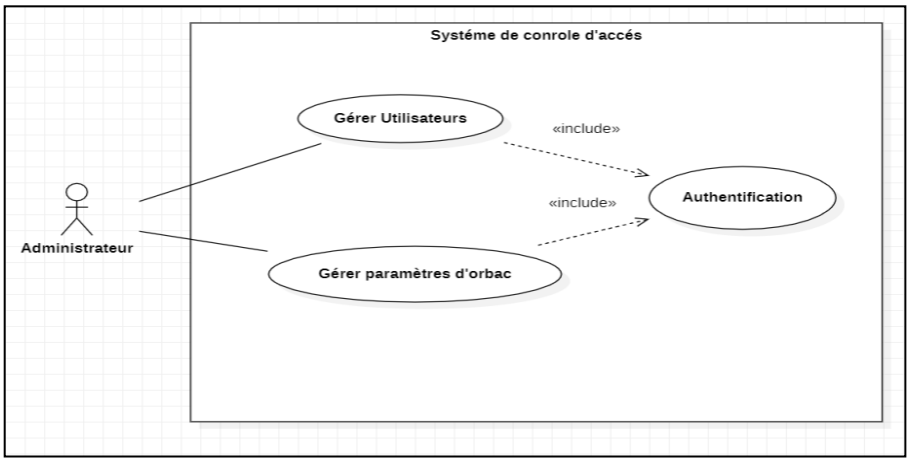


Figure 12:Cas d'utilisation de l'administrateur.

Description de cas d'utilisation de l'administrateur

acteur	Cas d'utilisation	description
administrateur	• S'authentifier	L'authentification avec le nom d'utilisateur et le mot de passe est obligatoire.
	• Gérer les utilisateurs	L'administrateur a la capacité d'ajouter, de modifier et de supprimer les utilisateurs. Cela comprend la gestion des informations de profil et des privilèges d'accès des utilisateurs.
	• Gérer les paramètres d'OrBAC	L'administrateur est capable d'ajouter, de supprimer, et de modifier les attributs d'OrBAC, ainsi que de gérer les permissions.

Tableau 3:Description des cas d'utilisation de l'administrateur.

II.3.2-Diagramme de séquence

Un diagramme de séquence est un type de diagramme d'interaction qui expose en détail la façon dont les opérations sont réalisées, notamment quels messages sont envoyés et à quel moment. Ce diagramme représente la séquence de messages transmis entre des objets. Il peut également représenter les structures de contrôle entre des objets [26].

Nous allons présenter un ensemble de diagrammes de séquence décrivant les principales interactions dans notre système.

II.3.2.1-Chiffrement

Après l'étape d'authentification, l'utilisateur choisit le fichier qui veut uploader il choisit une politique d'accès pour spécifier qu'il peut accéder à son fichier. Après, il demande au système de chiffrer le fichier.

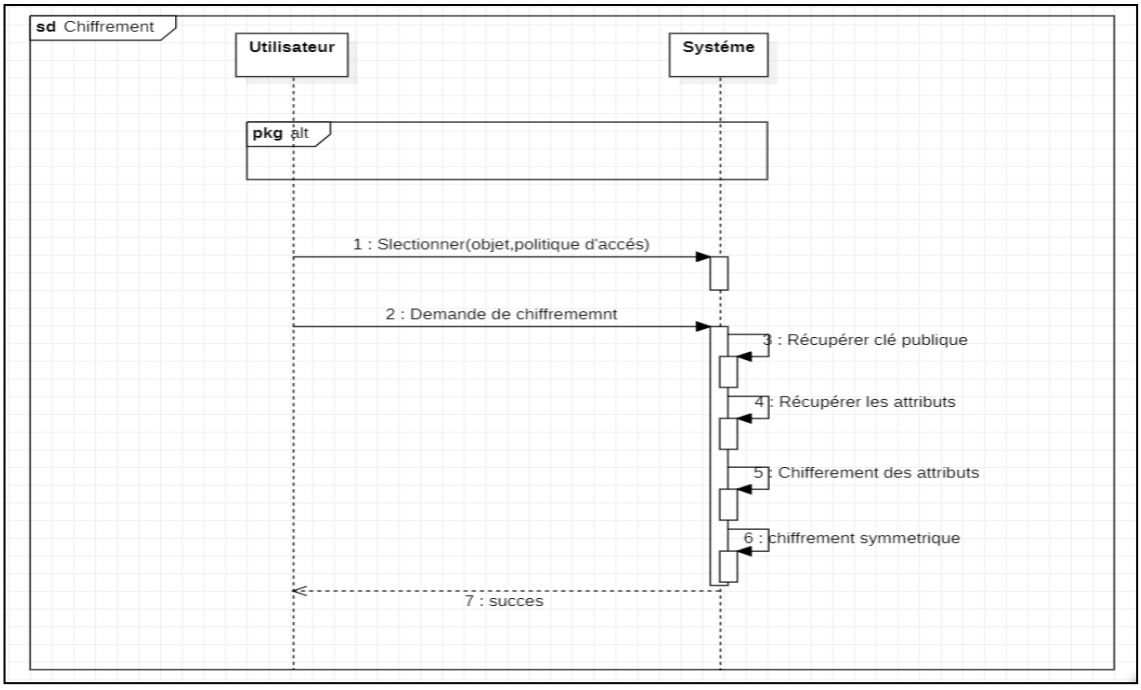


Figure 13:Diagramme de séquence « chiffrement ».

II.3.2.2-Déchiffrement

Pour que l'utilisateur puisse déchiffrer un fichier, il doit d'abord s'authentifier, ensuite il doit choisir le fichier qu'il veut consulter après comparaison entre ces informations et la politique de sécurité associée avec le fichier qu'il souhaite accéder, si les informations sont compatibles l'utilisateur accède. Sinon, un message d'erreur est affiché.

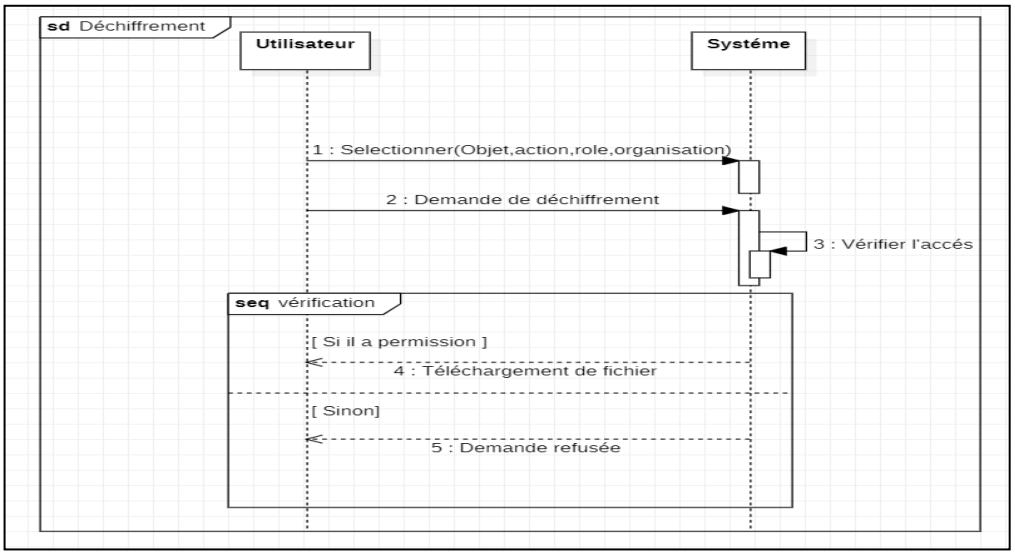


Figure 14:Diagramme de séquence « déchiffrement ».

Dans le cadre du rôle administrateur, nous avons choisi de montrer principalement le processus d'ajout d'un nouvel utilisateur et de nouvelle permission, qui sont des tâches administratives essentielles. Cependant, il convient de noter que les fonctions d'un administrateur vont bien au-delà de ces tâches. En effet, un administrateur peut gérer l'ensemble des utilisateurs du système, y compris la possibilité de modifier et de supprimer des comptes existants, ainsi que la gestion des attributs OrBAC.

II.3.2.3-L'ajout d'un utilisateur

Le diagramme suivant illustre le processus d'ajout d'un nouvel utilisateur par un administrateur dans notre système. Il met en évidence l'interaction entre l'administrateur et le système.

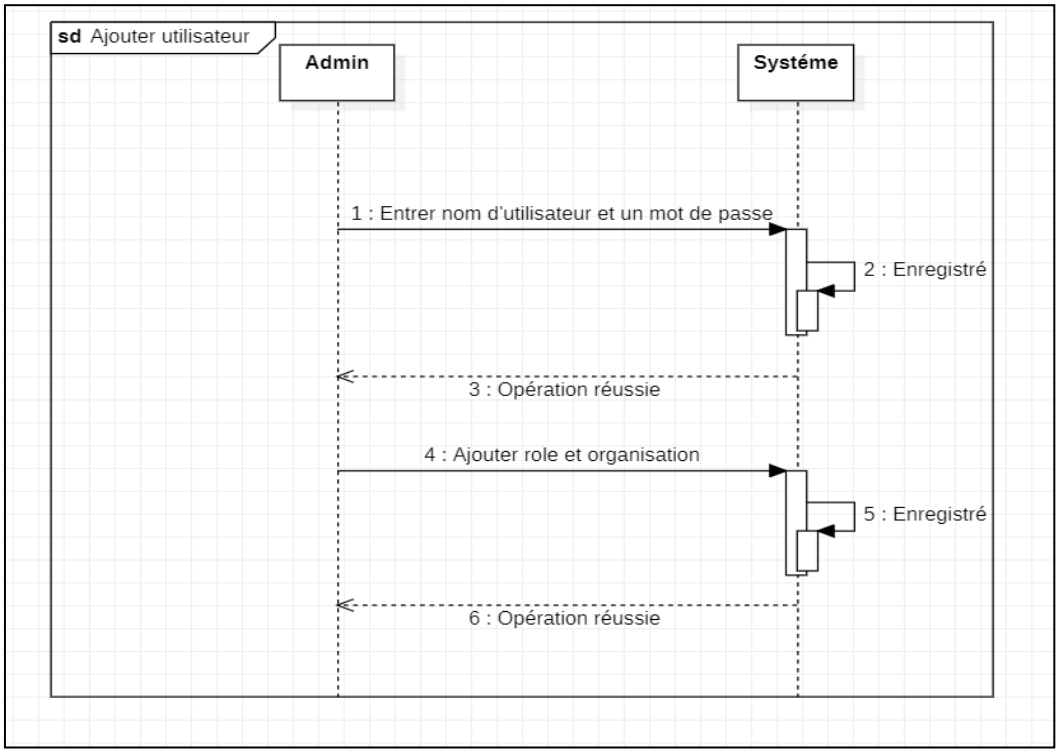


Figure 15:Diagramme de séquence « ajouter utilisateur ».

II.3.2.4-L'ajout d'une permission

L'administrateur est chargé de la gestion d'OrBAC, il peut également ajouter une nouvelle permission, ce diagramme illustre le processus.

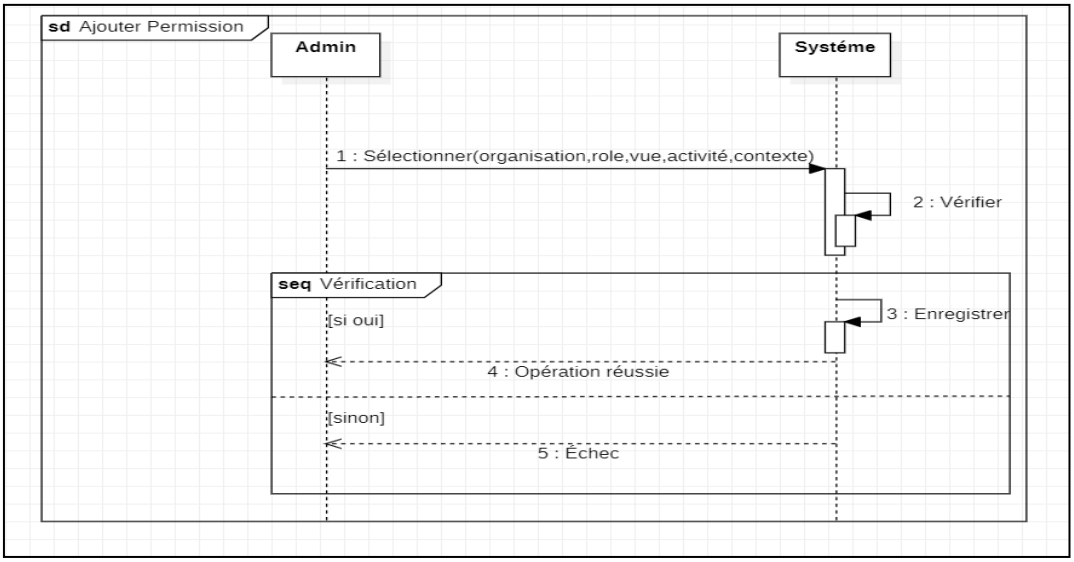


Figure 16:Diagramme de séquence « ajouter une permission ».

II.3.3-Diagramme de classe :

Les diagrammes de classe sont utiles à de nombreuses étapes de la conception d'un système. Lors de l'étape d'analyse, un diagramme de classe peut aider à comprendre les exigences de domaine de problème et à identifier ses composants [26].

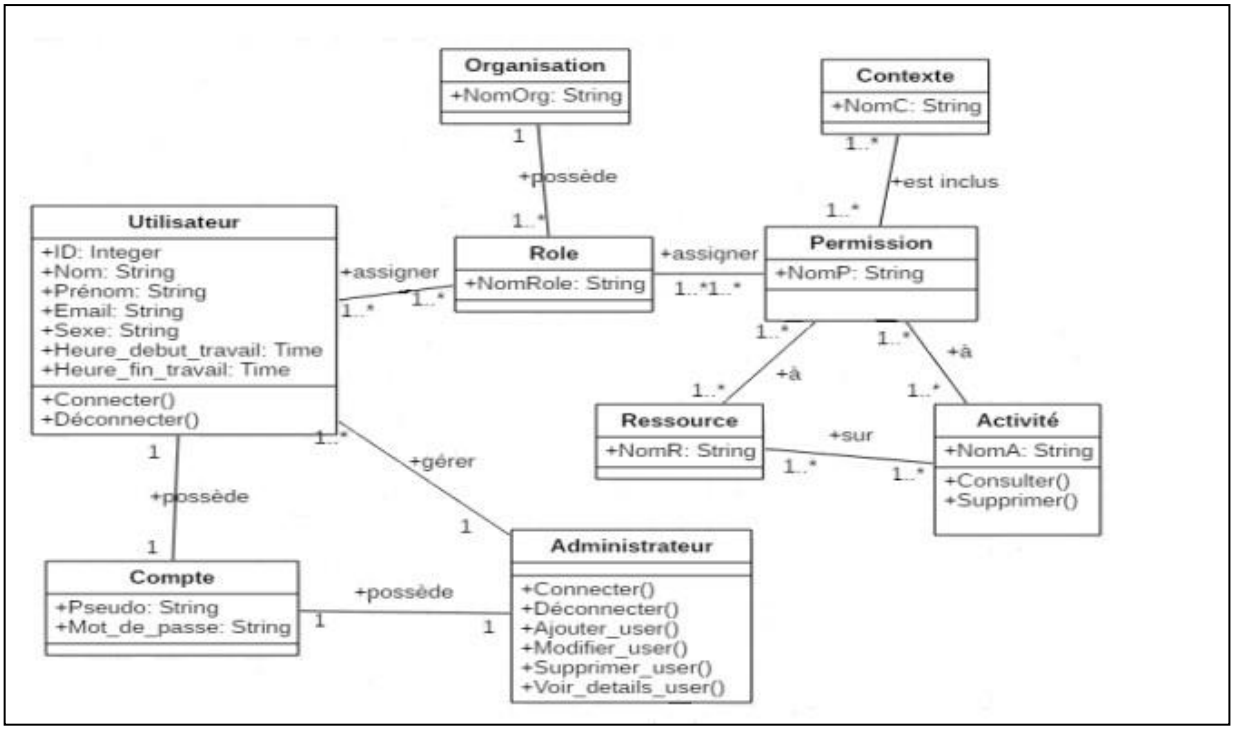


Figure 17:Diagramme de classe.

II.4-Conclusion

Dans ce chapitre, nous avons exploré la solution proposée, soulignant l'intégration d'OrBAC et de KP-ABE comme éléments clés de notre approche de sécurité. En utilisant le langage UML, nous avons modélisé notre système pour visualiser sa structure et son fonctionnement de manière claire.

Dans le dernier chapitre, nous passons à la phase d'implémentation et de déploiement de notre solution. Cette phase nous permettra de réaliser et de vérifier l'efficacité de notre approche dans un environnement réel.

Chapitre III:

Réalisation Et Expérimentation.

III.1-Introduction

Ce chapitre se concentre sur la réalisation de la solution proposée dans le chapitre précédent. Il aborde la phase de mise en œuvre de notre application, en intégrant les différents éléments décrits précédemment.

La première partie de ce chapitre met en avant l'environnement de développement utilisé pour réaliser la solution. Ensuite, nous examinerons en détail les différentes fonctionnalités implémentées, illustrées par des captures d'écran des interfaces graphiques principales. Enfin, la dernière partie est consacrée aux tests et à l'évaluation des résultats.

III.2-Environnement de développement

L'environnement de développement est l'ensemble des outils et des ressources nécessaires pour développer, tester et déployer des logiciels.

III.2.1-Les langages de programmation

1-HTML

HTML, en plein langage de balisage hypertexte, un système de formatage pour afficher le matériel récupéré sur Internet. Chaque unité de récupération est connue sous le nom de page Web (du World Wide Web), HTML est le langage de balisage pour l'encodage des pages Web [27].

2-CSS

Est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel que HTML ou XML [27].

3-Langage Python

Python est un langage de programmation interprété, dynamique et orienté objet. Il a été créé en 1991 par Guido van Rossum Set est aujourd'hui l'un des langages de programmation les plus populaires. Python est facile à apprendre et à lire, ce qui en fait un choix populaire pour les débutants en programmation [27].

III.2.2-Framework Django

Django est une infrastructure d'application (aussi appelée framework) côté serveur extrêmement populaire et dotée de beaucoup de fonctionnalités. Construit par des développeurs expérimentés. Il prend soin d'une grande partie des tracas du développement web, il est gratuit, à une communauté active et une documentation exhaustive. Django suit le modèle de conception MVC (Modèle-Vue-Contrôleur) et est basé sur le principe du DRY (Ne vous répétez pas), ce qui signifie qu'il encourage la réutilisation de code [28].

III.2.3 PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle orienté objet puissant et open source qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes.PostgreSQL fonctionne sur diverses plates-formes matérielles et sous différents systèmes d'exploitation, dont Unix, Linux, FreeBSD, Solaris, Microsoft Windows et Mac OS [29].

III.2.4 IPFS

IPFS est une suite modulaire de protocoles pour l'organisation et le transfert de données, conçus à partir de la base avec les principes d'adressage de contenu et de réseau peer-to-peer. Parce qu'IPFS est open-source, il y a plusieurs implémentations d'IPFS. Alors que l'IPFS a plus d'un cas d'utilisation, son principal cas d'utilisation est la publication de données (fichiers, répertoires, etc) de manière décentralisée [30].

Parmi les implémentations les plus utilisé :

- **IPFS Desktop**

IPFS Desktop réunit un nœud IPFS, un gestionnaire de fichiers, un gestionnaire de pairs et un explorateur de contenu en une seule application facile à utiliser. Cela permet aux utilisateurs non seulement de stocker et de récupérer des fichiers, mais aussi de gérer les pairs et d'explorer le contenu dans le réseau IPFS [30].

III.2.5 Docker

Docker est une plateforme libre qui permet aux développeurs de construire, de déployer, d'exécuter, de mettre à jour et de gérer des conteneurs. Il s'agit de composants normalisés et exécutables qui combinent le code source de l'application et les bibliothèques et dépendances de système d'exploitation nécessaires pour exécuter ce code dans n'importe quel environnement [31].

Les conteneurs simplifient le développement et la livraison des applications distribuées. Ils sont devenus de plus en plus populaires à mesure que les organisations se tournent vers le développement cloud et les environnements multi-cloud hybrides [31].

III.3-Description de l'implémentation

Les unités opérationnelles de notre système sont liées, en intégrant l'algorithme KP-ABE et le modèle OrBAC créant un modèle solide pour notre système de contrôle d'accès. Chacun de ces composants ou sous-systèmes, contribue à l'efficacité opérationnelle globale du système en apportant des compétences spéciales à la portée du problème en question. Ces sous-systèmes ont été développés en utilisant le langage de programmation Python, le framework Django et la bibliothèque de cryptographie charm. Les détails de chaque sous-système sont examinés dans la section suivante, qui met également en évidence la façon dont chacun interagit avec l'autre dans l'architecture du système plus large.

III.3.1-Mise en œuvre du modèle OrBAC

OrBAC est un modèle de contrôle d'accès qui étend le contrôle d'accès basé sur les rôles (RBAC) en introduisant le concept d'organisation. Dans OrBAC, les permissions sont définies en termes de rôles, d'activités et de vues, qui sont tous des concepts abstraits. Plus précisément, un rôle est autorisé à effectuer une certaine activité sur les objets représentés par une vue.

Dans notresolution, nous avons associé un ensemble de permissions à chaque vue définis dans le système. Lorsqu'un utilisateur tente d'accéder à un objet spécifique, le système vérifie si le rôle de l'utilisateur dispose de la permission d'accéder aux objets qui appartiennent à cette vue spécifique. Cette approche offre une flexibilité et une adaptabilité considérables, permettant une gestion efficace des permissions au sein de structures organisationnelles complexes.

Pour chaque attribut d'OrBAC nous avons créé un modèle. Tous les modèles ont été enregistrés dans le fichier admin.py de notre application Django, ce qui facilite la gestion des entrées de base de données via l'interface d'administration intégrée de Django. Et nous avons présenté la permission comme suivant :

```python
class Permission(models.Model):
    name = models.CharField(max_length=150, verbose_name="Name")
    organization = models.ForeignKey(Organization, on_delete=models.CASCADE, related_name='permissions')
    role = models.ForeignKey(Role, on_delete=models.CASCADE, related_name='permissions')
    activity = models.ForeignKey(Activity, on_delete=models.CASCADE, related_name='permissions')
    view = models.ForeignKey(View, on_delete=models.CASCADE, related_name='permissions')
    context = models.ForeignKey(Context, on_delete=models.CASCADE, related_name='permissions')

    class Meta:
        verbose_name = "Permission"
        verbose_name_plural = "Permissions"

    def __str__(self) -> str:
        return f"{self.label} -- {self.pk}"
```

Figure 18:Classe Permission.

III.3.2-Mise en œuvre de l'algorithme KP-ABE

Le chiffrement basé sur les attributs à politique de clé (KP-ABE) est un type de chiffrement basé sur les attributs (ABE), où le chiffrement est effectué à l'aide d'attributs. Dans le KP-ABE, la clé qui est utilisée pour déchiffrer l'information est créée sur la base d'une structure d'accès qui est liée à certains attributs. Cela signifie qu'une clé de déchiffrement ne déchiffre un texte chiffré que si les attributs associés au texte chiffré correspondent à la structure d'accès de la clé de déchiffrement.

Dans notre solution les attributs du modèle OrBAC sont utilisés comme des attributs de KP-ABE, c'est-à-dire la structure d'accès de la clé et les attributs associés au texte chiffré sont des attributs du modèle OrBAC. Nous avons implémenté l'algorithme KP-ABE à l'aide de la bibliothèque **charm**, cette bibliothèque fournit un ensemble de primitives cryptographiques qui sont extrêmement utiles dans la conception de systèmes cryptographiques complexes.

Nous avons utilisé la classe **KPabe**, qui hérite de la classe **ABEnc** de la bibliothèque **charm**.**KPabe** met en œuvre le système de chiffrement basé sur les attributs (KP-ABE). Pour illustrer cette mise en œuvre, des captures d'écran de l'implémentation seront présentées dans les figures suivantes :

```python
from charm.toolbox.pairinggroup import PairingGroup,ZR,G1,G2,GT,pair
from charm.toolbox.secretutil import SecretUtil
from charm.toolbox.ABEnc import ABEnc

debug = False
class KPabe(ABEnc):
    """
    >>> from charm.toolbox.pairinggroup import PairingGroup,GT
    >>> group = PairingGroup('MNT224')
    >>> kpabe = KPabe(group)
    >>> (master_public_key, master_key) = kpabe.setup()
    >>> policy = '(ONE or THREE) and (THREE or TWO)'
    >>> attributes = [ 'ONE', 'TWO', 'THREE', 'FOUR' ]
    >>> secret_key = kpabe.keygen(master_public_key, master_key, policy)
    >>> msg=group.random(GT)
    >>> cipher_text = kpabe.encrypt(master_public_key, msg, attributes)
    >>> decrypted_msg = kpabe.decrypt(cipher_text, secret_key)
    >>> decrypted_msg == msg
    True
    """
```

Figure 19:Classe KPABE.

Le processus de chiffrement a été réalisé en utilisant les outils de la bibliothèque **charm**, notamment la classe **HybridABEnc**, qui nous a permis d'utiliser l'algorithme **AES** pour le chiffrement symétrique, comme le montre la figure suivante :

```python
def kpabe_encrypt(message, objet):
    # Vérifie si la vue a des permissions
    if len(objet.view.permissions.all()) < 1:
        raise EmptyResultSet('Aucune permission disponible !')

    # Ajoute toutes les permissions de la vue aux attributs
    attributes = []
    for permission in objet.view.permissions.all():
        attributes.extend([permission.organization.name, permission.role.label, permission.activity.label, permission.view.label, permission.context.label])
        for action in permission.activity.actions.all():
            attributes.append(action.label)

    # Appelle la fonction encrypt de l'instance HybridABEEnc pour utiliser AES avec KP-ABE
    return kpsetup.hybridKPABE.encrypt(kpsetup.pk, message, attributes)
```

Figure 20:Chiffrement KP-OrBAC.

Cette fonction montre comment les attributs d'un modèle OrBAC peuvent être utilisés dans un schéma KP-ABE. Chaque permission de la vue apporte ses attributs au processus de chiffrement, garantissant que seules les parties ayant la structure d'accès correspondant peuvent déchiffrer le texte chiffré.

Le processus de déchiffrement fonctionne d'une façon similaire à la fonction de chiffrement, le texte chiffré peut être déchiffré si l'ensemble des attributs utilisés lors du chiffrement satisfait à la politique utilisée pour générer la clé privée.

```python
def kpabe_decrypt(subject,objet,action):

    # Vérifie si le sujet a des permissions
    if len(subject.role.permissions.all()) < 1:
        raise EmptyResultSet("Aucune permission disponible !")
    # Récupère la première permission
    permission=subject.role.permissions.all()[0]

    # Crée la structure d'acces de sujet avec ses permissions
    # Cette politique doit être satisfaite par les attributs du chiffrement (associé au fichier) pour réussir le déchiffrement
    actions = ' or '.join([f'{action.label}' for action in permission.activity.actions.all()])

    policy=f'(({permission.organization.name} and {permission.role.label} and {permission.activity.label} and ({actions}) and {permission.view.label} and {permission.context.label})'

    for permission in subject.role.permissions.all()[1:]:
        actions = ' or '.join([f'{action.label}' for action in permission.activity.actions.all()])
        policy += f' or ({permission.organization.name} and {permission.role.label} and {permission.activity.label} and ({actions}) and {permission.view.label} and {permission.context.label})'
    policy += ')'

    #Générer la clé privée (déchiffrement) à l'aide de la politique
    privatekey = kpsetup.hybridKPABE.keygen(kpsetup.pk, kpsetup.msk, policy)

    try:
        #Désérialiser les données binaires en un objet d'appariement Charm (cipher text)
        crypted_mssg = hybrid_pickle_pairing_loads(kpsetup.GROUP_OBJ, objet)
        # Décrypter le chiffrement à l'aide de la clé privée
        message = kpsetup.hybridKPABE.decrypt(crypted_mssg, privatekey)
        # Si le décryptage a réussi, return le message déchiffré et un string vide ()
        return message, ''

    except Exception as e:
        print("Exception:", e)
        #sinon return un message vide et un string  d'erreur non autorisée
        return '', "Vous n'êtes pas autorisé à télécharger le fichier"
```

Figure 21:DéchiffrementKP-OrBAC.

III.4-Présentation de l'application (Captures d'écran clés)

Notre application commence d'abord par l'authentification des utilisateurs, nous avons deux interfaces d'authentification, une pour les utilisateurs et l'autre pour les administrateurs de système.

III.4.1-Espace d'authentification

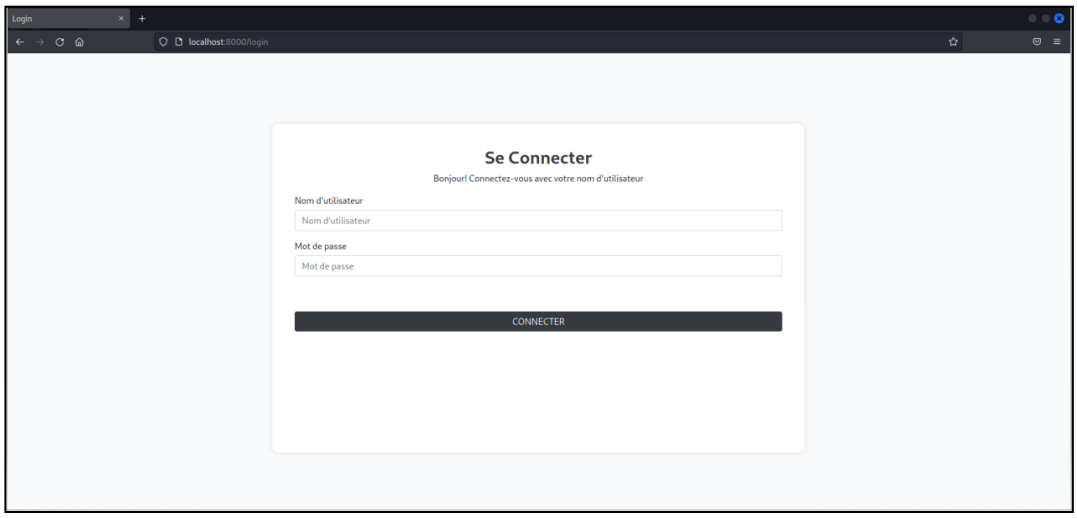


Figure 22:Interface d'authentification des utilisateurs.

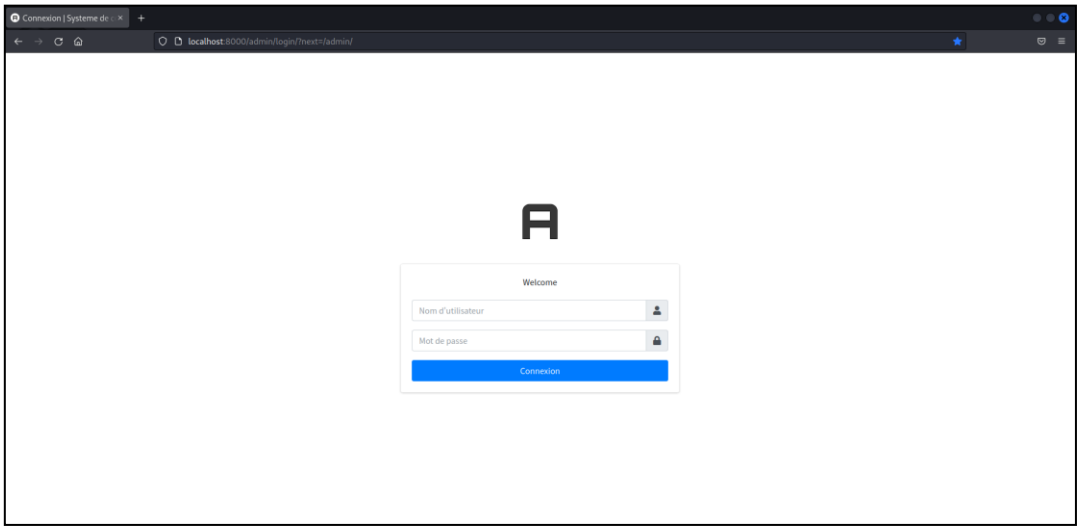


Figure 23:Interface d'authentification des administrateurs.

III.4.2-Espace administrateur

Après l'authentification, l'administrateur est redirigé vers une interface d'administration exhaustive. Cette interface comprend un ensemble complet d'outils permettant de gérer la politique d'accès de manière efficace et sécurisée. Les différentes actions disponibles dans l'interface comprennent la gestion des organisations, des activités, des rôles, des vues, des actions, des sujets, des objets, des contextes, des permissions et des utilisateurs.

1-Gestion d'OrBAC :

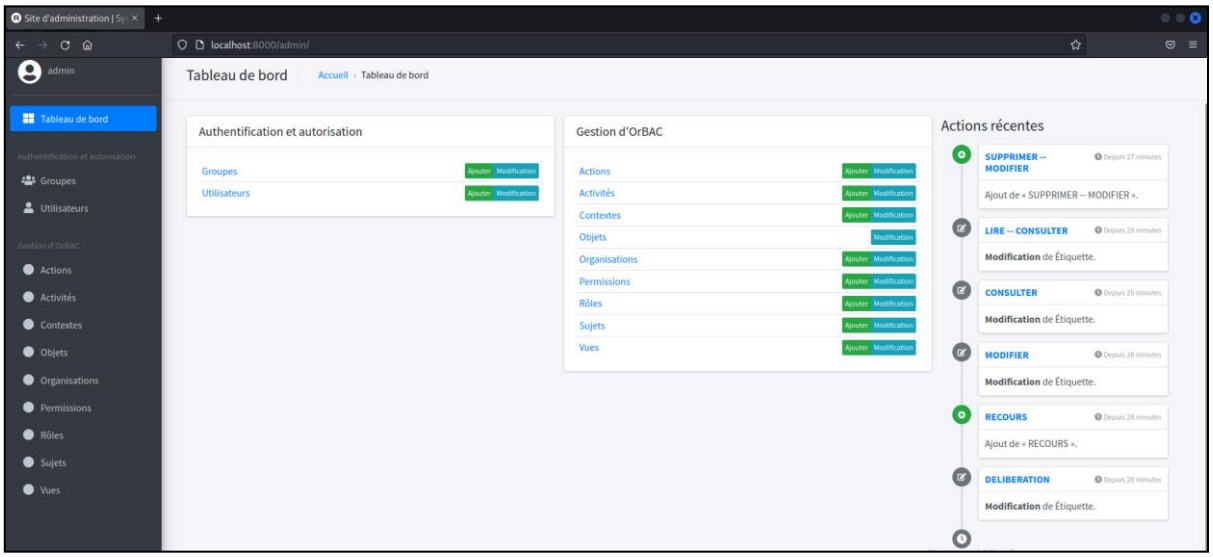


Figure 24:Espace administrateur.

L'interface d'administration offre un ensemble d'actions standardisées sur tous les attributs OrBAC - Organisations, Activités, Rôles, Vues, Actions, Sujets, Objets, Contextes, et Permissions. L'administrateur a la capacité d'ajouter, d'éditer ou de supprimer chaque attribut de manière autonome. Cela offre une flexibilité maximale pour ajuster la politique d'accès selon les besoins spécifiques de l'organisation.

Pour illustrer le fonctionnement, nous présentons ici les actions que l'administrateur peut effectuer sur l'entité 'Vue'. Il dispose d'une gamme d'options, notamment consulter les vues existantes, modifier une vue, supprimer une vue ou ajouter une nouvelle vue.

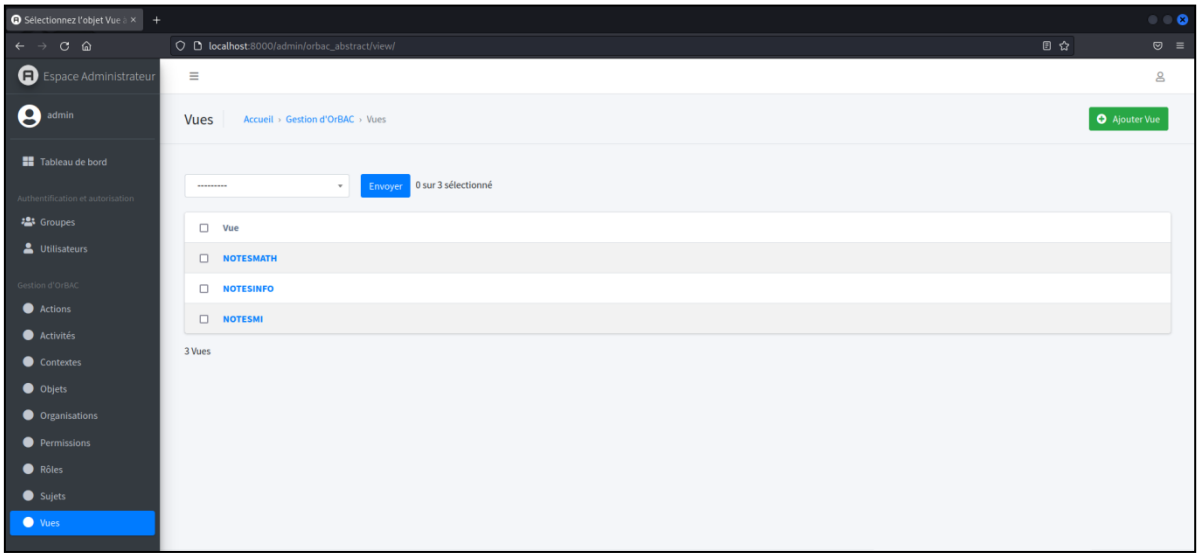
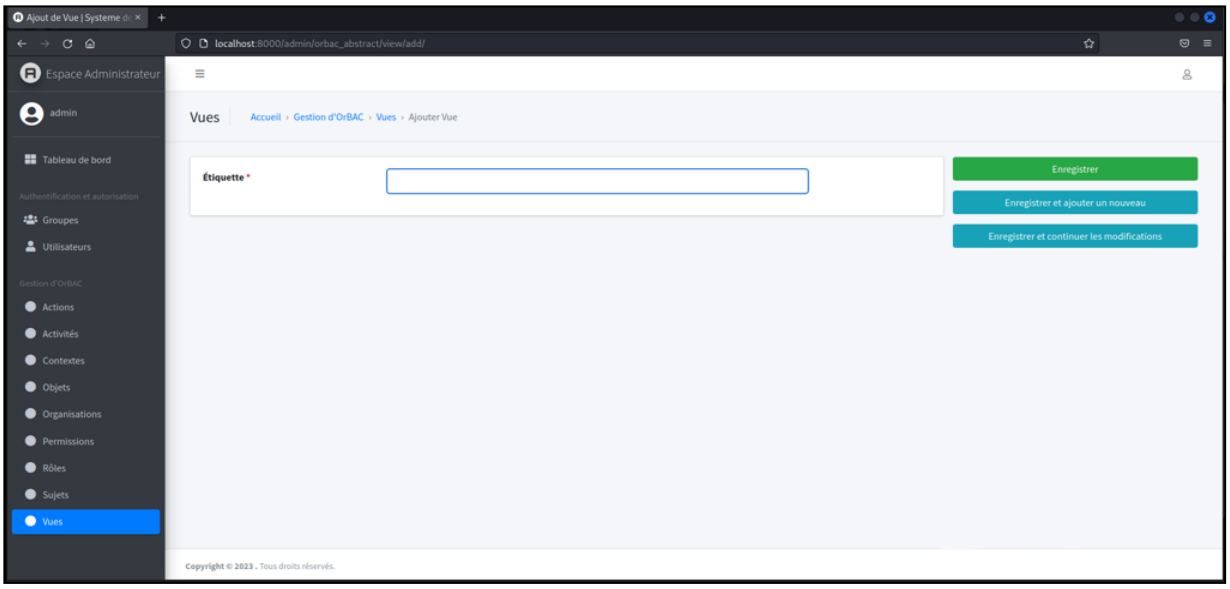


Figure 25:Gestion d'OrBAC « vue ».

Figure 26:Gestion d'OrBAC« Ajouter une nouvelle vue».



2-Gestion d'utilisateurs

L'administrateur dispose également de capacités pour gérer les utilisateurs. Ceci inclut des fonctions telles que l'ajout de nouveaux utilisateurs, la suppression d'utilisateurs existants, la modification des informations de l'utilisateur et l'attribution de rôles spécifiques aux utilisateurs.

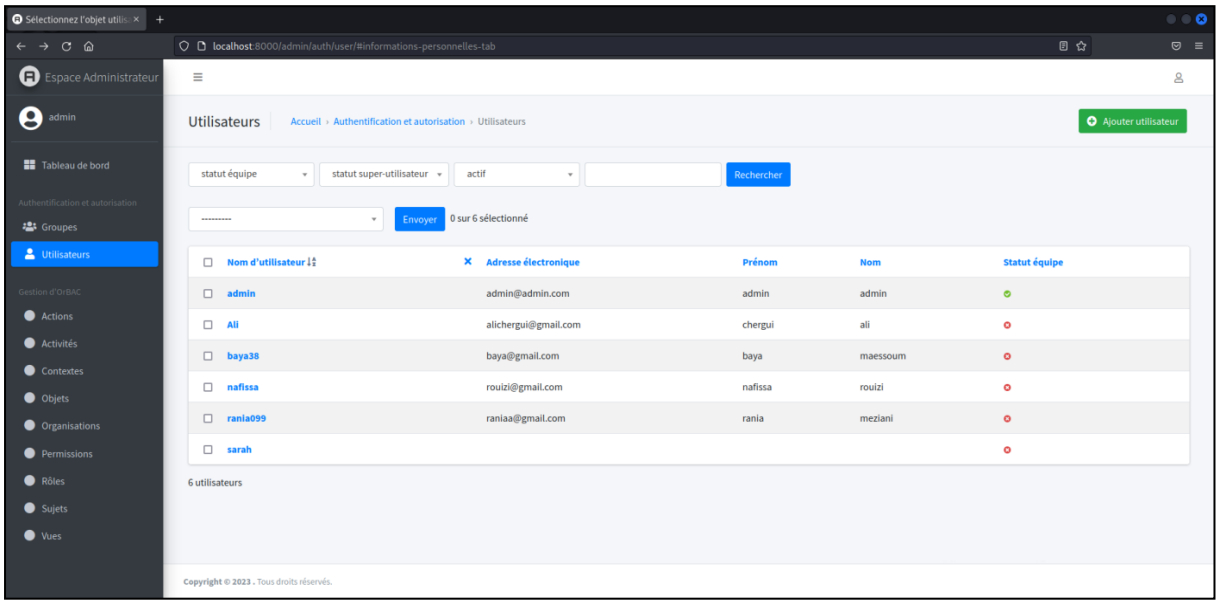


Figure 27:Gestion d'utilisateurs.

Bien que les captures d'écran présentées ne démontrent pas directement le scénario décrit précédemment, elles mettent en évidence les composants clés de l'application.

III.4.3-Espace utilisateur

L'authentification des utilisateurs nous dirige à une interface pour consulter et télécharger les fichiers décryptés. L'utilisateur choisit le fichier à consulter et l'action, s'il possède la permission d'accéder à ce fichier, le fichier est téléchargé sinon un message indiquant qu'il n'est pas autorisé sera affiché.

Pour des raisons de sécurité, nous avons ajouté deux champs (rôle et organisation) avant de demander n'importe quel fichier afin de confirmer l'identité de l'utilisateur. Ces informations sont confidentielles, et seul l'utilisateur en a connaissance. Cela aide à prévenir les tentatives d'accès par un utilisateur non autorisé.

Cas d'un utilisateur qui possède la permission :

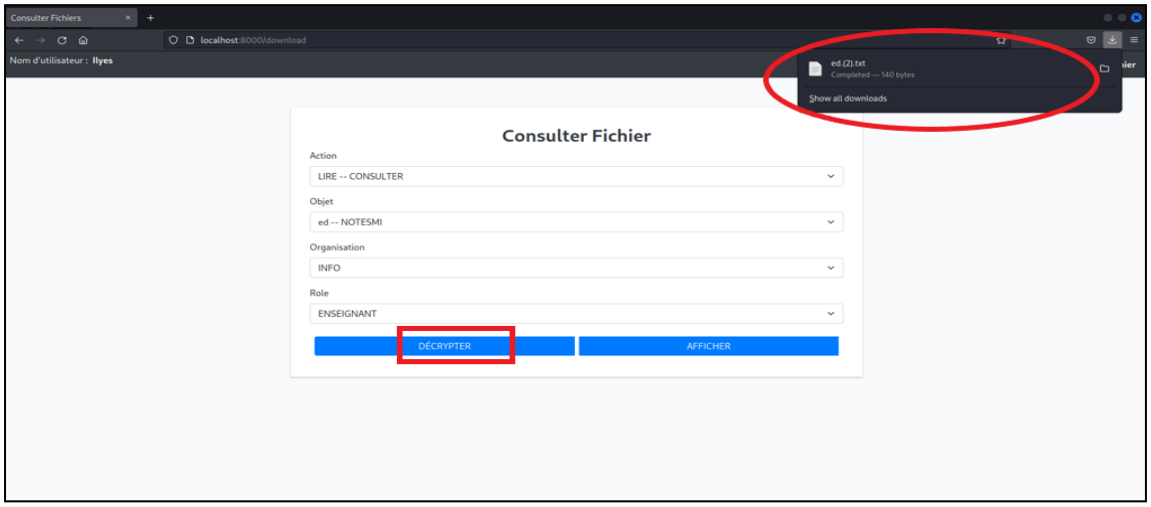


Figure 28:Demande d'accès « cas valide ».

Cas d'un utilisateur qui ne possède pas la permission :

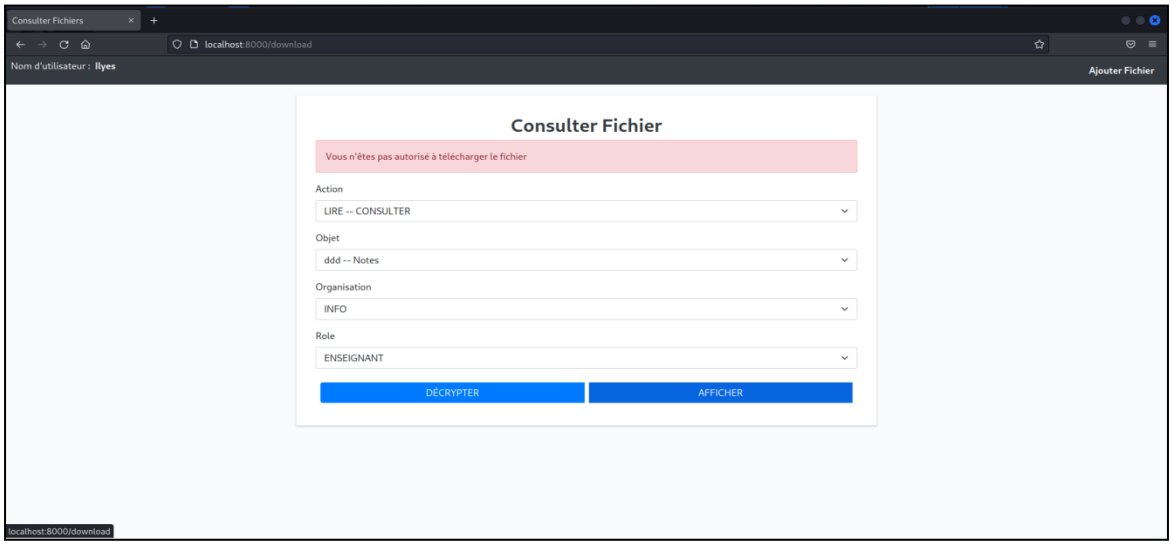


Figure 29:demande d'accès «cas invalide».

Et dans les deux cas l'utilisateur peut voir le fichier chiffré, nous avons ajouté cette option seulement pour illustrer la façon dont le fichier est stocké avant le déchiffrement.

Depuis cette interface l'utilisateur peut aller à la prochaine interface en cliquant sur le bouton 'Ajouter Fichier' qui permet d'ajouter des fichiers et les associe à une vue spécifique.

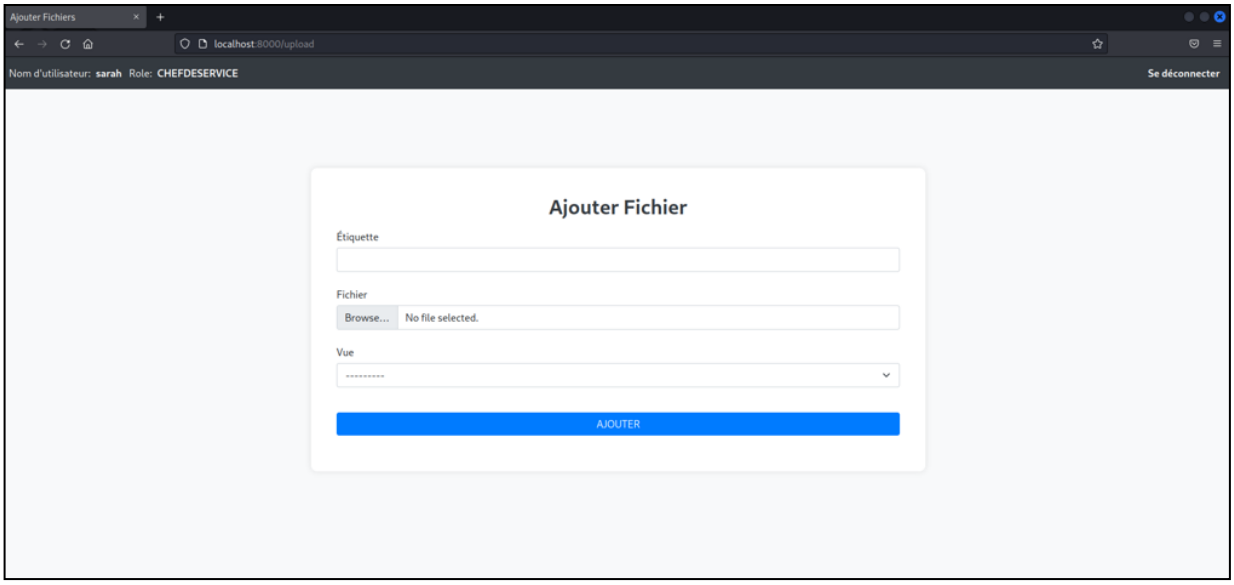


Figure 30:Interface d'ajout d'un fichier.

III.5-Publication de l'application sur le Cloud avec IPFS

Dans la partie finale de ce projet, nous avons mis notre application disponible sur le cloud en utilisant IPFS, un système de fichiers distribué et décentralisé. Nous avons placé le dossier de notre application sur IPFS, créant un identifiant de contenu (CID) unique qui peut être partagé avec les utilisateurs. Grâce à ce CID, les utilisateurs peuvent récupérer le dossier de l'application à tout moment. Les captures d'écran ci-dessous illustrent cette partie du processus.

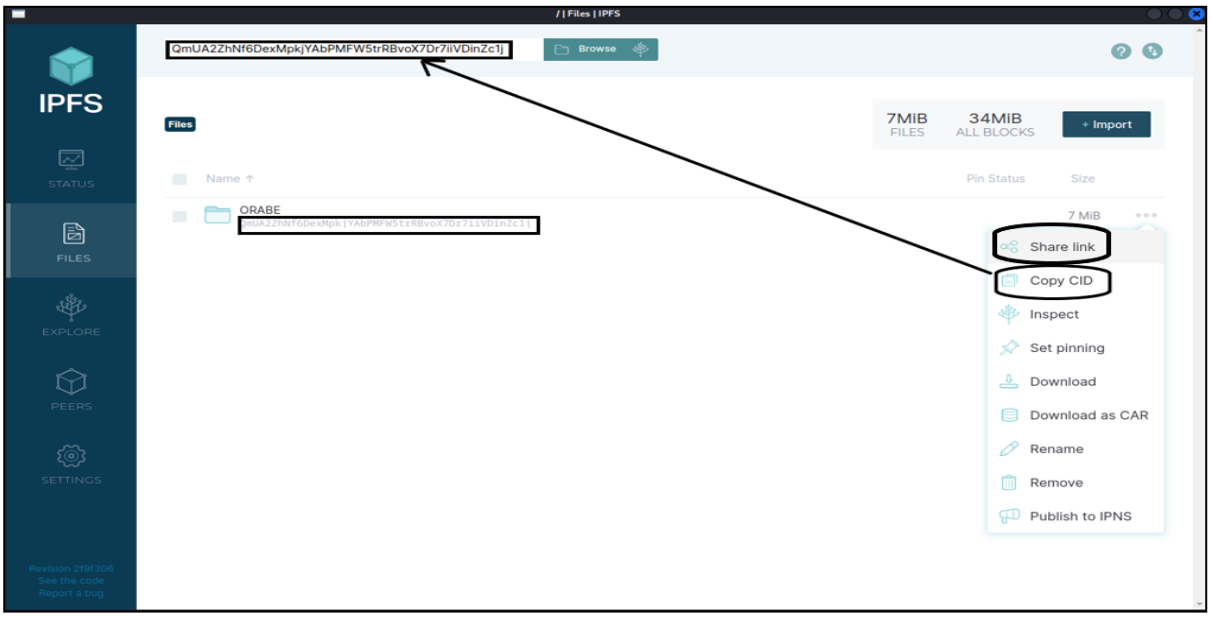


Figure 31:Publication de l'application dans cloud avec IPFS.

La figure ci-dessus met en évidence la disponibilité de notre application KP-OrBAC sur le cloud IPFS. Les utilisateurs ont la possibilité d'y accéder en utilisant le CID pour retrouver l'application.

Dans l'éventualité où un utilisateur souhaite accéder directement aux fichiers sans contrôle (impossible sans le hash de fichier), il va se retrouver face à des fichiers cryptés, comme illustre la figure suivante.

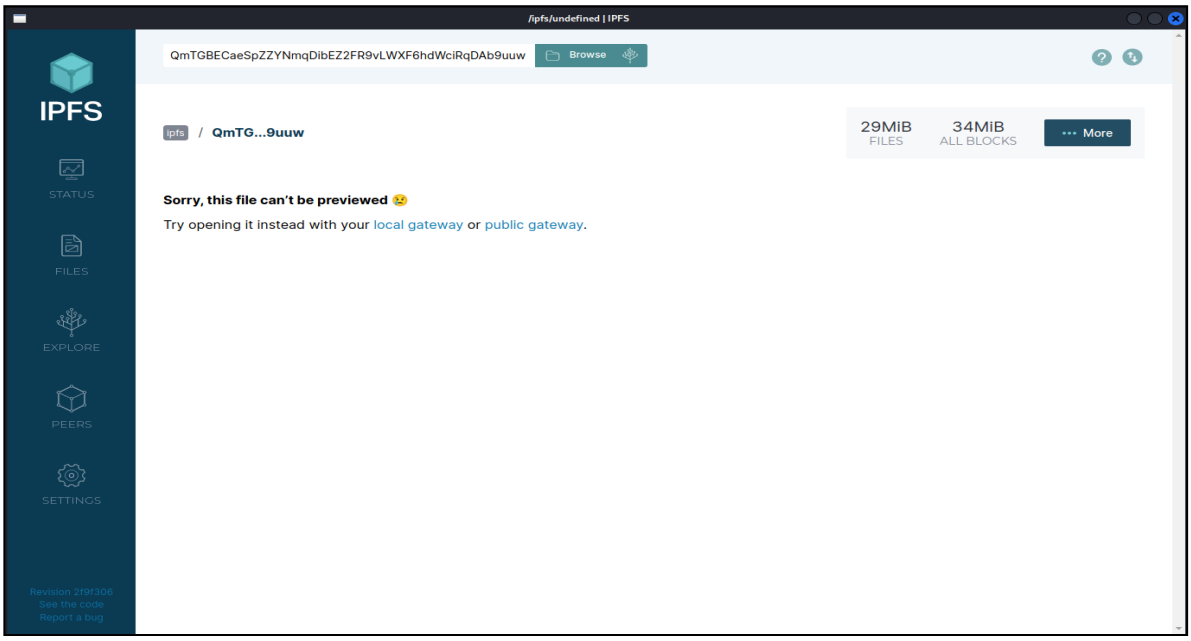


Figure 32:Accès au fichier chiffré sans autorisation.

Pour faciliter l'utilisation de notre application, nous avons utilisé Docker pour créer une image de notre application. Cela permet aux utilisateurs de lancer facilement l'application sur n'importe quel système avec Docker installé, indépendamment du système d'exploitation hôte. Il est à noter que l'installation de Docker est un prérequis pour l'utilisation de notre application. Cependant, Docker est un outil largement utilisé dans l'industrie du logiciel pour sa facilité d'utilisation et sa portabilité, rendant ce prérequis conforme à la pratique standard de l'industrie.

III.6-Test et discussion

Afin de valider notre proposition, nous avons évalué notre solution en comparant notre résultat avec d'autres travaux dans le domaine.

III.6.1-Paramètre d'évaluation

Pour analyser les performances de notre solution de contrôle d'accès basé sur OrBAC et KP-ABE et montrer nos résultats expérimentaux, nous comparons les résultats de notre solution avec un modèle de contrôle d'accès traditionnel en fonction de nombre de fichiers récupérés alors nous proposons d'évaluer la précision et le rappel de notre solution en s'inspirant des formules employées dans le domaine de la recherche d'informations [32] :

1-Le Rappel

Le rappel permet de connaître le pourcentage des fichiers que l'intrus cherchait et auxquels il a réussi à accéder par rapport à l'ensemble des fichiers qu'il cherchait. En d'autres termes, c'est le nombre de fichiers que l'intrus cherchait et a réussi à accéder (Vrai Positif) divisé par l'ensemble de fichiers que l'intrus cherchait (Vrai Positif + Faux Négatif). Sous forme mathématique, on a :

Rappel=nombre de fichiers récupérés / nombre total de fichiers visés pour l'accès.

2-La précision

La précision permet de connaître le pourcentage de fichiers réellement accessibles parmi tous ceux que l'intrus a tenté d'accéder. En d'autres termes, c'est le nombre de fichiers auxquels l'intrus a réussi à accéder (Vrai Positif) divisé par le total de fichiers que l'intrus a tenté d'accéder (Vrai Positif + Faux Positif).Cela nous donne sous forme mathématique :

Précision=nombre de fichiers récupérés/Nombre total de fichiers accessibles.

III.6.2-Résultats expérimentaux et comparaison

Afin d'évaluer la qualité de notre approche, nous proposons de la comparer avec une approche de contrôle d'accès classique qui n'utilise pas KP-OrBAC. Nous proposons l'exemple suivant :

Considérons un système avec un total de cent fichiers. Dans ce système, nous proposons deux scénarios de contrôle d'accès : un système classique et notre approche proposée "KP-OrBAC".

Dans le cas du scénario classique de contrôle d'accès, nous supposons qu'un intrus a obtenu un identifiant d'utilisateur valide, ce qui lui donne un accès illimité aux cent fichiers du système.

D'autre part, dans le scénario KP-OrBAC, nous proposons que l'intrus a également obtenu un identifiant d'utilisateur. Toutefois, en raison de nos mécanismes de contrôle d'accès améliorés, ces justificatifs d'identité ne donnent accès qu'à un sous-ensemble précis de huit fichiers du système (dans le cas de cet identifiant).

Nous examinons également deux cas particuliers dans notre comparaison. Dans le premier cas, l'intrus tente d'accéder à cinq fichiers, et dans le deuxième cas, l'intrus essaie d'accéder à dix fichiers. Il est à noter que dans ces deux cas, les fichiers souhaités font partie du sous-ensemble de huit fichiers auxquels l'utilisateur peut accéder dans le scénario KP-OrBAC.

Maintenant, évaluons le rappel et la précision des deux systèmes dans ces conditions :

fichiers	**Control d'accès classique**		**Control d'accès KP-OrBAC**			
	Identifiant + mot de passe		Identifiant + mot de passe		Identifiant + mot de passe+ (action+organisation+rôle)	
	rappel	précision	rappel	précision	rappel	Précision
5	5/5	5/100	0	0	3/5	3/8
10	10/10	10/100	0	0	3/10	3/8

Tableau 7: Evaluation de KP-OrBAC.

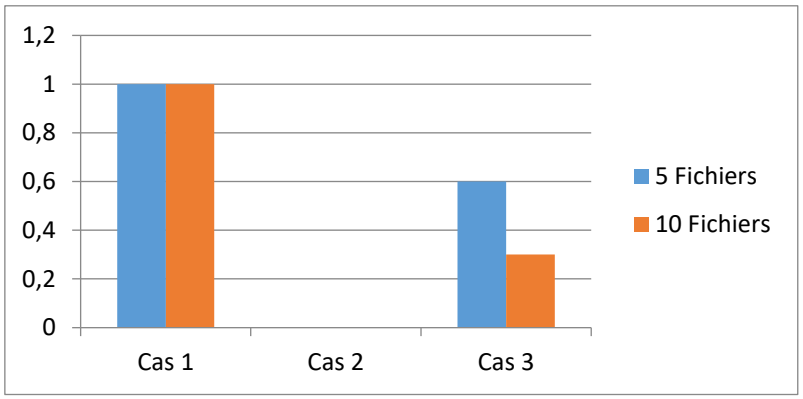


Figure 33:Rappel.

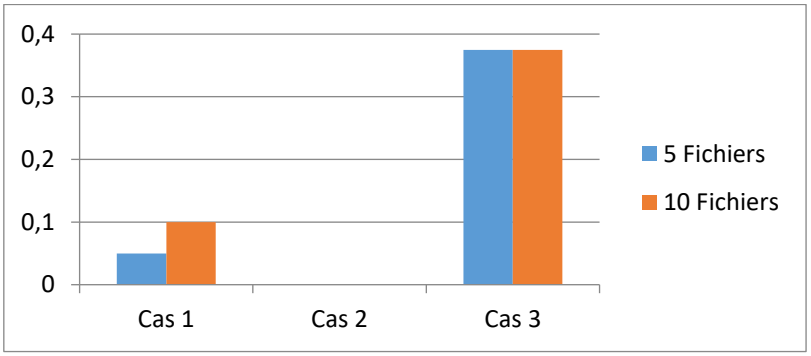


Figure 34:Précision.

Dans le premier cas, où l'authentification se fait avec un simple identifiant et un mot de passe, un intrus qui parvient à passer cette étape peut avoir un accès non restreint à tous les fichiers. Cela présente un risque élevé pour la sécurité des données, car il peut consulter et potentiellement modifier tous les fichiers du système.

Dans le deuxième cas, où l'authentification se fait également avec un identifiant et un mot de passe, mais avec des restrictions basées sur les actions autorisées, la situation diffère. Si l'intrus ne connaît pas les activités auxquelles il a accès, il ne sera pas en mesure de consulter ou de modifier les fichiers. Cependant, si l'intrus parvient à connaître les actions associées à son rôle (rôle d'utilisateur original qui possède l'identifiant), il aura uniquement accès aux fichiers autorisés pour ce rôle (cas 3). Cela offre une meilleure sécurité, car l'intrus ne peut accéder qu'aux fichiers spécifiquement autorisés en fonction de son rôleet des activités associées.

Nous avons évalué aussi notre solution en comparant notre résultat avec d'autres travaux dans le domaine.

Critères	KP-ABE	KP-OrBAC
Type de contrôle d'accès	Basé sur les attributs	Basé sur l'organisation et les attributs
Flexibilité	Peut être utilisé pour une variété d'applications de contrôle d'accès	Peut être utilisé pour des applications de contrôle d'accès plus complexes impliquant des structures organisationnelles
Sécurité	La sécurité dépend de la longueur des clés et de la complexité de la politique d'accès	La sécurité dépend de la longueur des clés et de la complexité de la politique d'accès, mais la structure organisationnelle ajoute une couche de sécurité supplémentaire
Gestion des clés	La gestion des clés est relativement simple car elle est basée sur les attributs	La gestion des clés peut être plus complexe en raison de la structure organisationnelle
Robustesse	Le système KP-ABE est relativement simple par rapport à KP-OrBAC	Le système KP-OrBAC est plus complexe en raison de la structure organisationnelle, mais cela peut offrir des avantages supplémentaires en termes de sécurité et de flexibilité
Applications courantes	Stockage sécurisé de données dans le cloud, messagerie sécurisée	Gestion des droits d'accès pour les entreprises et les organisations, stockage sécurisé de données dans le cloud

Tableau 8: Comparaison entre KP-ABE et KP-OrBAC.

III.7-Conclusion

En conclusion, ce chapitre décrit les détails de la mise en pratique de notre solution de contrôle d'accès. Nous avons décrit comment l'application a été créée en utilisant Django, comment le modèle OrBAC et le système de cryptage KP-ABE ont été utilisés, et comment IPFS a été utilisé pour distribuer l'application aux utilisateurs du cloud.

Nous avons sélectionné et inclus les captures d'écran les plus pertinentes pour illustrer de façon concise et efficace la mise en œuvre de notre système.

Dans la dernière partie de ce chapitre, nous avons testé notre application et l'avons comparée à d'autres modèles existants.

Conclusion générale et perspectives

Conclusion générale

À la fin de notre étude, notre principal objectif est de développer un nouveau modèle qui améliore la sécurité d'accès aux données stockées. Pour atteindre cet objectif, nous proposons un modèle de contrôle d'accès fondé sur le chiffrement basé sur les attributs et les politiques de clés (KP-ABE) et le modèle de contrôle d'accès basé sur l'organisation (OrBAC), spécialement conçu pour les environnements Cloud.

Au cours de la réalisation de ce projet nous avons fait une étude bibliographique que nous avons présentée dans le premier chapitre. En commençant par les concepts essentiels de sécurité de l'information, nous avons étudié les modèles de contrôle d'accès existants, en nous concentrant principalement sur le modèle OrBAC. En outre, nous avons discuté des principes de fonctionnement de KP-ABE. Les deux derniers ont été largement adoptés dans diverses applications à grande échelle et sont au cœur de la solution que nous proposons.

Ensuite nous avons proposé notre solution qui combine les deux approches, d'une façon que les attributs du modèle OrBAC sont utilisés comme des attributs de KP-ABE et seulement les personnes qui satisfait ces attributs peuvent accéder aux données.

Pour la conception et la modélisation de notre système nous avons choisi le langage de modélisation unifié (UML), un langage standard de l'industrie pour spécifier, visualiser, construire et documenter les artefacts des systèmes logiciels.

Dans l'étape de l'implémentation de notre solution proposée, nous avons implémenté ces deux approches dans une application web en utilisant les outils présentés dans le troisième chapitre (Django, Postgresql, Python, etc). Par la suite, nous avons rendu notre application disponible aux utilisateurs du cloud en publiant l'application dans IPFS. Nous espérons que notre travail contribuera à améliorer considérablement les protocoles de sécurité du cloud.

Les perspectives futures de recherche dans le domaine du contrôle d'accès informatique incluent

- Pour renforcer la sécurité de notre approche consiste à utiliser un mécanisme de chiffrement hybride. Ce mécanisme permettrait de chiffrer les clés AES avec un algorithme de chiffrement asymétrique, tel que RSA, offrant ainsi une protection supplémentaire contre le vol de clé. En utilisant cette approche, nous pourrions

profiter à la fois de la rapidité de chiffrement et de déchiffrement offerte par AES, ainsi que de la sécurité renforcée offerte par le chiffrement asymétrique.

- Des mécanismes supplémentaires doivent être mis en place pour assurer la responsabilisation, tels que la journalisation des actions des utilisateurs, la surveillance des activités et la gestion des logs d'audit. Ces éléments permettent de tracer et de vérifier les actions effectuées dans le système.
- Un mécanisme de mise à jour des clés doit être mis en place pour gérer efficacement le processus de mise à jour des clés : lorsqu'un attribut est révoqué, les clés associées à cet attribut doivent être mises à jour pour empêcher l'accès non autorisé.

En conclusion, la fusion d'OrBAC et de KP-ABE dans notre modèle de contrôle d'accès proposé offre un niveau de sécurité sans précédent pour le contrôle d'accès dans le cloud computing, le positionnant comme une solution intéressante pour les entreprises et les industries qui cherchent à protéger leurs données dans le cloud.

Bibliographie

[1] W. Stallings et L. Brown, Computer Security: Principles and Practice, 3rd ed. Pearson, 2015.

[2] J. Vacca, Cloud Computing Security: Foundations and Challenges, 2nd ed, CRC Press, 2020.

[3] R. Dumont, *Cryptographie et Sécurité informatique,* Université de Liège, 2009-2010.

[4] S. Alayda, N. Almowaysher et M. Humayun, *A Novel Hybrid Approach for Access Control in Cloud Computing,* vol. 13, T. I. J. o. E. R. and, Éd., International Research Publication House, 2020.

[5] R. Sandhu, E. J. Coyne et H. L. Feinstein, «Role-based access control models,» vol. 29, n° 12, pp. 38-47, 1996.

[6] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, Y. Deswarte, A. Miege, C. Saurel et G. Trouessin, «Organization Based Access Control,» chez *IEEE 4th International Workshop on Policies for Distributed Systems and Networks (Policy 2003)*, Lake Come, Italy, 4-6 juin 2003.

[7] T. Bellal, «Expression d'une politique de sécurité dans un réseau social,» 2010.

[8] A. J. Menezes, . P. C. van et S. A. Oorschot, Handbook of Applied Cryptography, CRC Press, 1997, pp. 2-11.

[9] C. Sourabh, P. Smita, A. S. Safikul et S. Goutam, «A comparative survey of Symmetric and Asymmetric Key Cryptography,» chez *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, 2014.

[10] A. Sahai et B. Waters, Fuzzy Identity-Based Encryption, 2005, pp. 457-473.

[11] V. Goyal, P. Omkant , A. Sahai et . B. Waters, «Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,» 2006.

[12] H. ZIJIAN, «Attribute-Based Encryption with dynamic attribute feature applied in Vehicular Ad Hoc Networks,» 2022.

[13] J. Bethencourt, A. Sahai et B. Waters, «Ciphertext-Policy Attribute-Based Encryption,» 2007.

[14] C. Langaliya et R. Aluvalu, «Enhancing Cloud Security through Access Control Models: A Survey,» *International Journal of Computer Applications,* vol. 112, p. 8, 2015.

[15] R. Jaichandran , K. Shunmuganathan , V. Subapriya , G. Rahul , S. H. Shahal et R. Rahul , «A Hybrid Encryption Model with Attribute Based Encryption and

Advanced Encryption Standard Techniques,» vol. 12, pp. 334-336, 5 avril 2021.

[16] «Computer Security Resource Center,» [En ligne]. Available: www.csrc.nist.gov. [Accès le 29 avril 2023].

[17] A. RajaniKanth et L. Muddana , *A Survey on Access Control Models in Cloud Computing,* vol. 1, Proceedings of the 49th Annual Convention of the Computer Society of India (CSI): Springer International Publishing, 2015, pp. 653-664.

[18] R. Ostrovsky, A. Sahai et B. Waters, *Attribute based encryption with non-monotonic access structures,* 2007, pp. 195-203.

[19] A. Lewko, O. Tatusuaki , A. Sahai, T. Katsuyuki et B. Waters, «Fully secure functional encryption:Attribute-based encryption and (hierarchical) inner product encryption,» p. 62–91, 2010.

[20] L. Xiaohui , L. Rongxing et L. Xiaodong , «Ciphertext Policy Attribute Based Encryption with Efficient Revocation,» 2010.

[21] y. Zhang, D. Zheng et R. H. Deng, «Security and privacy in smart health:efficient policy-hiding attribute-based access control,» p. 2130–2145, 2018.

[22] P.-S. Chung, M.-S. Hwang et C.-C. lee, «A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments,» july 2013.

[23] R. Buyya, J. Broberg et A. Goscinsk, «Cloud Computing: Principles and Paradigms,» vol. 664, pp. 13-15, Mars 2011.

[24] S. Murugesan et I. Bojanova, Cloud Security. Encyclopedia of Cloud Computing, 2016, p. 205–219.

[25] «A Guide to the Cloud Computing Pyramid: IaaS, PaaS, & SaaS.,» 9 April 2020. [En ligne]. Available: https://sitetoolset.com/blog/2020/04/08/a-guide-to-the-cloud-computing-pyramid-iaas-paas-saas/.

[26] «diagrams-uml-models,» [En ligne]. Available: https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-uml-models. [Accès le 30 mai 2023].

[27] «journaldunet» [En ligne]. Available: https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering. [Accès le 1 aout 2023].

[28] «Introduction à Django. MDN Web Docs» Mozilla, [En ligne]. Available: https://developer.mozilla.org/fr/docs/Learn/Server-side/Django. [Accès le 1 mai 2023].

[29] «definition-postgresql» [En ligne]. Available: https://www.oracle.com/fr/database/definition-postgresql.html. [Accès le 30 mai 2023].

[30] «docs.ipfs.tech/concepts/what-is-ipfs,» [En ligne]. Available:

https://docs.ipfs.tech/concepts/what-is-ipfs/#defining-ipfs. [Accès le 2 juin 2023].

[31] «www.ibm.com,» [En ligne]. Available: https://www.ibm.com/fr-fr. [Accès le 2 juin 2023].