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

Projet de fin d'étude pour l'obtention du diplôme Master

Sécurité des Systèmes d'Information

Conception et réalisation d'une approche de recherche d'information sécurisée sur des données structurées cryptées dans un Cloud

Sujet proposé par :

Mr Boucenna Fateh.

Présenté par :

Merouane Yasmine.

Marakache Fouzia.

Supervisé par :

Mme Zahra Fatma Zohra.

Devant le jury :

Mme Fareh Présidente du jury.

Mme Ouahrani Examinatrice.

- Juillet 2019 -

REMERCIEMENTS

Louanges à celui sans qui rien n'aurait été.

Nous adressons ensuite nos remerciements les plus sincères et les plus chaleureux à notre promotrice Mme ZAHRA pour son encadrement, ses conseils, et son aide.

Nos profonds remerciements vont à Mr BOUCENNA, notre Encadreur pour nous avoir proposé ce thème et aidé inconditionnellement à le réaliser tout au long de notre Stage, ainsi que pour son constant et réconfortant soutien.

Nous tenons à formuler notre reconnaissance aux membres du jury.

Nous tenons à remercier tous les employés de service de sécurité des systèmes d'information de CERIST (Alger) pour l'aide qu'ils nous ont apportée.

Et enfin, merci à tous ceux qui ont contribué de près ou de loin à nous apporter de l'aide.

DEDICACE

Une fleur d’amour et de reconnaissance est dédiée à mes parents.

À mes sœurs : Warda, Wahiba, Ihcene et Soundous.

À mes frères : Youcef, Ayoub, Taher et Houcine.

À mon neveu : Ilyes.

À mes nièces : Maram, Lina et Tesnim.

À toute ma famille.

Et à tous mes amis.

Fouzia.

DEDICACE

Une fleur d’amour et de reconnaissance est dédiée à mes parents.

À mon frère : Rabeh, Cherif, Faize.

À ma nièce : Tesnim.

À mes cousines : Houda, Fatima, Sabiha.

À toute ma famille.

Et à tous mes amis.

Yasmine.

Résumé

Les propriétaires de données sont motivés de plus en plus à externaliser leurs données structurées sur des serveurs Clouds, pour bénéficier de ses multiples avantages. Ces données sont sensibles et elles doivent être protégées contre (les attaques externes / serveur Cloud). Dans ce contexte nous avons proposé et implémenté une solution qui permet d'effectuer une recherche efficace et sécurisée sur des données chiffrées structurées dans le Cloud. Nous avons choisi le chiffrement AES pour chiffrer les tables externalisés. Pour qu'on puisse appliquer l'indexation sur des données structurées on a proposé une méthode d'indexation basée sur le modèle vectoriel. L'index a été chiffré avec la méthode SKNN, et dans le but d'accélérer le processus de recherche un arbre binaire est exploité et démontré par des tests dont les résultats sont satisfaisants.

Mots-clés

Cloud computing, le modèle vectoriel, recherche d'information sur des données chiffrées, le chiffrement SKNN, le chiffrement AES.

Abstract

Data owners are increasingly motivated to outsource their data on Cloud servers, to benefit from its many advantages. This data is sensitive and must be protected against (external attacks / Cloud server). In this context, we have proposed and implemented a solution that allows an efficient and secure search on encrypted data structured in the Cloud. We chose AES encryption to encrypt outsourced tables. In order to apply indexing to structured data, an indexing method based on the vector model has been proposed. The index was encrypted with the SKNN method, and in order to speed up the search process a binary tree is exploited and demonstrated by tests whose results are satisfactory.

Keywords

Cloud computing, the vector model, search for information on encrypted data, AES encryption, SKNN encryption.

Table des matières

Tables des matières

Chapitre 2 : Cloud computing

Tables des matières

Chapitre 03 : Le chiffrement des données (Cryptographie)

Chapitre 4 : La recherche d'information sur des données chiffrées dans le Cloud computing

Tables des matières

Tables des matières

Partie 2 : Conception et Implémentation

Chapitre 5 : Conception de l'approche proposée

Chapitre 6 : Implementation et test

Liste des figures

Liste des tableaux

Glossaire

Terme	Définition
Le chiffrement	D'après Duclos [1] « Consiste à transformer un message en un autre de manière à ne plus reconnaître le premier.»
Le déchiffrement	D'après Duclos [1] « Est l'opération inverse de chiffrement qui permet de récupérer le texte en clair à partir du texte chiffré.»
Système cryptographique	D'après Fattahi [2] « Un système cryptographique est un ensemble d'algorithmes cryptographiques, de textes clairs, de textes chiffrés et de clés de chiffrement et de déchiffrement.»
La stéganographie	D'après Alayrangues et al [3] «Consiste à dissimuler une information dans une autre information. Une telle activité peut, par exemple, consister à cacher des chiffres ou des lettres dans une (image,...etc.).»
Texte en clair	D'après Anstett [4] « C'est l'information à protéger.»
Texte chiffré	D'après Anstett [4] « C'est le résultat du chiffrement du texte en clair. »

Glossaire

Terme	Définition
Clef de chiffrement	La clef [5] est le secret partagé utilisé pour chiffrer le texte en clair en texte chiffré et pour déchiffrer le texte chiffré en texte en clair. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.

Introduction générale

1. Contexte et motivation

L'évolution des systèmes d'information a conduit à la production d'un volume d'information sans précédent .Cependant, la difficulté repose dans la localisation précise de ce que l'on recherche dans cette masse d'information et dans le besoin de grandes capacités de stockage avec un coût réduit.

Avec l'arrivée des systèmes de recherche d'information (SRI) et le Cloud computing, la manipulation de ces informations devient simple et facile. Les services Cloud attirent les entreprises car elles souhaitent délivrer un service de qualité tout en réduisant les coûts (de la maintenance du matériel et des logiciels), et de profiter de la simplicité et des hautes performances offertes par ces solutions. Mais quand une entreprise adopte un service de Cloud computing, elle doit abandonner le contrôle direct de tout système qu'elle déplace vers le Cloud, ce qui engendre naturellement des inquiétudes sur la sécurité.

2. Problématique

Les clients des services de Cloud computing actuellement n'ont aucun moyen de vérifier la confidentialité et l'intégrité de leurs données et de leurs traitements. Pour cela le cryptage vient comme une solution au problème. Mais quand un utilisateur veut effectuer une recherche sur les données, les méthodes classiques de recherche ne sont plus efficaces, d'où la nécessité d'un nouveau système de recherche sur des données chiffrées.

La recherche en pratique nécessite la délégation de l'opération de chiffrement et de déchiffrement des index au Cloud. Cette option ne peut pas garantir la sécurité des données de l'utilisateur, parce que le serveur Cloud est considéré « honnête mais curieux » c'est-à-dire que le serveur Cloud suit honnêtement le protocole de la recherche pour accomplir son rôle, même si les données structurées sont chiffrées, le serveur Cloud peut effectuer des analyses statistiques à partir de la relation entre la requête et le document pour récolter des informations supplémentaires sur ses utilisateurs.

3. Objectifs

L'objectif de ce projet est de concevoir et implémenter une solution qui permet d'effectuer une recherche efficace, accélérée et sécurisée sur des données chiffrées structurées dans le Cloud. Notre solution doit garantir une bonne représentation pour l'index (table/requête) afin

d'empêcher le Cloud de faire des analyses statistiques sur les données structurées sensibles, et de choisir une meilleure technique pour les chiffrer et les utiliser sur le serveur Cloud sans avoir à les déchiffrer.

4. Organisation du mémoire

Notre mémoire est structuré comme suit :

La première partie de notre mémoire est consacrée à la revue de la littérature existante sur le sujet, cette partie comprend 4 chapitres :

- Chapitre 1 :

  Est dédié à la description des généralités de la recherche d'information.

- Chapitre 2 :

  Nous introduisons les concepts de base du Cloud Computing.

- Chapitre 3 :

  Porte sur les généralités de la cryptographie.

- Chapitre 4 :

  Aborde les différentes techniques de la recherche d'information sur les données chiffrées dans la littérature. Il a pour objectifs de définir les concepts essentiels sur lesquels est basé notre travail.

La deuxième partie de notre mémoire est consacrée à l'analyse des besoins, la conception et l'implémentation du système à développer, cette partie comprend les chapitres 5 et 6 :

- Chapitre 5 :

  C'est le chapitre noyau de notre travail où nous ferons une description de l'approche proposée.

- Chapitre 6 :

  Ce chapitre s'intitule la réalisation et l'implémentation, dans lequel nous définirons les outils de développement à utiliser. Nous illustrerons également quelques interfaces de notre application.

Enfin, une conclusion générale qui résume les connaissances acquises durant la réalisation du projet et traite des perspectives futures.

Partie 01

État de l’art

Chapitre 01

Recherche d’information

1. Introduction

La recherche d'information est une discipline de recherche qui intègre des modèles et des techniques dont le but est de faciliter l'accès à l'information et trouver ceux qui correspondent au mieux à l'attente de l'utilisateur [6]**.** Et cette dernière est réalisée par des outils informatiques appelés Systèmes de Recherche d'Information (SRI).

Ce chapitre a pour but de présenter le domaine de la RI. Nous présentons les concepts de base de la RI. En particulier, nous décrivons les notions de document, de requête et de pertinence, les processus d'indexation, ainsi que, les modèles de RI.

2. Définitions

Dans cette partie nous allons définir la RI et le SRI :

2.1. La recherche d'information

La recherche d'information (RI) [7] « est un ensemble des méthodes et techniques pour l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information pertinente pour un utilisateur.»

2.2. Les systèmes de recherche d'information

Un système de recherche d'information (SRI) [8] est un système informatique qui permet de retourner les informations pertinentes pour satisfaire le besoin en information d'un utilisateur, exprimé à l'aide d'une requête dans un temps d'exécution acceptable. Afin d'optimiser ce temps, il est nécessaire d'effectuer certaines opérations sur les documents bruts afin de faciliter leur exploitation.

Indexer un document consiste à extraire un ensemble de mots-clés qui formeront ses descripteurs. Ce processus de recherche, couramment appelé Processus en U de Recherche d'information, est représenté schématiquement sur la figure 1.

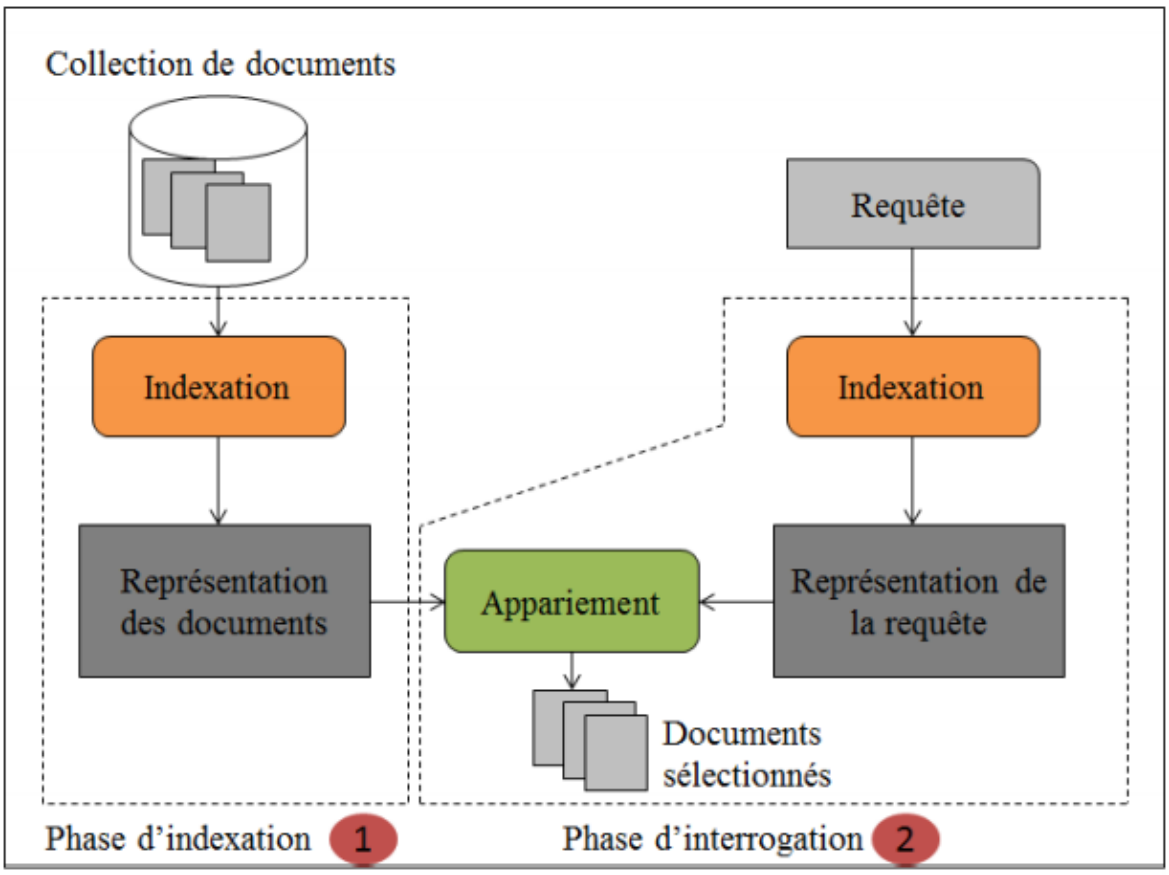

Figure 1 : Processus en U de Recherche d'Information [9].

Ce processus est composé de deux fonctions principales [10] :

A. L'indexation des documents et des requêtes.

B. L'appariement requête-documents qui traite la requête dans le but de sélectionner des documents à présenter à l'utilisateur.

3. Conception de base de la RI

Les concepts de base de la RI sont :

3.1. Document

D'après Zemirli et al [11] « Un document peut être un texte, un morceau de texte, une page web, une image, une vidéo, etc. On peut appeler document toute unité qui peut constituer une réponse à un besoin informationnel de l'utilisateur.»

3.2. Requête

D'après Zemirli et al [11] « Une requête constitue l'expression mentale d'un utilisateur, souvent sous forme d'un ensemble de mots-clés.»

3.3. Collection de documents

La collection de documents (ou fond documentaire) constitue l'ensemble des informations exploitables et accessibles. Elle est constituée d'un ensemble de documents. Dans le cas général et pour un souci d'optimalité, la base constitue des représentations simplifiées mais suffisantes pour ces documents. Ces représentations sont étudiées de telles sortes que la gestion (ajout, suppression d'un document) ou l'interrogation (recherche) de la base se font dans les meilleures conditions de coût [6].

3.4. Besoin en information

D'après Bouramoul et al [6] « C'est l'expression mentale d'un utilisateur. Ce besoin est exprimé par une requête spécifiée dans un formalisme propre au système. Le formalisme de spécification de la requête peut être en langage naturel.»

3.5. Pertinence

La pertinence [12] est une notion fondamentale dans le domaine de la RI .Elle peut être définie comme la correspondance (relation) entre le document et la requête **,** deux types de pertinence ont été distingués dans la littérature : la pertinence système et la pertinence utilisateur **:**

3.5.1. La pertinence système

La pertinence système [12] [13] est définie à travers les modèles de RI. Elle est souvent présentée par un score attribué afin d'évaluer l'adéquation du contenu des documents vis-à-vis de celui de la requête. Ce score est généralement évalué en fonction des poids des mots de la requête dans le document interrogé. Ces poids représentent l'importance des mots pour le contenu d'un document**.** Ce type de pertinence est objectif et déterministe.

3.5.2. Pertinence utilisateur

Quant à elle, est liée à l'évaluation effectuée par l'utilisateur en ce qui concerne l'information renvoyée par le système. Les évaluations de l'utilisateur peuvent être aussi

implicites ou explicite, et elles peuvent être exploitées dans la construction de son profil qui sera ensuite intégré dans le processus de la RI personnalisée [12].

3.6. Correspondance entre la requête et les documents

D'après Hannech [12] « Une fois la requête spécifiée, le système tente de retrouver les documents qui correspondent à la requête en se basant sur une mesure de similarité. »

4. Indexation

L'indexation ce résume en 3 points [14] sont:

1. L'unité d'indexation : Concerne le choix de ce qu'il y a à indexer (représentation des documents ou des requêtes).
2. Extraction des descripteurs (mots-clefs, termes).
3. Représentation des descripteurs par une liste de termes significatifs pour l'unité textuelle correspondante.

4.1. Indexation de documents

Dans la phase d'indexation les documents et les requêtes sont analysés afin d'extraire une liste des mots-clés, appelée « index » ou « descripteur », qui caractérise leur contenu. Généralement, ces mots sont associés à des poids en fonction de leur degré de représentativité par rapport au contenu du document. Peut-être manuelle, automatique ou semi-automatique [15] :

4.1.1. Manuelle

D'après Mitran [13] « Chaque document de la collection est analysé par un spécialiste du domaine ou un documentaliste. Bien que cette indexation produise des termes d'indexation de bonne qualité, mais elle est très difficile à réaliser pour des collections volumineuses et en plus elle est coûteuse. »

4.1.2. Semi-automatique

L'indexation semi-automatique représente la combinaison des deux indexations (manuelle et automatique). Premièrement, un processus automatique extrait les descripteurs de chaque document de la collection. Ensuite, les documentalistes ont la tâche de choisir quels termes

décrivent le mieux les documents. D'habitude, ils se basent sur des vocabulaires contrôlés, tels que les dictionnaires, les thésaurus ou les ontologies [13]**.**

4.1.3. Automatique

C'est la forme la plus répandue d'indexation utilisée par la plupart des SRI. Le processus consiste en la production automatique des descripteurs (termes d'index) d'un texte. Dans le cas des documents textuels, chaque terme est un élément potentiel de l'index du document qui le contient. Il est identifié selon un processus standard intégrant l'extraction, la suppression des mots vides, la normalisation et la pondération [16].

L'indexation automatique [17] regroupe un ensemble de traitements automatisés sur un document comme :

- **L'extraction des mots :** Ce processus consiste à analyser le texte d'un document afin d'extraire ses mots en reconnaissant les espaces de séparation des mots, les ponctuations, etc.

- **L'élimination des mots vides :** Un document contient souvent des mots non significatifs appelés mots vides (pronoms personnels, prépositions). L'élimination de ces mots se fait à l'aide d'une liste prédéfinie de mots vides (appelée stop-List) ou en supprimant les mots ayant une fréquence dépassant un certain seuil. Éliminer les mots vides permet de réduire la taille de l'index, gagner en espace mémoire et optimiser le temps d'exécution.

- **La lemmatisation :** Ce traitement consiste à radicaliser les mots restants, c'est à dire réduire les mots à leur forme canonique. Grâce à la lemmatisation, les documents contenant différentes formes d'un même terme auront les mêmes chances d'être restitués ce qui améliore la capacité d'un SRI à retrouver les documents pertinents. Parmi les méthodes utilisées pour la lemmatisation on peut citer l'algorithme de Porter pour les textes en anglais et la troncature (Mayfield et McNamee) pour les autres langues (Français, Italien, Allemand).

- **La pondération :** Les termes d'un document n'ont pas souvent la même importance. Un terme qui apparaît dans la majorité des documents de la collection aura moins

d'importance qu'un terme qui existe dans quelques documents seulement. Plusieurs fonctions de pondération de termes ont été proposées dans la littérature. La plupart de ces fonctions combinent des variantes des facteurs TF (Term Frequency) et IDF (Inverse Document Frequency) qui mesurent un poids local (dans le document) et global (dans la collection) d'un terme.

- ✓ Tf_{ij} est la fréquence d'occurrences du terme t_j dans le document d_i.
- ✓ Idf_j sa fréquence documentaire inverse. Idfi=log |D | / | {d_j : ti ε d_j} | ou

| D | : nombre total de documents dans le corpus.

| {d_j : ti ε d_j} | : nombre de documents où le terme ti apparaît.

5. Modèles de recherche

Le modèle de recherche d'information [18] c'est le modèle noyau d'un SRI. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de cette mesure de pertinence. Ces modèles ont en commun le vocabulaire d'indexation basé sur le formalisme mots clés et diffèrent principalement par le modèle d'appariement requête-document.

Le vocabulaire d'indexation V = {ti}, i ε {1, ..., n} est constitué de n mots ou racines de mots qui apparaissent dans les documents.

Un modèle de RI est défini par un quadruplet (D, Q, F, R (q, d)) : où

- D est l'ensemble de documents.

- Q est l'ensemble de requêtes.

- F est le schéma du modèle théorique de représentation des documents et des requêtes.

- R (q, d) est la fonction de pertinence du document d à la requête q.

5.1. L'appariement requête-document

Les SRI [6] intègrent un processus de recherche/décision qui permet de sélectionner l'information jugée pertinente pour l'utilisateur. À cet effet, une correspondance entre les termes de la requête d'un utilisateur et ceux des documents s'effectue au niveau de

l'appariement document-requête, cette étape sert à renvoyer une liste de documents ordonnés selon un degré de pertinence, Seuls les documents dont la similitude dépasse un seuil prédéfini sont sélectionnés par le SRI. La fonction de correspondance est un élément clé d'un SRI, car la qualité des résultats dépend de l'aptitude du système à calculer une pertinence des documents la plus proche possible du jugement de pertinence de l'utilisateur.

Il existe deux types d'appariement [12] :

A. Appariement exact

D'après Hannech [12] « Cet appariement peut-être exact tel est le cas avec les modèles booléens dans lequel les documents résultants ont tous la même pertinence.»

B. Appariement approché

D'après Hannech [12] «Les documents résultants peuvent être ordonnés selon le degré de pertinence vis-à-vis la requête. Cette valeur de pertinence est calculée à partir d'une probabilité ou une similarité appelée en anglais « *Retrieval Status Value* » et est notée RSV (q, d), où « q » est une requête et « d » un document. »

5.2. Le Modèle Booléen

Le modèle booléen [19] est le premier modèle de la RI. Il est basé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document.

Exemple 1 : La représentation d'un document est comme suit : d = t1, t2, t3..., tn.

Une requête est représentée par une expression logique quelconque de termes utilisant les opérateurs (OR, AND, NOT).

Exemple 2 : La représentation d'une requête est comme suit :

"q = (t1 ∨ t2) ∧ (t3 ∧ t4) ".

La fonction de correspondance est basée sur l'hypothèse de présence/absence des termes t de la requête q dans le document d_j et vérifie si l'index de chaque document d_j implique l'expression logique de la requête q.

Le résultat de cette fonction est donc binaire, et il est décrit comme suit :

RSV (q, d) = {1,0}.

L'appariement exact est basé sur la présence ou l'absence des termes de la requête dans les documents.

5.2.1. Les avantages du Modèle Booléen

Les principaux avantages du modèle Booléen [19] sont :

1. Le modèle est transparent et simple à comprendre pour l'utilisateur :
   A. Pas de paramètres « cachés ».
   B. Raison de sélection d'un document claire : il répond à une formule logique.
2. Adapté pour les spécialistes et les vocabulaires contraints.

5.2.2. Les inconvénients du Modèle Booléen

Les principaux inconvénients du modèle Booléen [19] sont :

1. La sélection d'un document est basée sur une décision binaire.
2. Pas d'ordre pour les documents sélectionnés.
3. Formulation de la requête difficile pas toujours évidente pour beaucoup d'utilisateurs.
4. Problème de collections volumineuses : le nombre de documents retournés peut être considérable.

5.3. Le modèle Vectoriel (*VSM : Vector Space Model*)

Ce modèle [20][21] préconise la représentation des documents et des requêtes utilisateurs sous forme de vecteurs dans un espace de à t dimensions ou t sont les différents termes d'index présents:

A. L'index d'un document d est le vecteur = (d_1, d_2, d_3,…., d_n), où d_i est le poids d'un terme i dans le document d.
B. La requête est définie par un vecteur q= (q_1, q_2, . . . q_n) où q_i est le poids (souvent 0 ou 1 selon que le terme appartient ou pas â la requête) du terme i dans la requête q.
C. La pertinence du document correspond au degré de similarité entre le vecteur de la requête et celui du document.

Le coefficient de similarité (i.e., RSV) est calculé entre chaque document et chaque requête afin de trouver les documents dont le vecteur de représentation est le plus colinéaire avec le vecteur de la requête. La corrélation de deux vecteurs document-requête, qui représente la mesure de similarité entre les vecteurs de q et de d, peut être calculé par :

Produit scalaire : RSV (q, d) = cos (q, d).

5.3.1. Les avantages du modèle vectoriel (*VSM : Vector Space Model*)

Les principaux avantages du modèle vectoriel résident dans [22]:

1. Sa simplicité.
2. Les performances sont meilleures grâce à la pondération des termes (rapidité).
3. La notion de pertinence un degré d'approximation.
4. La fonction d'appariement permet de trier les documents.

5.3.2. L'inconvénient du modèle vectoriel (*VSM : Vector Space Model*)

« La représentation vectorielle suppose l'indépendance .» [22]

5.4. Le modèle probabiliste (*Probabilistic Model*)

Ce modèle [11][12] aborde le problème de la recherche d'information dans un cadre probabiliste. La pertinence document-requête est traduite par le calcul de la probabilité de pertinence d'un document par rapport à une requête. Le principe de base consiste à retrouver des documents qui ont en même temps une forte probabilité d'être pertinents, et une faible probabilité d'être non pertinents. Le modèle probabiliste évalue la pertinence du document dj pour la requête q. Un document est sélectionné si la probabilité que le document d soit pertinent, notée p(R/D), est supérieure à la probabilité que d soit non pertinent pour q, notée p(R/D) où R soit est l'événement de pertinence ou l'événement de non pertinence.

5.4.1. Les avantages du Modèle probabiliste (*Probabilistic Model*)

Les principaux avantages du modèle probabiliste résident dans [12]:

1. l'utilisateur peut intervenir dans le processus pour améliorer les performances.
2. La fonction d'appariement permet de trier les documents par rapport à la requête.

5.4.2. L'inconvénient du modèle probabiliste (*Probabilistic Model*)

Le modèle considère que tous les termes sont indépendants [12]**.**

6. Conclusion

Le passage vers l'informatique en nuage (*Le Cloud computing*) impacte fortement sur les systèmes d'information vastes et complexes qui contiennent des données sensibles. Pour cela la recherche d'information passe à un haut niveau pour protéger les données qui sont stockées dans cette technologie pour protéger la vie privée de l'utilisateur contre tous les types d'attaques, plusieurs approches ont été proposées pour assurer la sécurité que nous étudierons en détail plus tard.

Chapitre2

Cloud computing

1. Introduction

Le Cloud computing [23] est une révolution économique et technologique, dans lequel les ressources informatiques sont fournies en tant que service via internet. En particulier, ces ressources peuvent être provisionnées de façon dynamique et libérées en fonction de la demande de service et avec un effort minimal de gestion. Il présente une meilleure solution pour gérer les données, les infrastructures,... etc.

Dans ce chapitre nous avons tout d'abord définir Le Cloud computing, ensuite nous verrons les différents services et les modèles du Cloud computing, et nous terminerons par les principaux caractéristiques du Cloud.

2. Définition

L'institut National des normes et de la technologie (*NIST : National Institue of Standards and Technology*) définit l'informatique en nuage comme :

Le Cloud computing [24] est un modèle qui permet un accès réseau pratique et sur demande à un ensemble de ressources informatiques configurables (réseaux, serveurs, stockage, applications et services, par exemple) qui peuvent être rapidement provisionnés et libérés avec un effort de gestion minimal ou une interaction fournisseur de services.

3. Différents services du Cloud computing

L'informatique en nuage est composée de trois services :

1. Le SaaS : Software as a Service.
2. Le PaaS : Platform as a Service.
3. Le IaaS : Infrastructure as a Service.

3.1. IaaS (*Infrastructure as a Service* / Infrastructure fournie en mode service)

L'infrastructure As A Service [24] consiste à louer des ressources d'infrastructure, tels que de l'espace de stockage et d'autres capacités informatiques, sont fournies à titre de service aux utilisateurs. L'accès à la ressource est complet et sans restriction, équivalent à la mise à disposition d'une infrastructure physique réelle.

3.2. PaaS (*Platform as a Service* / Plate-forme fournie en mode service)

Le PaaS [24] permet aux consommateurs de créer des logiciels à l'aide du langage de programmation, des bibliothèques, des services et des outils fournis par le fournisseur. Le consommateur contrôle également le déploiement de logiciels sur l'infrastructure en nuage et paramètres de configuration. Les réseaux, serveurs, stockage et autres services sont fournis en fonction des besoins du client.

3.3. SaaS (*Software as a Service* / Logiciel fourni en mode service)

Le SaaS [25] ou Le logiciel à la demande est un modèle de distribution de logiciels qui fournit accès à des logiciels spécifiques via Internet, tels que Google App, Office 360. Dans SaaS, le fournisseur est connu sous le nom de fournisseur SaaS qui héberge l'application à son centre des données.

L'interface vers l'application se fait généralement par le biais d'un navigateur Web standard. SaaS offre ses avantages à un usage privé ou à une entreprise. Par exemple, au lieu d'acheter la licence d'une application, l'utilisateur final peut obtenir les mêmes fonctions qu'un service hébergé fourni par des serveurs distants. Puisque l'application est gérée à distance et de manière centralisée par fournisseurs, la complexité de l'installation, de la maintenance et de la mise à niveau du logiciel a été diminuée sans intervention de l'utilisateur.

4. Différents modèles du Cloud computing

Le NIST [24] définit plusieurs modèles du Cloud computing qui sont:

4.1. Cloud privé

Est réservé à l'usage exclusif d'une seule organisation. Il peut être possédé, géré et opéré par cette organisation, un intervenant extérieur ou une combinaison des deux. Il est situé dans les locaux de l'organisation ou dans ceux d'un hébergeur externe.

4.2. Cloud public

Est destiné à un usage public. Et peut-être possédé, géré et opéré par un organisme privé, public, académique ou une combinaison de ceux-ci. Il est situé chez un hébergeur.

4.3. Cloud communautaire

Est réservé à l'usage d'une communauté spécifique de consommateurs partageant des intérêts communs : missions, exigences de sécurité, partage d'informations et ou d'applications , Il peut être possédé, géré et opéré par un ou plusieurs organismes participant à la communauté, un intervenant extérieur ou une combinaison d'entre eux. Il est situé dans les locaux de l'organisation ou dans ceux d'un hébergeur externe.

4.4. Cloud hybride

Est composé d'au moins deux modèles différents (privé, public ou communautaire) qui conservent leur autonomie mais qui sont liés entre eux par des technologies (propriétaires ou non) assurant la portabilité des données et des applications.

5. Les caractéristiques essentielles d'un Cloud

Le Cloud computing doit présenter les caractéristiques suivantes :

A. Utilité informatique

Les Clouds sont la nouvelle forme d'utilité informatique, ils permettre une externalisation aisée de l'infrastructure des ressources informatiques, en particulier transférer les applications internes vers un fournisseur Cloud (public) dédié, réduisant ainsi les coûts d'administration et de gestion. Les Clouds améliorent l'utilisation des ressources, principalement par approvisionnement élastique [26].

B. La réduction des coûts

L'utilisateur paie uniquement le service en fonction de son taux d'utilisation (alors qu'il payait par forfait auparavant) [26].

C. L''élasticité

Il s'agit d'une des caractéristiques les plus essentielles du Cloud. Elle définit la capacité d'une infrastructure donnée à s'adapter de manière dynamique au changement [26].

D. La capacité à s'adapter

Le Cloud doit fournir un ensemble d'automatismes lui permettant de s'autogérer. Son administration devra nécessiter le minimum possible d'interventions humaines [26].

E. La qualité de service

A l'aide de métriques telles que le temps de réponse, le nombre d'opérations à la seconde, le service fournit des garanties à ses utilisateurs. Il n'appartient plus à l'utilisateur de devoir décider quelles ressources déployer mais plutôt de définir des bornes que le service doit satisfaire. Le Cloud computing s'adaptera de manière à assurer ses bornes[26].

F. La haute disponibilité

Permet essentiellement l'accès "n'importe où, n'importe quand". La ressource réelle dédiée à l'utilisateur peut ainsi être complètement inconnu et varier constamment en fonction de principe d'élasticité [26].

G. Un retour sur l'investissement

Le paiement à l'utilisation est particulièrement intéressant pour les entreprises de petite taille qui peuvent à présent profiter des avantages d'un service fonctionnel dès le départ. L'idée sous-jacente est donc la suivante : le service deviendra coûteux pour une société dans la mesure où il est fort utilisé, c'est-à-dire à la condition qu'il lui rapporte de l'argent. On passe dés lors de dépenses d'investissement en capital (l'achat de serveurs d'application) aux dépenses d'exploitation (l'achat de ressources consommables) [26].

H. Une démarche écologique

L'allocation de ressources à la stricte nécessité permet de réduire la consommation énergique des parcs informatiques. Outre l'aspect économique, ces réductions énergétiques permettent de diminuer l'empreinte écologique de la société [26].

6. La sécurité des données dans le Cloud

Des menaces de sécurité accrues doivent être surmontées pour profiter pleinement de ce nouveau paradigme informatique. Certains problèmes de sécurité [27] sont énumérés et discutés ci-dessous:

6.1. Le chiffrement dans le cloud

Le client peut utiliser le chiffrement des données pour le stockage dans le cloud, mais il reste à définir qui doit contrôler les clés de chiffrement et de déchiffrement. En toute logique, ceci doit être géré par le client [27].

6.2. Le stockage

Les données peuvent être transférées entre plusieurs datacenter géographiquement éloignés. Le particulier ou l'entreprise ne connait pas la position des données entre chaque datacenter [27].

6.3. L'intégrité des données

Dans le cloud, il est nécessaire d'assurer l'intégrité des données pendant un transfert ou un stockage. Il faut donc que les opérations sur les données soient contrôlées afin de n'effectuer que les opérations qui sont autorisées [27].

7. Conclusion

Le Cloud computing [23] est un nouveau développement technologique qui pourrait avoir un impact considérable sur le monde. Il offre de nombreux avantages aux utilisateurs et aux entreprises. Mais malgré ces avantages, la sécurité des données reste un sujet d'inquiétude pour les entreprises et représente un frein majeur pour l'adoption de cette technologie. Pour cela le cryptage vient comme une solution au problème, pour sécuriser les données des utilisateurs.

Chapitre 03

Le chiffrement des données

(Cryptographie)

1. Introduction

La cryptographie [28] fait aujourd'hui partie de la vie de tous les jours, on l'utilise tous les jours sans même nous en rendre compte: soit en surfant sur Internet ou en envoyant des messages instantanés à nos proches, soit en réglant l'addition avec une carte bancaire,mais qu'est-ce que la cryptographie ? Comment cela fonctionne-t-il ? C'est à ces questions que nous allons essayer de répondre dans ce chapitre. Nous essayerons donc de savoir comment on peut envoyer sans risque (une information, un message, des données ..., etc.) à quelqu'un, sans qu'une personne extérieure puisse le lire. Pour répondre à ces questions, nous allons tout d'abord définir la cryptographie, pour présenter ensuite les différentes techniques existantes. Nous parlerons aussi de la cryptanalyse et nous terminerons par définir les objectifs de la cryptographie.

2. La cryptologie

L'origine du mot cryptologie réside dans la Grèce antique. La cryptologie est un mot composé de deux éléments :

«*cryptos*» qui signifie caché et **« *logos* »** qui signifie mot, qui signifie littéralement la science du secret .la cryptologie est l'art de cacher une information au sein d'un message chiffré [29].

Cette science se divise en deux parties :

2.1. La cryptographie

D'après Jouguet [30] « Comprend des méthodes très variées (des systèmes de chiffrement pour rendre les messages chiffrés), certaines méthodes étant plus efficaces que les autres.»

2.2. La cryptanalyse

D'après Jouguet [30] « La cryptanalyse est la science de l'attaque des systèmes cryptographiques. Son objectif est de briser (casser) un algorithme de chiffrement, c'est-à-dire la récupération de l'information sur (les clairs, la clé).»

3. Domaines de cryptographie

On a deux grands domaines de cryptographie :

3.1. Chiffrement symétrique (Chiffrement à clé secrète)

Dans le chiffrement à clé secrète, encore appelé chiffrement symétrique ou chiffrement conventionnel [31] : les clés de chiffrement et de déchiffrement sont identiques ou facilement déductibles l'une de l'autre, connues uniquement par l'émetteur et le destinataire. La confidentialité du message échangé repose uniquement sur le secret de la clé partagée (voir figure ci-dessous).

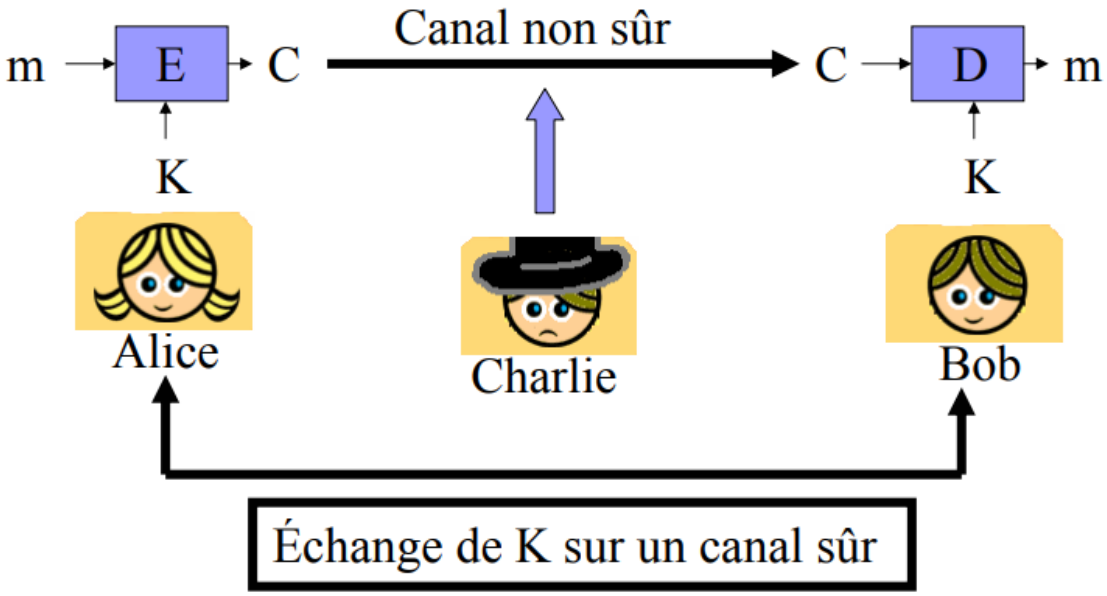

Figure 2: Le fonctionnement du chiffrement symétrique, les deux clés sont identiques[32].

Parmi les chiffrements symétriques on cite AES (*Advanced Encryption Standard*), SKNN (*Secure K Nearest Neighbor*) :

3.1.1. Chiffrement AES

On a deux parties de chiffrement AES :

A. Présentation générale de l'algorithme AES

L'AES (*Advanced Encryption Standard*) [33] est comme son nom l'indique (un standard de cryptage symétrique) .Le système de chiffrement à clé secrète AES est un système basé sur le système Rijdndael construit par Joan Daemen et Vincent Rijmen.

Ce système de chiffrement (AES) a été instigué par le NIST (*National Institue of Standards and Technology*) et il est également approuvé par la NSA (*National Security Agency*) .

C’est un algorithme de chiffrement par blocs :

A. Les blocs de données en entrée et en sortie sont des blocs de 128 bits, c’est à dire de 16 octets.
B. Les clés secrètes ont au choix suivant la version du système : 128 bits (16 octets), 192 bits (24 octets) ou 256 bits (32 octets).

B. Détails techniques

L'AES [33] opère sur des blocs de 128 bits qu'il transforme en blocs cryptés de 128 bits par une séquence de N opérations ou « rounds », à partir d'une clé de 128, 192 ou 256 bits. Suivant la taille de celle-ci, le nombre de rounds diffère : respectivement 10, 12 et 14 rounds. Le schéma suivant décrit succinctement le déroulement du chiffrement :

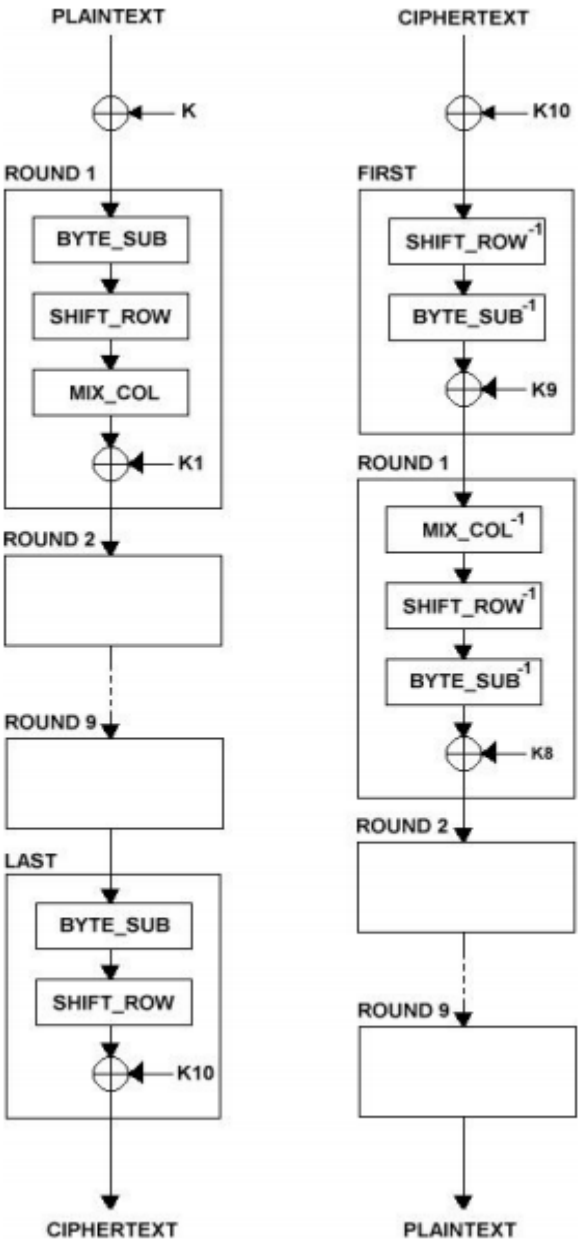

Figure 3 : Le fonctionnement du chiffrement AES [33].

1. Byte_Sub (Byte Substitution)

Est une fonction non linéaire opérant indépendamment sur chaque bloc à partir d'une table dite de substitution (Sbox).

2. Shift_Row

Est une fonction opérant des décalages (typiquement elle prend l'entrée en 4 morceaux de 4 octets et opère des décalages vers la gauche de 0, 1, 2 et 3 octets pour les morceaux 1, 2, 3 et 4 respectivement).

3. Mix_Col

Est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut être exprimée mathématiquement par un produit matriciel.

4. Le signe + entouré d'un cercle : désigne l'opération de OU exclusif (XOR).

K_n : est la n ième sous-clé calculée par un algorithme à partir de la clé principale K.

Remarque

Le déchiffrement consiste à appliquer les opérations inverses, dans l'ordre inverse et avec des sous-clés également dans l'ordre inverse.

3.1.2. La méthode de cryptage SKNN

KNN sécurisé (k-voisin sécurisé), est un chiffrement symétrique utilisé dans le domaine de recherche d'information entre deux vecteurs pour sélectionner les résultats les plus proches.

D'après Cao et al [34], il est nécessaire de chiffrer l'index de la collection (ensemble de concepts vecteurs représentant les documents (Vi') construit par le propriétaire des données) ainsi que les requêtes des utilisateurs avant de les envoyer au serveur Cloud. La clé de chiffrement proposée (S, M1, M2) :

1) S est un vecteur de taille (m + U +1). (S : c'est un vecteur booléen, m étant le nombre total de concepts (Vi'), U : nombres des mots fictifs).

2) M1 et M2 sont deux matrices de taille (m + U + 1) avec m étant le nombre total de concepts (Vi') et U : nombres des mots fictifs.

Le processus de cryptage se fait en trois (3) étapes (extension, division et multiplication) comme suit :

A. Extension

1) Pour les documents :

1. On ajoute des dimensions U + 1 à chaque vecteur de document Di de taille m.
2. La valeur 1 est affectée à la (m + 1) ième dimension. Considérant que, une valeur aléatoire est affectée à la (m + j + 1) ième dimension (Où j ∈ [1, U]). Les dernières dimensions correspondent à des mots fictifs.

$$\overrightarrow{D_i} = \{ D_i \ , 1, \varepsilon_i^1, \varepsilon_i^2, \varepsilon_i^3, \ldots, \varepsilon_i^u \}$$

2) Pour la requête :

1. Le vecteur de requête (qui est également de taille m) est multiplié par un paramètre aléatoire r.
2. Une dimension avec une valeur aléatoire t est ajoutée au vecteur obtenu à la (m + 1) ième dimension.
3. Les dimensions U sont ajoutées à ce vecteur.
4. Une valeur α est affectée à la (m+ j +1) ième dimension (avec $\alpha^j \in \{0,1\}$).

$\overrightarrow{Q} = \{r.\ Q, t, \alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^U\} / \alpha^j \in \{0,1\}$

B. Division

1. Chaque vecteur de documents $\overrightarrow{D_i}$ est divisé en deux vecteurs $\{\overrightarrow{D'_i}, \overrightarrow{D''_i}\}$,
2. Chaque vecteur de requêtes $\overrightarrow{Q}$ est divisé en deux vecteurs $\{\overrightarrow{Q'}, \overrightarrow{Q''}\}$.
3. Le vecteur S est utilisé comme indicateur de fractionnement.

En effet, **si le j ième élément de S est égal à 0 :**

1. alors $\overrightarrow{D'_i}[j]$ et $\overrightarrow{D''_i}[j]$ Aura la même valeur que $\overrightarrow{D_i}[j]$.
2. chacun des deux éléments $\overrightarrow{Q'}[j]$ et $\overrightarrow{Q''}[j]$ Aura une valeur aléatoire telle que leur somme soit égale à $\overrightarrow{Q}[j]$.

Si le j ième élément de S est égal à 1 :

Nous suivons le même principe, sauf que le vecteur de document et le vecteur de requête sont commutés :

1. $\overrightarrow{Q'}[j]$ et $\overrightarrow{Q''}[j]$ Aura la même valeur que $\overrightarrow{Q}[j]$.
2. Chacun des deux éléments $\overrightarrow{D'_i}[j]$ et $\overrightarrow{D''_i}[j]$ Aura une valeur aléatoire telle que leur somme soit égale à $\overrightarrow{D_i}[j]$.

C. Multiplication

1. Enfin, les matrices M1 et M2 sont utilisées pour finaliser le chiffrement de chaque vecteur de document de la manière suivante : $I_i = \{\overrightarrow{D'_{i..}} M_1^T, \overrightarrow{D''i}.M_2^T\}$
2. Pour crypter les deux vecteurs de la requête on applique :
   $T_q = \{M_1^{-1}.\overrightarrow{Q'}, M_2^{-1}.\overrightarrow{Q''}\}$
3. En appliquant le produit scalaire entre un vecteur de document et un vecteur de requête, on obtient :
   $Ii.\ Tq = \{\overrightarrow{D'_{i..}} M_1^T, \overrightarrow{D''i}.M_2^T\} \times \{M_1^{-1}.\overrightarrow{Q'}, M_2^{-1}.\overrightarrow{Q''}\}$
   $= \overrightarrow{D'_i} \times \overrightarrow{Q'} + \overrightarrow{D''_i} \times \overrightarrow{Q''}$
   $= \{ D_i ,1, \varepsilon_i^1, \varepsilon_i^2, \varepsilon_i^3, \ldots\ldots\ldots\ldots \varepsilon_i^u\} \times \{r.\ Q, t, \alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^U\}$
   $= r.\ D_i .Q + \sum_{j=1}^{u} \varepsilon_i^j . \alpha^j + t$

Les paramètres aléatoires $\{\varepsilon_i^j, \alpha^j, t, r\}$ sont utilisés pour masquer le score de similarité réel entre un document et une requête. Cependant, les scores de similarité alternatifs sont utiles pour trier les documents par pertinence.

Il est nécessaire que les paramètres $\varepsilon_i^j, \alpha^j, t$ seront comme un couple de valeurs. À savoir :

$\varepsilon_i^j = (\varepsilon'^j_i, \varepsilon''^j_i)$,

$\alpha j = (\alpha'_j, \alpha''_j)$ où $\alpha'_j = \alpha''_j$.

$t = (t', t'')$

Le paramètre r est toujours comme une valeur unique.

3.2. Chiffrement asymétrique (Chiffrement à clé publique)

Le chiffrement asymétrique [31] diffère radicalement du précédent. Les clés de chiffrement et de déchiffrement sont différentes. La clé de chiffrement ainsi que l'algorithme sont connus de tous. La sécurité du système repose sur le secret de la clé de déchiffrement et sur l'impossibilité (au moins en pratique) de déduire la clé de déchiffrement (dite clé privée) de la connaissance de la clé de chiffrement (dite clé publique). Chaque protagoniste possède une paire de clés, la clé publique étant publiée et la clé privée n'étant connue que de son propriétaire (voir figure ci-dessous).

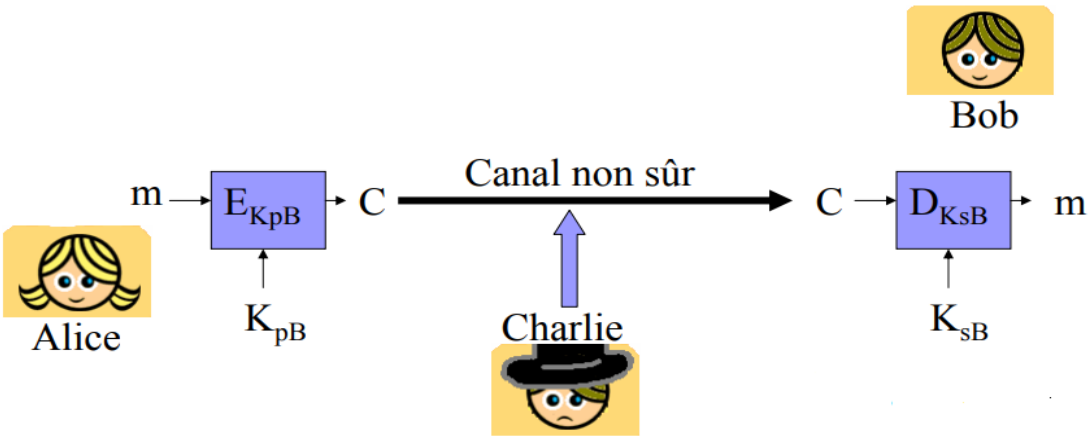

Figure 4 : Le fonctionnement du chiffrement asymétrique [32].

3.2.1. Chiffrement RSA

Afin de mieux comprendre l'algorithme, On le décompose en deux parties :

A. Présentation générale de l'algorithme RSA

Cette méthode [35] a été inventée en 1978 par trois mathématiciens, Rivet, Shamir et Adleman. Ce qui fait son originalité c'est que la clé de chiffrement est connue de tous (clé public), et cependant une seule personne peut déchiffrer le message avec son clé privée, c'est-à-dire qui ne puisse pas être décrypté par un tiers qui intercepterait le message.

RSA est considéré comme un système de sécurité absolue dès qu'on utilise de grands nombres premiers. Autrement dit vous ne pouvez pas déterminer p et q.

B. Détails techniques

Dans cette méthode [35] on à 2 entités l'émetteur et le récepteur, Nous appellerons Alice l'émittrice du message, et Bob le déstinataire.

1. Bob choisi deux nombres premiers p et q.

2. Bob calcule le produit n = p.q.

3. Bob choisi un entier premier e avec Q(n) tel que $1< e < Q(n)$ et $Q(n)= (p – 1) (q – 1)$.

4. Bob calcule la clé secrète d tel que $1 < d < Q(n)$ et $e.d \equiv 1 [Q(n)]$.

5. Bob rend l'information (RSA, n, e) publique dans un annuaire.

6. Alice crypte un message M par $M \rightarrow M^e[n]$ et il envoie le résultat à Bob tel que (e, n de Bob).

7. Enfin on Suppose que Bob a reçu le message chiffré C de la part de Alice, en utilisant sa clé secrète décode alors le message crypté par $C \rightarrow C^d[n]$ (d, n de Bob).

4. Objectifs de la cryptographie

Les principales fonctionnalités offertes par la cryptographie sont :

4.1. Le chiffrement

D'après Videau [31] « Il est utilisé pour protéger le contenu d'un message contre sa divulgation à toute personne ne possédant pas le secret lui permettant de le lire en clair. »

4.2. L'identification

D'après Videau [31] « Elle permet d'assurer l'identité de la personne dont est issu le message reçu. »

4.3. L'authentification

D'après Videau [31] « Assure que les données reçues n'ont subi aucune modification depuis leur émission première. »

4.4. La signature

D’après Videau [31] « Comme la signature manuscrite protège la provenance, l'intégrité et garantit la non-répudiation d'un message. »

5. Conclusion

La cryptographie a défini les notions de sécurité et prouvé la sécurité des crypto-systèmes de chiffrement, des codes d’authentification de messages et des signatures numériques. De plus, des protocoles de plus haut niveau, comme des systèmes de communications sécurisées ou de votes électroniques, ont été conçus**.**

Aujourd’hui, la cryptographie tente de concevoir des systèmes plus sûrs et plus efficaces. Mais les crypto-systèmes sont encore en danger à cause des attaques qui sont dû à une mauvaise implémentation des mécanismes cryptographiques.

Chapitre 4

La recherche d’information sur des données chiffrées dans le Cloud

1. Introduction

En raison de la popularité croissante du Cloud computing [36], de plus en plus de propriétaires de données sont motivés à externaliser leurs données sur des serveurs Cloud pour une grande commodité et réduction des coûts de gestion des données.

Le fait que les propriétaires de données et le serveur Cloud ne sont plus dans le même domaine de confiance , a conduit à des préoccupations croissantes de leurs données, dans le but de garantir la sécurité des données et pour des besoins de confidentialité, les données sensibles doivent être cryptées avant l'externalisation ,ce qui rend la tâche de récupération des données très difficile, la difficulté qui se pose est dans la recherche sur les données qui se trouve dans le Cloud sans que le moteur de recherche ne sache les mots clés de la recherche.

2. L'architecture générale d'un système de recherche d'information crypté

On va résumer les points principaux de système comme suit :

2.1. Architecture de système

La figure suivante représente l'architecture générale de système :

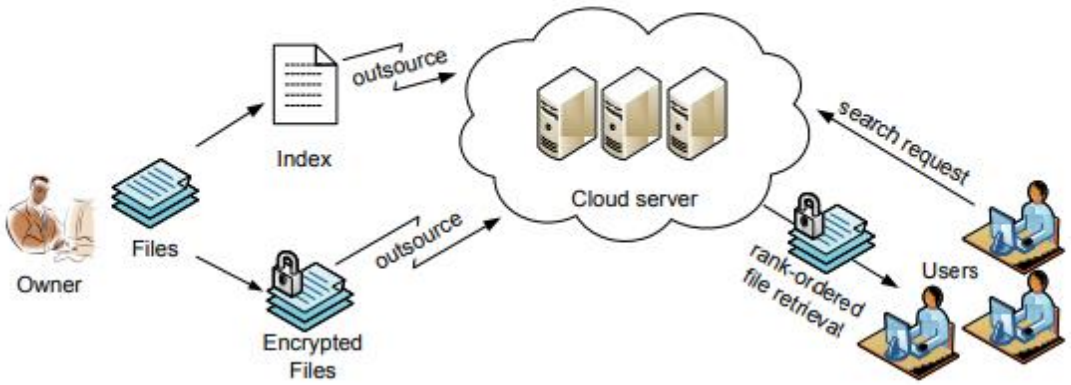

Figure 5 : Architecture de système [37].

2.2. Principe

Un système de recherche d'information crypté est composé de 3 entités principales [37] sont:

A. Propriétaire des données (*Data owner)*

Ce dernier possède N documents et il veut les externaliser, le processus de l'externalisation passe comme suit :

1. Il calcule l'index de ces documents.
2. Il chiffre l'index et le document séparément.
3. Il envoie l'index et le document chiffré au serveur Cloud.

B. Utilisateur des données

Le processus de la recherche passe comme suit :

1. Un utilisateur autorisé génère une requête (chiffrée) au serveur Cloud.
2. Le serveur Cloud exécute la recherche sur l'index.
3. Le Cloud retourne la collection correspondante des documents cryptés les mieux classés.
4. Les fichiers seront déchiffrés avec un algorithme de décryptage.

C. Serveur Cloud

1. Stocke la collection de documents cryptée C et l'index chiffré I pour les données du propriétaire.
2. Le Cloud retourne la collection correspondante des documents cryptés les mieux classés aux utilisateurs.

3. Les approches de recherches sur une donnée cryptée dans la littérature

Plusieurs approches ont été proposées ces dernières années, nous étudierons certaines qui sont les plus connues :

3.1. Méthode traditionnelle de la recherche sur des données cryptées

Selon Yang et al [38] c'est la solution la plus simple, son principe : est de télécharger toutes les données et de les décrypter localement puis rechercher le terme dans le texte en clair. Cette solution n'est pas pratique, en raison de l'énorme quantité de données et le coût de la bande passante dans les cloud. Cette dernière est basée sur le cryptage symétrique.

3.2. Une recherche sécurisée et dynamique avec plusieurs mots clés sur des données chiffrées en nuage (*A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data*)

On va résumer les points principaux de l'approche comme suit :

3.2.1. Architecture de système

La figure suivante représente l'architecture générale de système :

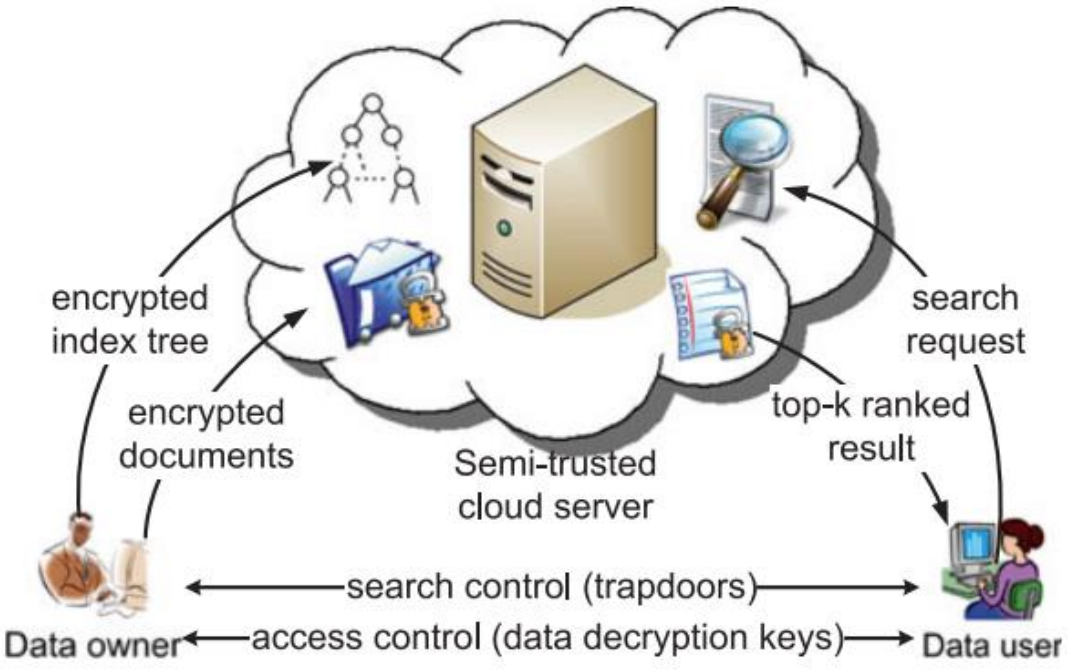

Figure 6 : Architecture de système [39].

3.2.2. Principe

Wang et al [39] proposent un schéma de recherche sécurisé basé sur une arborescence utilisant les données chiffrées du Cloud, qui prend en charge la recherche avec plusieurs mots clés et le fonctionnement dynamique de la collection de documents (des opérations de mise à jour dynamiques telles que la suppression et l'insertion de documents en particulier).Le modèle d'espace vectoriel et le modèle largement utilisé tf-idf sont combinés dans la construction d'index et la génération des requêtes pour fournir une recherche avec plusieurs mots clés. En outre, le processus de recherche parallèle peut être effectué pour réduire encore le coût du temps.

3.2.3. Le fonctionnement de l'approche

Le modèle de système proposé par Wang et al [39] implique trois droits différents :

A. Propriétaire des données

1. Crée une collection de documents F = {f1 ; f2... Fn}.
2. Construit un index I de la collection des documents F sous forme d'un arbre (en clair).
3. Génère une collection de documents cryptés C pour F.
4. Génère l'index sécurisé I avec l'algorithme SKNN.
5. Envoie la collection cryptée C et l'index sécurisé I au serveur de nuage.
6. En toute sécurité distribue les informations clés de la génération de trappes (y compris les mots clés IDF)
7. En plus, il est responsable de la mise à jour des informations localement et l'envoie au serveur.

B. Utilisateur des données

Les utilisateurs de données sont les utilisateurs autorisés à accéder aux documents du propriétaire des données.

1. Avec t mots-clés de requête, l'utilisateur autorisé peut générer une trappe TD selon le mécanisme de contrôle de recherche pour récupérer k documents cryptés du serveur de nuage.

2. Ensuite, l'utilisateur de données peut déchiffrer les documents avec la clé secrète partagée.

C. Serveur Cloud

1. Stocke la collection de documents cryptée C.
2. Stocke l'index arbre chiffré I.
3. Reçoit la trappe TD des utilisateurs de données.
4. Le serveur Cloud exécute la recherche sur l'index arbre chiffré I, et retourne enfin la collection correspondante des documents cryptés les mieux classés.
5. Reçoit les informations de mise à jour du propriétaire des données, et il doit faire la mise à jour de l'index I et le document chiffré C en fonction des informations reçues.

3.2.4. Les avantages de l'approche

Les principaux avantages du ce système [39] sont :

A. En raison de la structure particulière de l'index arborescent, le schéma de recherche proposé permet de réaliser de manière flexible un temps de recherche réduit et de traiter la suppression et l'insertion de documents.

B. Confidentialité de l'index, requête, les valeurs TF des mots-clés stockés dans l'index et les valeurs IDF de la requête et les informations en texte brut.

C. Dissociabilité des trappes : Le serveur Cloud ne doit pas être capable de déterminer si deux requêtes cryptées (trappes) sont générés à partir de la même demande de recherche.

D. Confidentialité des mots clés : Le serveur de nuage n'a pas pu identifier le mot clé spécifique dans la requête, l'index ou le document collecte en analysant les informations statistiques.

3.3. Recherche d'images cryptées sécurisée et efficace avec contrôle d'accès (*SEISA : Secure and Efficient Encrypted Image Search With Access Control*)

On va résumer les points principaux de l'approche comme suit :

3.3.1. Architecture de système

La figure suivante représente l'architecture générale de système :

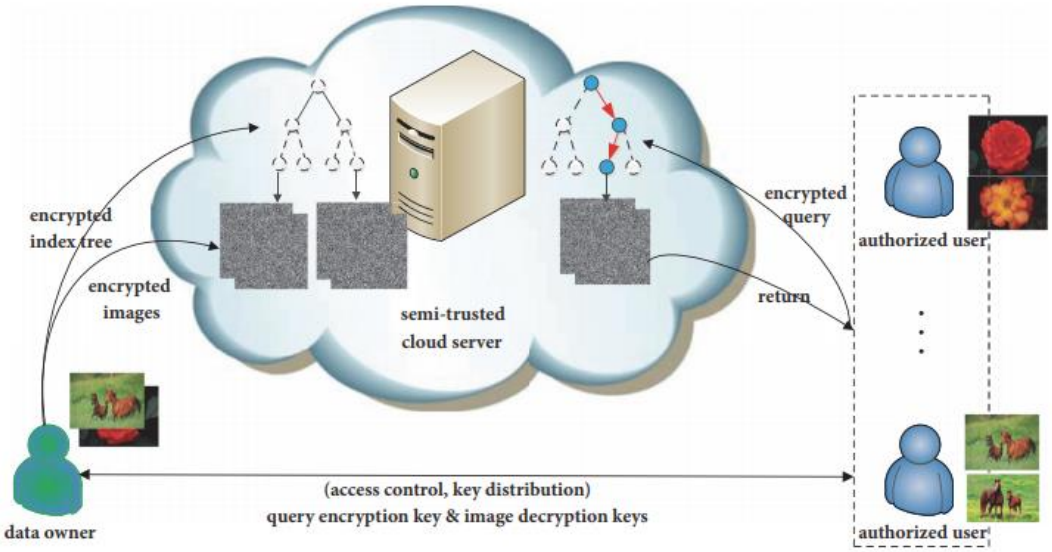

Figure 7 : Architecture de système [40].

3.3.2. Principe

SEISA [41] est un système de recherche d'images sécurisé et léger sur les données cryptées prend en charge efficacement le contrôle d'accès à la recherche, qui permet aux propriétaires de données de définir qui peut rechercher une image spécifique.

3.3.3. Le fonctionnement de l'approche

Ce système [41] comprend trois parties **:**

A. Le propriétaire des données

1. Sous-traite un grand nombre d'images dans le format crypté.
2. Pour améliorer l'efficacité de la recherche, il crée un arbre d'index basé sur les fonctionnalités intégrées. Ensuite, il utilise une autre clé secrète pour chiffrer l'arborescence de l'index.

3. Il permet de définir des règles d'accès à la recherche pour chaque image.
4. Le propriétaire envoie des clés secrètes à chaque utilisateur autorisé via un canal sécurisé.

B. L'utilisateur des données

1. Il a la capacité de s'authentifier mutuellement avec le propriétaire des données.
2. Il peut effectuer une recherche dans le jeu de données d'image en soumettant une recherche cryptée demande au serveur Cloud.
3. Avec les textes chiffrés des images retournées par le nuage, les utilisateurs des données peuvent utiliser leurs propres clés pour récupérer les images.
4. Avec des stratégies définies, les utilisateurs des données ne peuvent rechercher que des images. Qu'il / elle a des rôles d'utilisateur appropriés.

C. Le serveur Cloud

1. Il stocke les images cryptées de la base de données et l'arborescence d'index sécurisée du propriétaire.
2. Après avoir reçu une requête chiffrée, le Cloud exécute l'opération de recherche dans l'arborescence d'index sécurisé.
3. Récupère des textes chiffrés d'images pertinentes pour les utilisateurs sans déchiffrer les données.

3.3.4. Les avantages de l'approche

Les principaux avantages de ce système [41] sont :

A. Protège simultanément la confidentialité des propriétaires de données et des utilisateurs en empêchant le serveur Cloud de connaitre les informations suivantes :

1. Confidentialité de toutes les images et de leurs vecteurs descripteurs correspondants.
2. La confidentialité de toutes les images de recherche issues des demandes de recherche.
3. Confidentialité de toutes les données associées à la construction de l'index.
4. Impossible de lier les requêtes, les attaquants ne devraient pas être en mesure de dire si deux ou plusieurs recherches sont issues de la même image de recherche.

B. Prend en charge efficacement le contrôle d'accès à la recherche.

3.4. Recherche par mot-clé flou vérifiable et efficace sur des données chiffrées dans le Cloud Computing (*Efficient Verifiable Fuzzy Keyword Search over Encrypted Data in Cloud Computing*)

On va résumer les points principaux de l'approche comme suit :

3.4.1. Architecture de système

La figure suivante représente l'architecture générale de système :

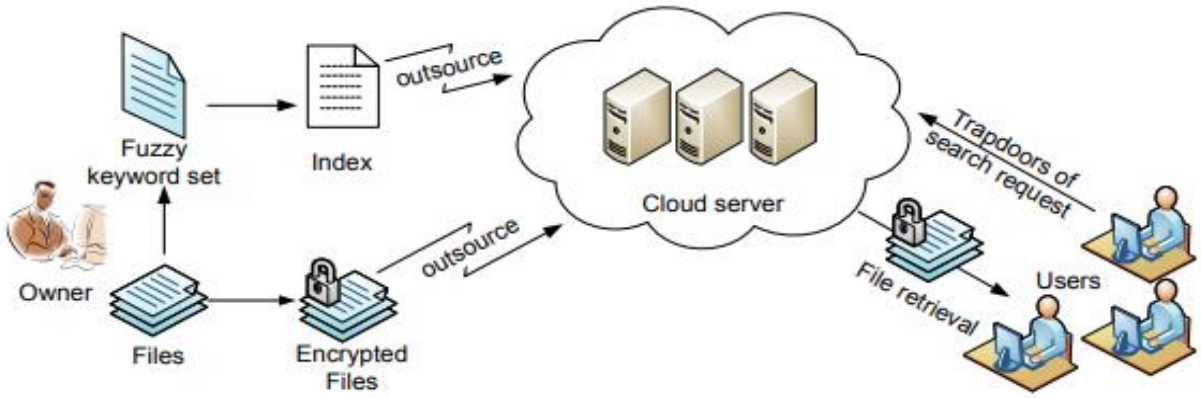

Figure 8 : Architecture de système [42].

3.4.2. Principe

Wang et al [43] proposent une approche de recherche multi-termes floue qui élimine l'exigence d'un dictionnaire de mots-clés prédéfini. Le flou du mot clé est capturé par une structure de données innovante et conception, et présente donc une grande efficacité en termes de calcul et espace de rangement grâce aux techniques de hachage sensible à la localité (LSH) et les filtres de Bloom.

3.4.3. Les objectifs de conception

La conception proposée par Wang et al [43] porte la sécurité et les performances suivantes:

A. Recherche floue avec plusieurs mots clés

L'objectif principal est de prendre en charge la recherche floue contenant plusieurs mots clés. Par exemple, Fichiers liés à la «sécurité du réseau» doivent être trouvés pour requête mal orthographiée «sécurité de netword».

B. Garantie de confidentialité

Ce système devrait garantir la confidentialité en évitant de divulguer des informations sur les données fichiers ou les mots-clés de requête au-delà des résultats de la recherche pour le serveur Cloud.

C. Précision du résultat

Puisqu'il s'agit d'une recherche floue, la précision du résultat est une mesure de performance importante. Ce schéma devrait trouver les résultats aussi précis que possible et garder la précision dans une plage acceptable.

D. Aucun dictionnaire prédéfini

La nécessité d'un dictionnaire prédéfini, le dictionnaire est un facteur limitant qui rend les opérations des données dynamiques, telles que la mise à jour de l'ensemble de données / index, sont très difficiles. Cette conception élimine cette exigence contrairement à de nombreuses solutions précédentes.

3.4.4. Les techniques utilisées

Deux techniques importantes [43] sont utilisées dans cette conception Bloom filtre et hachage sensible à la localité (LSH) :

1. Filtre Bloom

Un filtre Bloom est un tableau s de m bits, tous qui sont mis à 0 initialement. Soit un ensemble S = {a1, a2, …, an}, un filtre Bloom utilise l fonctions de hachage indépendantes h.

A. Pour insérer un élément a ∈ S :

Il faut mettre tous les bits de la position hi (a) à 1.

B. Pour tester si un élément q est dans S :

Envoyez-le à chacune des l fonctions de hachage pour obtenir les l positions de tableau. Si le bit à une position quelconque est 0, alors q n'est pas inclut dans S; sinon le q appartient à S ou q donne un faux positif.

2. Hachage sensible à la localisation

Il utilise la Distance euclidienne et la représentation vectorielle pour définir la similarité entre les termes, la fonction LSH hache les éléments proches à la même valeur de hachage avec une probabilité plus élevée que les éléments qui sont éloignés pour fournir une recherche floue et efficace.

3.5. EPCBIR : Schéma de récupération d'images basé sur le contenu, efficace et préservant la confidentialité, dans le cloud computing (*An efficient and privacy-preserving content-based image retrieval scheme in cloud computing*)

On va résumer les points principaux de l'approche comme suit :

3.5.1. Architecture du système

La figure suivante représente l'architecture générale de système :

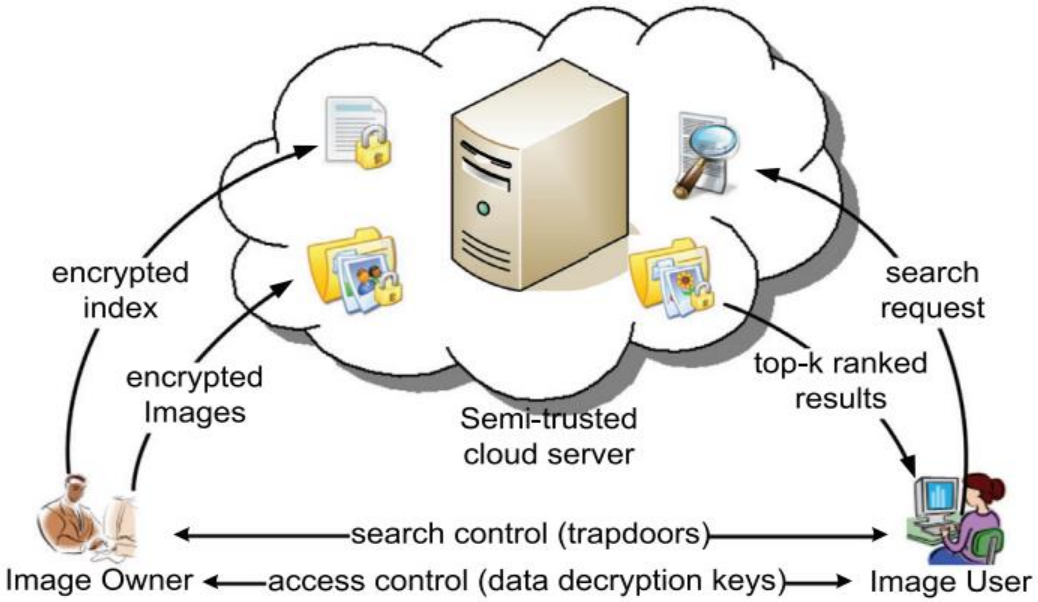

Figure 9 : Architecture de système [44].

3.5.2. Principe

Ce schéma [44] a été construit par Lu et al pour CBIR (la récupération des images basée sur le contenu préservant la confidentialité dans le Cloud computing). Le service CBIR est généralement très complexe en termes de stockage et de calcul donc l'informatique offre une grande opportunité pour l'accès à la demande aux vastes ressources de calcul et de stockage, qui est un choix attrayant pour le stockage d'images et l'externalisation CBIR.

3.5.3. Le fonctionnement de l'approche

Le système [44] [45] comprend trois parties **:** Le propriétaire de l'image, l'utilisateur de l'image et le serveur Cloud.

A. Le propriétaire de l'image

Le propriétaire de l'image veut externaliser ses données locales, c'est-à-dire une collection de n images M = {m1, m2, . . ., mn}, vers le serveur Cloud sous forme cryptée C = {c1, c2, . . ., cn}, tout en gardant la possibilité de rechercher sur les images cryptées.
Tout d'abord, le propriétaire de l'image extrait les vecteurs caractéristiques F = {f1, f2, . . ., fn} à partir de M, puis il construit un index de recherche sécurisé I sur F. Ensuite, la collection d'images cryptées C et l'index I sont externalisés vers le serveur de Cloud.
Le propriétaire de l'image prend également la responsabilité d'autoriser les utilisateurs d'image.

B. Les utilisateurs d'image

Sont ceux qui sont autorisés à récupérer des images du serveur Cloud. Pour demander une recherche, l'utilisateur de l'image commence par générer une trappe d'accès TD pour l'image de la requête, puis la soumettre au serveur Cloud. Après avoir reçu les images résultantes, l'utilisateur peut les décrypter avec la clé secrète partagée par le propriétaire de l'image.

C. Le serveur Cloud

Il stocke la collection d'images cryptées C et l'index I pour le propriétaire de l'image et il traite les requêtes des utilisateurs.

3.5.4. Les principales contributions

Les principales contributions [44] sont résumées comme suit :

1. L'algorithme KNN sécurisé est utilisé pour protéger les vecteurs de caractéristiques, ce qui permet au serveur Cloud de classer la recherche.

2. Un index de deux couches est construit :

   A. La partie supérieure est un ensemble de tables de pré-filtrage, ce qui contribue à augmenter l'efficacité de la recherche.
   B. Le plus bas est l'index one-one map qui peut être utilisé pour classer les résultats de la recherche.

3.5.5. Les avantages de l'approche

Les principaux avantages du ce système [44] [46] [47] sont :

A. Les images sont cryptées par les cartes chaotiques .Les cartes chaotiques était synonyme de désordre et de confusion peuvent bien protéger l'histogramme et les statistiques de corrélation.
B. Les vecteurs de caractéristiques sont cryptés par l'algorithme KNN sécurisé, qui s'avère être sûr contre le modèle Ciphertext-only Attack (COA).
C. Les tables de pré-filtrage sont en outre protégées par une fonction de hachage à sens unique.

3.6. Comparaison entre les techniques d'accélération

Tableau 1 : Comparaison entre les techniques d'accélération**.**

Approches / Critères	**A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data** [39]	**Efficient Verifiable Fuzzy Keyword Search over Encrypted Data in Cloud Computing** [43]	**An efficient and privacy-preserving content-based image retrieval scheme in cloud computing** [44] [45]
Qualité des résultats de recherche retournés	Retourne des résultats exacts	Donne des faux positifs (filtre bloom)	Retourne des résultats approximatifs
Complexité de la recherche	O(n) dans les pires des cas (arbres dégénérés) et O (log n) dans les meilleurs des cas (arbres bien équilibrés) où n est la hauteur de l'arbre.	O(k) où k est toujours le nombre de fonctions de hachage du filtre.	O (n'). tel que : n'< n où n est le nombre total des images. et (n') est le nombre des fonctions de hachages des images similaires dans un groupe.

Dans le tableau ci-dessus on a fait une comparaison entre les techniques d'accélération utilisées dans chaque approche. Pour comparer les techniques d'accélération on a pris deux critères en considération (Complexité de la recherche et la qualité des résultats de recherches retournés). La complexité de recherche est minimale dans les fonctions de hachage par rapport aux arbres binaires de recherche, mais la qualité et la certitude des résultats de recherches retournés est un critère très important lorsqu'on travaille sur des bases de données. Pour cella on a choisi les arbres binaires de recherche pour accélérer la recherche.

4. La recherche d'information sécurisée sur des données structurées dans le Cloud

Les méthodes que nous avons étudiées dans ce chapitre ont travaillé sur les données non structurées mais nous on se focalise sur les données structurées.

Nous avons trouvé qu'une solution[48] [49] qui travaille sur ce type de données, cette solution est basée sur le chiffrement homomorphe qui est en cours de développement. Le cryptage totalement homomorphique (FHE) permet de faire des calcule sur des entrées chiffrées sans avoir à les déchiffrer. Mais, le problème de ce chiffrement réside dans les performances de la recherche dus au coût des opérations, Aussi les questions d'application du chiffrement homomorphe à des chaines de caractères restent ouvertes, comment peut-on traduire la multiplication ou l'addition de mot par mot.

5. Conclusion

Dans ce chapitre nous avons proposé une brève revue de la littérature sur de nombreuses techniques de la recherche d'information sur des données chiffrées dans un environnement en nuage. Ces techniques ont données à la recherche d'information sur une donnée cryptée la possibilité d'être, mais malgré ça la recherche chiffrée reste jusqu'à aujourd'hui un des défis major de la recherche d'information. Le but de ce chapitre était de choisir une technique d'accélération et de l'implémenter.

Partie 2

Conception et Implémentation

Chapitre 5

Conception de l’approche proposée

1. Introduction

Ce chapitre a pour but de définir le contexte général de notre projet, et de mettre en évidence la solution proposée en précisant une description de l'approche proposée et une Analyse de sécurité qui constitue une étape fondamentale dans notre travail.

2. Description de l'approche proposée

On se focalise dans ce travail sur la recherche d'information sécurisée sur des données structurées chiffrées, alors que les méthodes étudiées dans le chapitre précédent ont travaillé sur des données non structurées.

Le processus de recherche d'information chiffrée de l'approche proposée se base sur quatre entités principales (voir la figure suivante) : le propriétaire de données, l'utilisateur de données, le serveur d'authentification et le serveur Cloud.

Remarque : Les machines des utilisateurs et des propriétaires de données sont sécurisées.

Figure 10 : Architecture de système.

2.1. Entités du processus de recherche

2.1.1. Le propriétaire des données

C'est la personne qui possède un ensemble des données structurées et qui veut les externaliser au serveur Cloud, les tâches principales de cette entité sont :

1. Le choix de la table des données à externaliser.
2. La création de l'index de la table choisie dans l'étape précédente.
3. Le chiffrement de la table avec AES.
4. Le chiffrement et la représentation de l'index sous forme d'un arbre.
5. L'externalisation de la table et des index chiffrés.
6. Et en fin, le transfert des clés aux utilisateurs autorisés via un canal sécurisé (RSA voir chapitre 3).

2.1.2. L'utilisateur des données

C'est la personne qui veut effectuer une recherche sur les données externalisées par le propriétaire, les tâches principales de l'utilisateur des données sont :

1. Formulation de la requête.
2. La création des index après avoir récupérer les clés et le dictionnaire.
3. Le chiffrement des index de la requête.
4. L'envoi de l'index chiffré au serveur Cloud.

2.1.3. Serveur Cloud

Son rôle se résume dans :

1. Le stockage des données et des index chiffrés.
2. Le calcul des résultats de la recherche.
3. L'envoi des résultats chiffrés aux utilisateurs autorisés (Le déchiffrement passe au niveau de l'utilisateur).

2.1.4. Serveur d'authentification

Les opérations effectuées par ce serveur sont :

1. Stockage de la base de données d'authentification (hachée).
2. Authentification des utilisateurs.

2.2. Processus de recherche

1. Authentification

Après l'authentification au serveur SGBD, le propriétaire et les utilisateurs accèdent à leurs espaces personnels.

2. Externalisation

A. Création de l'index

Pour que nous puissions appliquer l'indexation sur des données structurées nous proposons une technique basée sur le modèle vectoriel. Le processus d'indexation d'une table est comme suit :

A. Dans un premier temps nous choisissons sur quelle colonne (attribut) de la table l'index sera créé.

- **Exemple**

En prenant les données suivantes qui sont stockées dans une table, nous choisissons la quatrième colonne comme étant un index :

MOHAMED; BENMEDDAH; chlef; lieutenant

SID AHMED; BENNANNI; tipaza; capitaine

B. Un dictionnaire sera créé selon la colonne d'indexation choisie dans l'étape précédente, en éliminant les valeurs redondantes.

- Prenons l'exemple précédent, le dictionnaire obtenu est : (lieutenant, capitaine).

C. Chaque tuple de la table sera présenté par un vecteur binaire d, en utilisant le principe du modèle vectoriel étudié dans le premier chapitre. Cette représentation sera basée sur le dictionnaire créé précédemment.

La création de l'index est faite suivant le pseudo algorithme suivant :

Algorithme 1 créer_index_document

```
1: Entrée fichier_table, numéro_col_index, taille_dictionnaire ,nombre_ligne_table
2: Sortie matrice_index
3: Début
4: Tab_col=col_index(fichier_table,numéro_col_index);
5: Tab_dict=dictionnaire(fichier_table,numéro_col_index);
6: Entier: Matrice_index[nombre_ligne_table][taille_dictionnaire];
7: Pour j de 0 à nombre_ligne_table faire
8:     Pour i de 0 à taille_dictionnaire faire
9:         Si tab_col[j]=tab_dict[i] Alors
10:            matrice_index[j][i] ←1;
11:        Sinon
12:            matrice_index[j][i] ←0;
13:    Fin pour;
14: Fin pour;
15: Fin;
```

Figure 11 : Pseudo algorithme de la création de l'index.

- En suivant l'exemple précédent, les vecteurs d'index obtenus sont :

  Le vecteur du premier uplet est : (1 ; 0).

  Le vecteur du deuxième uplet est : (0 ; 1).

B. L'ajout des éléments fictifs

Afin de garantir la confidentialité des mots clés nous ajoutons des éléments fictifs (voir SKNN chapitre 3).

- Travaillons sur l'exemple précédent, ajoutons au hasard deux éléments fictifs dont l'index de chaque ligne sera représenté par un vecteur de quatre éléments (2 éléments de dictionnaire et 2 éléments fictifs)

1 ; 0 ; **1 ; -0.0043083117**

0 ; 1 ; **1 ; 0.009334679**

0 ; 0 ; 1 ;-0.007834712

0 ; 0 ; 1 ;-0.0039741523

C. Représentation de l'index sous forme d'un arbre

a. L'objectif de cette représentation

Si le volume des informations est important, la recherche d'information prendra beaucoup de temps. Pour accélérer et effectuer une recherche rapide et efficace nous allons représenter les index sous forme d'arbres binaires (la technique d'accélération utilisée dans l'approche «***A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data*** »).

b. Construction de l'index arborescent

Au cours du processus de construction de l'index, nous commençons par générer un nœud d'arbre pour chaque ligne de la table. Ces nœuds sont les nœuds feuilles de l'arbre d'index, chaque nœud fils possède un vecteur d'index, Ensuite, les nœuds pères d'arbre sont générés à partir de ces nœuds feuilles comme suit :

Les nœuds feuilles sont regroupés deux par deux, c'est-à-dire chaque nœud père possède deux fils (fils gauche et fils droit) et un vecteur qui représente le plus grand score de pertinence possible de ses enfants.

En suivant l'exemple précédent, nous obtenons :

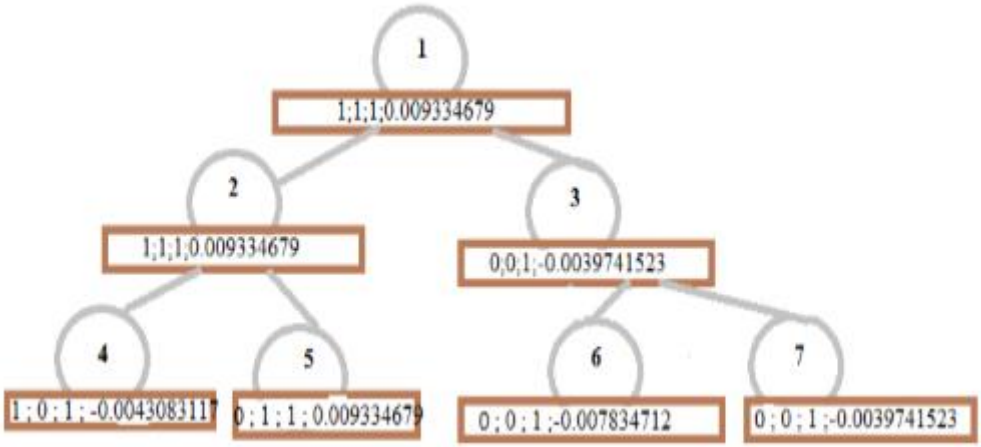

Figure 12 : Exemple d'index arborescent.

D. Chiffrement des index

Même si les données sont cryptées, il existe une possibilité d'extraire d'autres informations sensibles. Donc la recherche doit être effectuée de manière sécurisée pour permettre aux données d'être récupérées en toute sécurité. Pour cela nous utilisons la méthode SKNN pour chiffrer les index.

Remarque : la génération des clés SKNN est faite par le propriétaire des données.

Entrée : l'arbre obtenu dans la phase précédente, deux matrices aléatoires et un vecteur binaire S (clés de SKNN) :

Matrice 1

1	3	7	3
0	6	9	2
8	2	5	9
0	2	2	2

Matrice 2

9	5	7	1
5	2	0	8
9	1	2	5
5	2	3	7

Vecteur S : (1010).

Sortie : deux arbres chiffrés (les index arborescents).

Arbre 1

2.142857 ; 6.947241 ; 11.268669 ; 4.697241 ;
2.142857 ; 6.947241 ; 11.268669 ; 4.697241 ;
2.0 ; 0.4920517 ; 1.2420517 ; 2.2420516 ;
1.1428572 ; 0.66995484 ; 1.6163834 ; 1.5449549 ;
2.0 ; 6.518669 ; 10.268669 ; 4.268669 ;
2.0 ; 0.48433056 ; 1.2343305 ; 2.2343307 ;
1.3333334 ; 0.32538503 ; 0.8253851 ; 1.4920517 ;

Arbre 2

20.63596 ; 7.1793833 ; 7.778004 ; 13.297485 ;
20.63596 ; 7.1793833 ; 7.778004 ; 13.297485 ;
7.4801292 ; 0.82538503 ; 1.6547441 ; 4.1388474 ;
15.567744 ; 5.1520977 ; 7.737075 ; 5.201985 ;
11.796674 ; 2.7686694 ; 1.528004 ; 11.815343 ;
6.7108264 ; 0.7343306 ; 1.4764959 ; 3.695157 ;
7.4801292 ; 0.82538503 ; 1.6547441 ; 4.1388474 ;

E. Le chiffrement de la table

Nous allons utiliser une des techniques les plus puissantes de chiffrement qui existe dans la littérature pour chiffrer la table, dans le but de protéger les données externalisés contre les attaques. Cette technique de chiffrement est appelée AES (Voir chapitre 3).

- En suivant l'exemple précédent, nous obtenons :

POnFIuvpsyRa55LdN4B/pg==;Rgi9YGaj2FPjDeeHwrR7mQ==;eJ8qSYL4FjyMf3FJs6yo4g==;Q76Zlvdw5C5uLYaPdFXxDw== ;
KftTPwi0RHTmtNJ7xJWM1Q==;uyJxV9SSP5jlS1N4PlUJHA==;ZNp7/dRKNMjlAKkZvlPB8g==;tFIzTu2U05i7xkfEmRB4iQ==;

F. Externalisation de la table et des index chiffrés

C'est la dernière étape dans la phase d'externalisation où le propriétaire envoie les index et la table chiffrée au serveur Cloud.

3. Demande des clés

Si un utilisateur souhaite effectuer une recherche sur les données externalisées, il doit demander les clés au propriétaire.

4. L'envoi des clés

Lorsque le propriétaire reçoit des demandes des clés pour faire la recherche sur les données externalisées, il peut autoriser certains utilisateurs en leurs envoyant les clés et le dictionnaire via un canal chiffré (voir RSA chapitre 3).

5. Processus de la recherche (recherche sécurisée)

A. Formulation de la requête

L'utilisateur exprime son besoin sous forme d'une requête.

- Toujours dans le même exemple, la requête est : capitaine.

B. Création des index de la requête

Après avoir récupérer les clés de déchiffrement et le dictionnaire, nous créons l'index de la requête. Pour indexer une requête nous utilisons le principe de la représentation des documents, où la requête sera représentée sous forme d'un vecteur binaire q.

- En suivant l'exemple précédent, l'index de la requête est :

Q : (0.0 ; 1.0).

C. Le chiffrement des index de la requête :

Pour le chiffrement des index (requête) nous allons utiliser le chiffrement SKNN (voir chapitre 3).

- En suivant l'exemple précédent :

Entrée : l'index q, les matrices inversés, vecteur S (clés de SKNN)

Sortie : deux index chiffrés

Index1 : -0.047308892 ; 0.4147576 ;-0.26433903 ; 0.21780312 ;

Index2 : 3.7581503 ; 61.06753 ;- 48.54557 ; 0.65796554 ;

D. L'envoie des index chiffrés de la requête au serveur Cloud.

E. La recherche (passe au niveau du serveur Cloud)

La recherche commence par le nœud racine et atteint les nœuds feuilles (parcours par largeur), il fait des calculs pour trouver la similarité entre le document et la requête (voir SKNN chapitre 3).

Le processus de la recherche s'effectue suivant le pseudo algorithme ci-après :

```
Algorithme2 parcours_par_largeur (réel [] Q1, réel [] Q2, réel [] [] D1, réel [] [] D2, entier k)
1  : Entrée : Sum, Sum2, taille_D1, k_score, S ;
2  : Sortie : pose ;
3  : Début
4  : Sum←0 ;
5  : S←0 ;
6  : Sum2←0 ;
7  : Pour i de 0 à taille_D1      faire
8  :      Sum ←  Sum + ((D1 [k] [i]) * (Q1 [i]));
9  :      Sum2 ← Sum2+ ((D2 [k] [i]) * (Q2 [i]));
10 : Fin pour ;
11 : S←Sum+Sum2 ;
12 : pose←k ;
13 : Si  S> k_score alors
14 :       Parcours_par_largeur (Q1. Q2, D1, D2, ((2*k) +1)) ;
15 :       Parcours_par_largeur (Q1. Q2, D1, D2, ((2k) +2)) ;
16 : Sinon
17 : Écrire (pose) ;
18 : Fin ;
```

Figure 13 : Pseudo algorithme du processus de la recherche.

- En suivant l'exemple précédent, nous obtenons le résultat suivant :

Le quatrième nœud « le deuxième tuple ».

6. Résultats chiffrés

Après le calcul de la similarité, le serveur Cloud envoie aux utilisateurs autorisés les résultats pertinents chiffrés suivants.

■ **Tuple 2 :**

4KftTPwi0RHTmtNJ7xJWM1Q==;uyJxV9SSP5jlS1N4PlUJHA==;ZNp7/dRKNMjlA KkZvlPB8g==;tFIzTu2U05i7xkfEmRB4iQ==;

7. Déchiffrement des résultats

Afin de préserver la confidentialité des informations, le processus de déchiffrement passe au niveau de l'utilisateur, cette opération est effectuée à l'aide de la clé AES.

■ Les résultats retournés sont :

SID AHMED; BENNANNI; tipaza; capitaine;

3. Analyse de sécurité

L'approche proposée vise à respecter les quatre contraintes exigées dans une méthode de recherche d'information sur des données cryptées :

3.1. La protection du contenu

A. Le contenu des bases de données est préservé grâce à l'utilisation de l'algorithme de chiffrement AES.

B. Le contenu des index est assuré par l'utilisation de l'algorithme de chiffrement SKNN et l'indexation est faite par la machine du propriétaire et non par un serveur distant.

3.2. La confidentialité des mots clés

Cette contrainte est assurée par l'ajout des mots fictifs dans chaque vecteur d'index, ce qui empêche le serveur Cloud de connaître les mots clés de la base.

3.3. La non traçabilité des mots clés

La multiplication du vecteur requête par un nombre aléatoire r, l'ajout de la valeur t et les valeurs fictifs, permet de générer plusieurs variantes pour une seule requête. Ce qui rend la traçabilité de ces variantes impossible.

3.4. Protection des résultats de recherche

A. Deux requêtes peuvent avoir le même résultat bien que les vecteurs de requêtes, sont complètement différents.

B. Les résultats de la recherche ne sont pas stockés au serveur Cloud. Donc la séquence est insuffisante pour menacer les données.

4. Conclusion

Après avoir présenté le contexte général de notre travail, et la mise en place d'une démarche de développement à suivre durant la phase de la réalisation et une analyse de sécurité**,** nous passons dans le chapitre suivant à la dernière étape qui consiste à réaliser et implémenter le système étudié.

Chapitre 6

Implémentation et test

1. Introduction

Nous avons vu dans les chapitres précédents en quoi consiste notre application, et dans ce dernier chapitre nous présentons brièvement l'environnement de travail ainsi que les différents outils utilisés pour l'implémentation et le développement de notre application.

2. L'environnement de travail et les outils de développement

Nous allons présenter les différents outils nécessaires pour la mise en place de notre application.

2.1. Le système d'exploitation

Nous avons utilisé le système d'exploitation Windows 7-64 bits.

2.2. Caractéristiques de l'ordinateur

L'ordinateur qui a été exploité possède la configuration présentée dans le tableau suivant :

Tableau 2 : Les caractéristiques de l'ordinateur.

Unité	Caractéristiques
Processeur	Intel(R) Core(TM) I3-4030U Cpu @ 1.90GHz 1.90 GHz
Mémoire RAM	4.00 Go
Disque dur	500 Gb
Écran	15 pouces

2.3. Langage de programmation

Java est un langage de programmation à usage général, orienté objet dont sa syntaxe est proche du C. Il se caractérise par sa grande sécurité, la richesse de ses bibliothèques, sa simplicité,...etc, qui font de lui un langage redoutable puissant et performant.

Pour cela on a choisi java comme langage d'implémentation de notre application.

3. Présentation de l'application

Notre projet consiste à réaliser une application Multi Client-Serveur dans un réseau local, Cette application est une simulation du système de recherche d'informations sécurisée sur des données structurées chiffrées dans le Cloud, en utilisant :

1. Une nouvelle méthode d'indexation pour qu'on puisse l'appliquer sur des données structurées.
2. La méthode SKNN pour le chiffrement des index.
3. La méthode de chiffrement symétrique AES pour chiffrer et déchiffrer les fichiers envoyés au serveur Cloud.
4. La technique de l'approche "**A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data**" pour l'accélération.

3.1. Les interfaces graphiques

Nous représentons maintenant un ensemble de captures d'écran sur les principales fonctionnalités de notre application.

A. Interface d'authentification

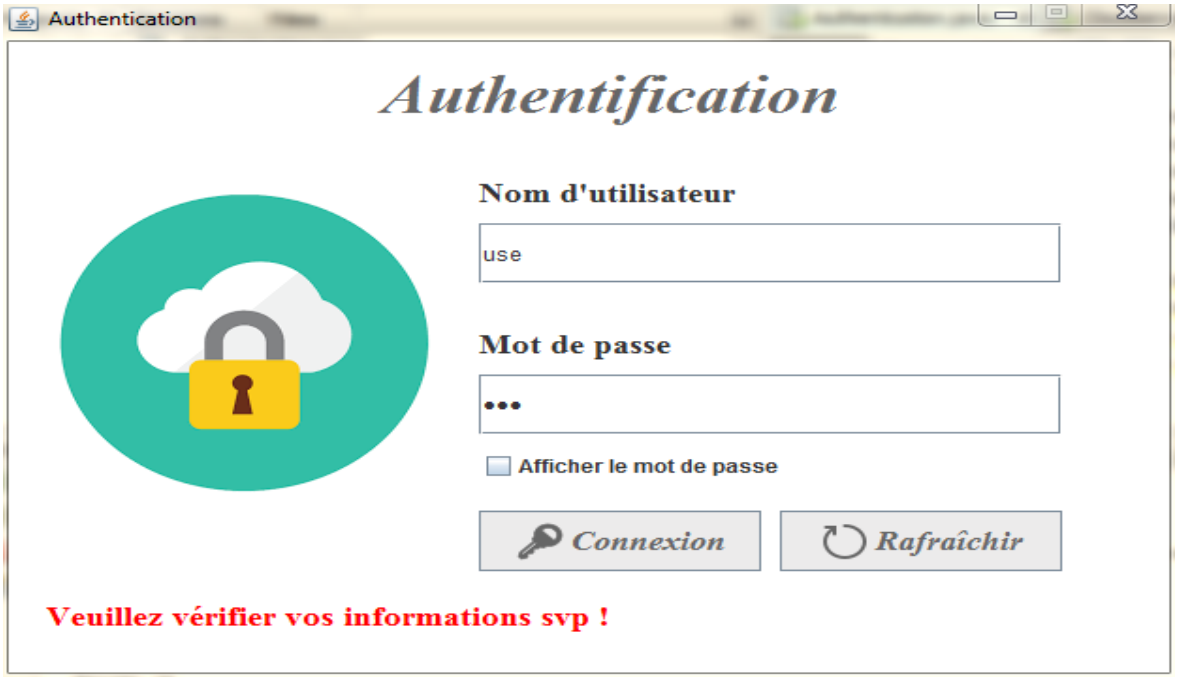

Figure 14 : Interface d'authentification.

Dans cette interface l'utilisateur doit taper son pseudo et son mot de passe correctement afin d'accéder à la fenêtre principale personnalisée selon son niveau d'habilitation (utilisateur des données / propriétaire des données) autrement une erreur d'authentification s'affichera et dans ce cas-là il faut réinsérer les informations d'une manière correcte (figure 14).

B. Interface de propriétaire de données

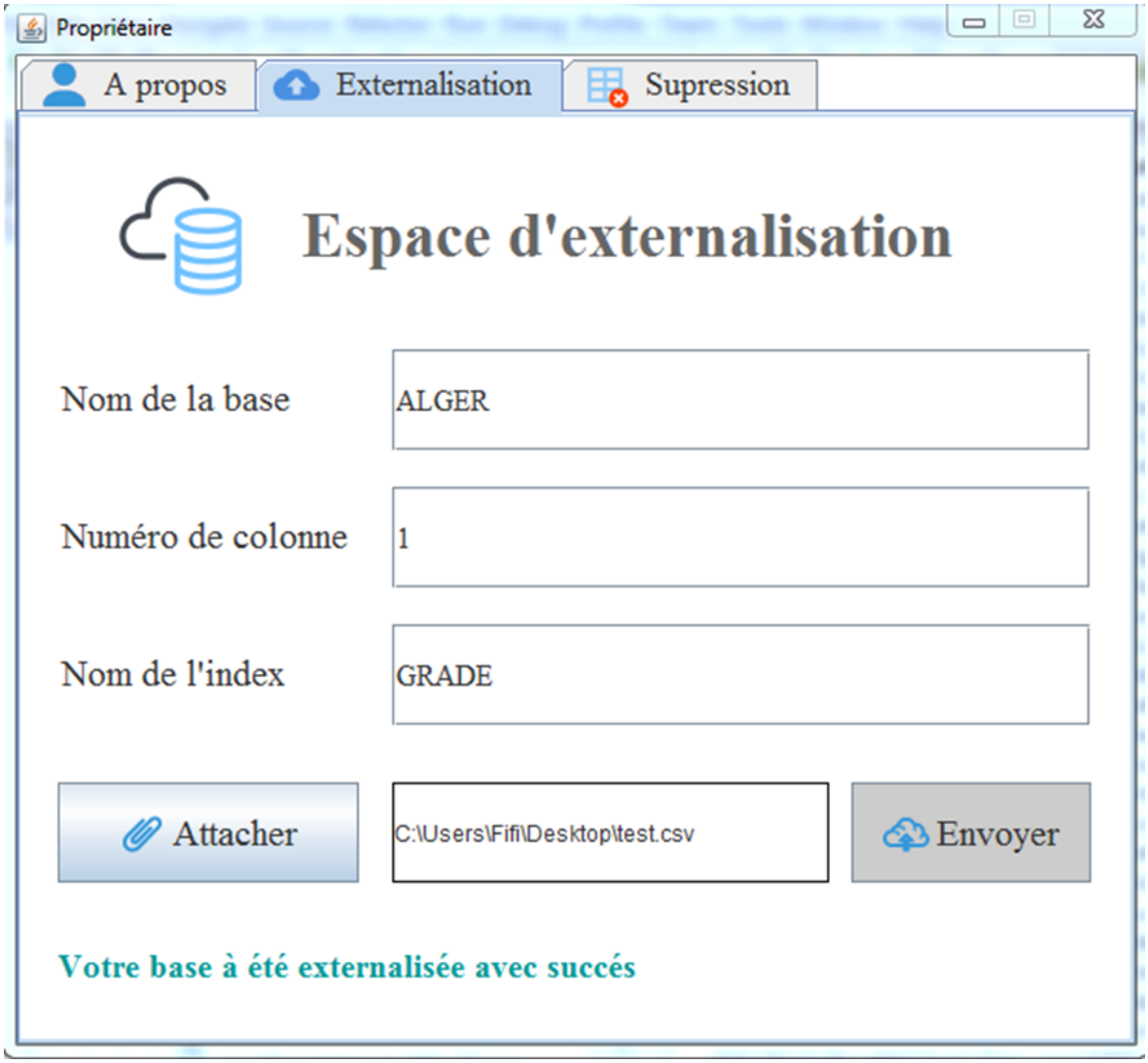

Figure 15 : Interface d'externalisation du propriétaire de données.

Cet onglet (Figure 15) représente l'espace d'externalisation du propriétaire de données pour externaliser la base de données vers le serveur Cloud, le processus se fait comme suit : Choisir le nom de la base, le nom d'index et le numéro de colonne sur lequel l'index sera créé, et pour sélectionner la base de données le propriétaire clique sur le bouton sélectionner ,après la sélection le chemin de base sera affiché dans le jText field. Pour externaliser la base de données, le propriétaire clique sur le bouton envoyer, ainsi, les opérations suivantes seront exécutées :

1. Chiffrement de la base des données
2. Création d'index.
3. Représentation de l'index sous forme d'un arbre.
4. Chiffrement de l'index.
5. L'externalisation des index et de la base chiffrée.

Après l'externalisation un message sera affiché dans l'espace d'externalisation « votre base de données a été envoyée ».

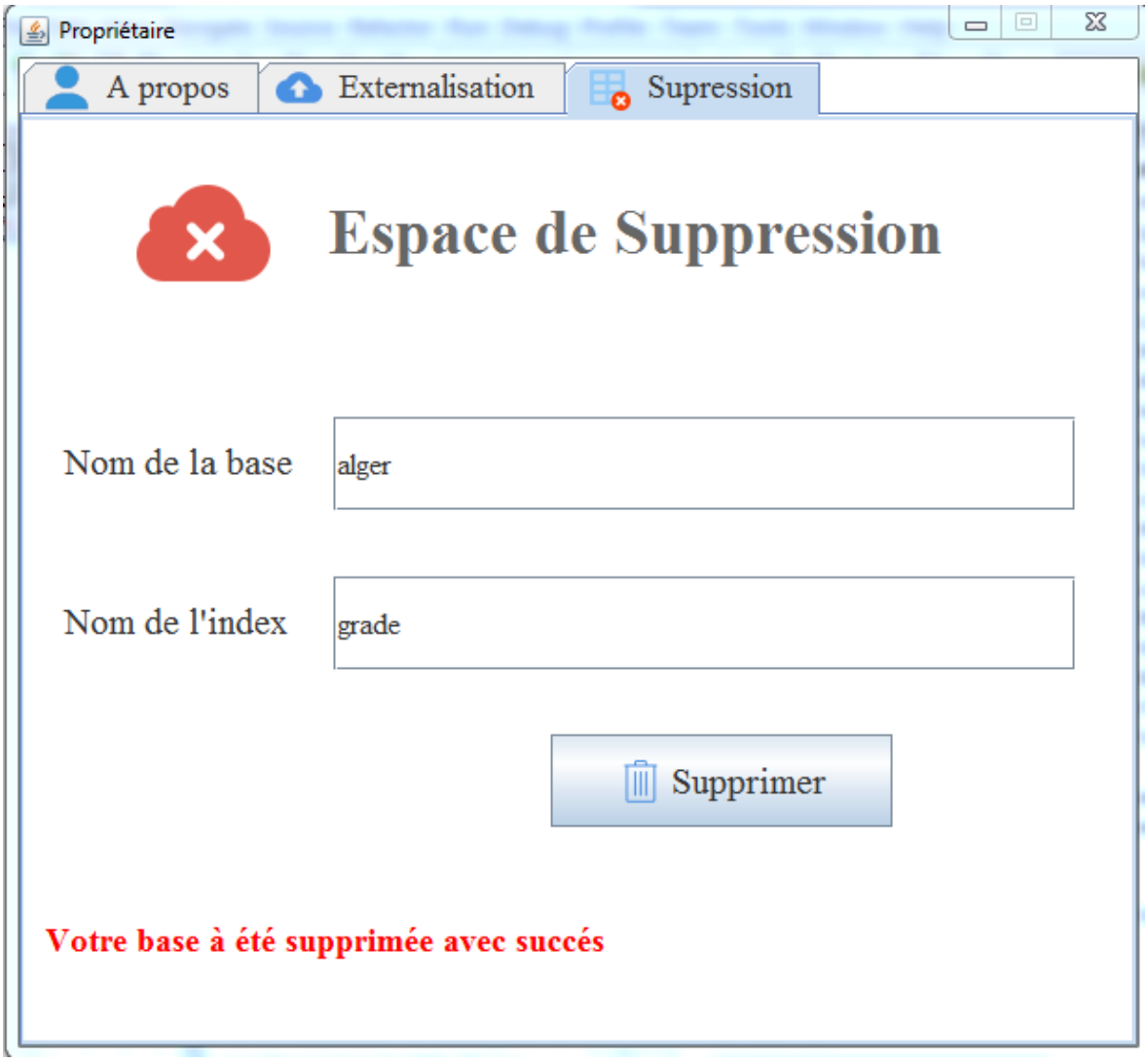

Figure 16 : Interface de suppression du propriétaire de données.

Cet onglet (Figure 16) représente l'espace de suppression du propriétaire de données pour supprimer une base de données du serveur Cloud, le processus se fait comme suit :
Choisir le nom de la base, le nom d'index, la base. Les index sont supprimés en appuyant sur le bouton supprimer.

C. Interface de serveur Cloud

Figure 17 : Interface du serveur Cloud.

Le serveur Cloud doit être activé avant de commencer à utiliser l'application par les utilisateurs. Afin de ne pas laisser le serveur Cloud exécuté en arrière-plan, nous allons le visualiser par une interface pour observer tous les processus qui sont en cours d'exécution à son niveau. Le serveur est activé en appuyant sur le bouton activer, arrêté en appuyant sur le bouton désactiver, et pour effacer les opérations affichées sur l'interface nous appuyons sur le bouton rafraîchir.

D. Interface d'utilisateur de données

Figure 18 : Interface d'utilisateur de données.

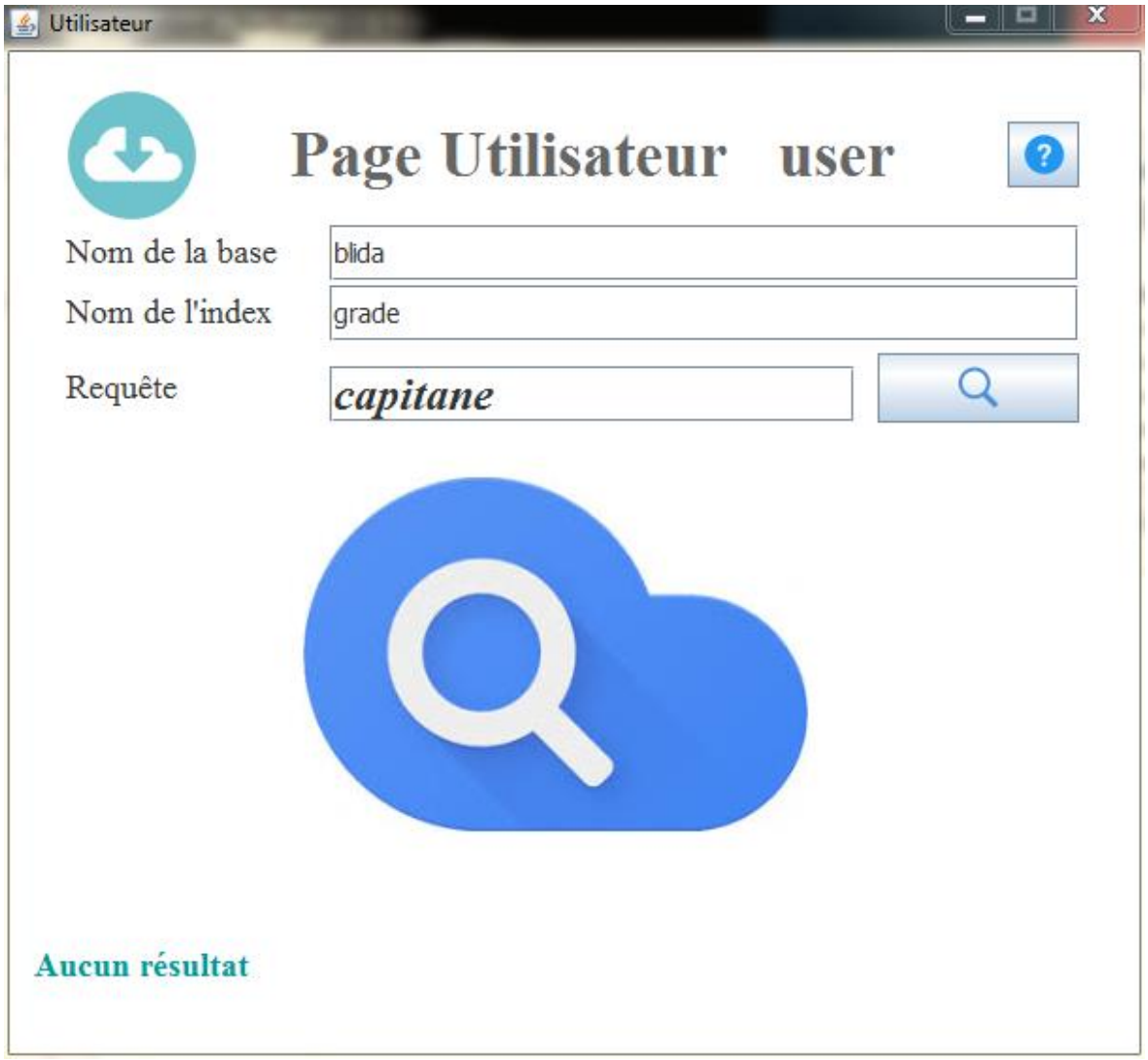

Figure 19 : Interface d'utilisateur de données.

Les Figures 18/19 représentent la page de l'utilisateur qui lui permet de faire une recherche sur les bases de données stockées dans le serveur Cloud. L'utilisateur doit saisir le nom de la base, le nom d'index et il exprime son besoin sous forme d'une requête. Pour lancer la recherche l'utilisateur doit appuyer sur le bouton rechercher. En appuyant sur le bouton, les opérations suivantes seront exécutées :

1. Création de l'index de la requête.
2. Chiffrement de l'index.
3. L'envoi de l'index chiffré au serveur Cloud.
4. Réception des résultats chiffrés.
5. Déchiffrement des résultats.

Et en fin les résultats suivants seront affichés :

1. Si les informations saisis existent (le nom de la base, l'index et les mots clés) le serveur Cloud retourne les résultats pertinents à l'utilisateur qui s'afficheront sous forme d'un tableau, le nombre total des documents pertinents sera aussi affiché.
2. Sinon le serveur Cloud envoie une réponse qui s'affichera dans l'interface de l'utilisateur «Aucun résultat ».

4. Test et performances

Les tests effectués visent à évaluer les performances de notre solution en terme de temps et du volume des données générées. Nous donnons ici quelques résultats expérimentaux avec un petit échantillon de test comparatif avant et après l'accélération.

(La taille de dictionnaire est fixée à 11 lignes et le nombre de lignes de la table est variant).

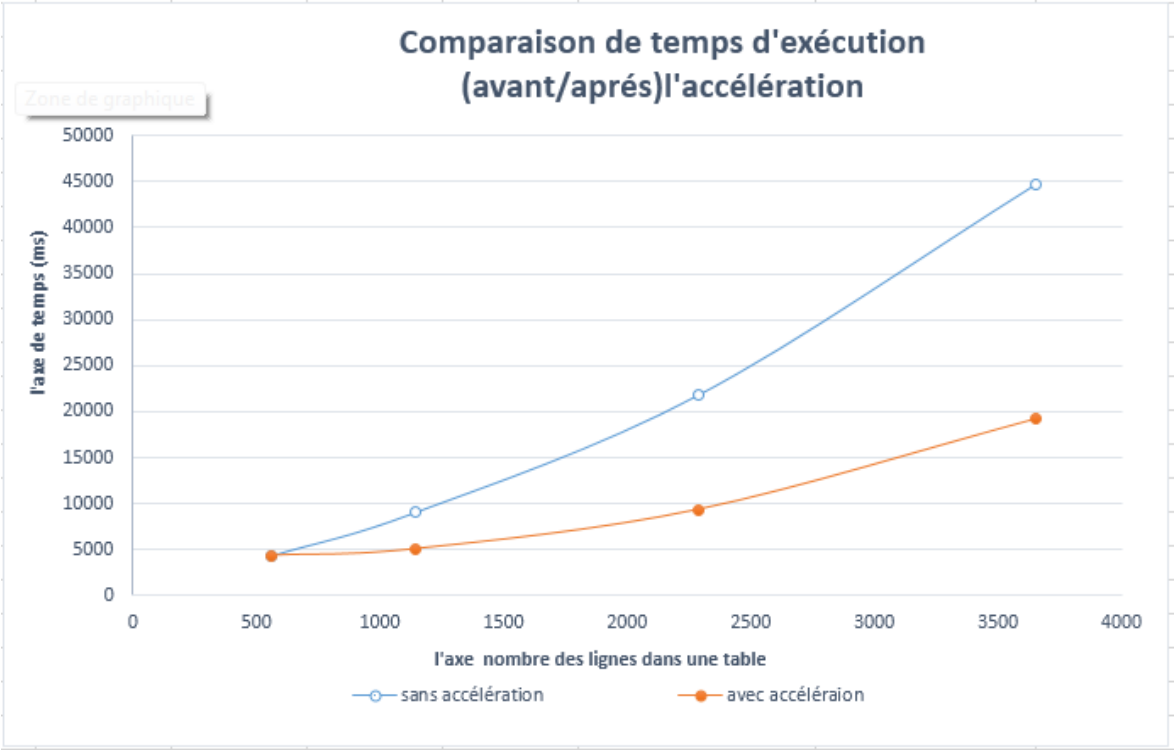

Figure 20 : Comparaison de temps d'exécution (avant/après) l'accélération.

Tableau 3 : Comparaison du temps d'exécution (avant/après)l'accélération.

Nombre de lignes	Avant l'accélération	après l'accélération
558	4366	4362
1144	9102	5087
2288	21810	9379
3652	44628	19247

5. Conclusion

Ce chapitre constitue le dernier volet de ce rapport, dans lequel nous avons présenté l'environnement de travail, et les différents résultats obtenus par la mise en œuvre de l'approche proposée. Pour finaliser cette partie on a ajouté des captures d'écran de notre application qui présentent les principales fonctionnalités de cette dernière.

Conclusion générale

A travers ce projet nous avons soulevé une problématique qui relie trois domaines d'une grande importance dans nos jours : la recherche d'information, l'externalisation des données vers le Cloud et la sécurité de ces données.

Notre solution vise à apporter une solution efficace, sécurisée, performante qui permet de renforcer l'aspect de sécurité du contenu des données externalisées et de garantir le bon fonctionnement d'une recherche d'information sur des données structurées chiffrées dans le Cloud.

Afin d'atteindre nos objectifs, on a utilisé le chiffrement AES pour chiffrer les données structurées externalisées dans le but d'empêcher toute divulgation d'informations liées aux utilisateurs en cas d'attaques visant leurs données. Et pour qu'on puisse appliquer l'indexation sur des données structurées on a proposé une méthode d'indexation basée sur le modèle vectoriel. Ces index ont été chiffrés avec la méthode SKNN pour renforcer la sécurité des index. Ainsi nous avons utilisé une technique d'accélération basée sur les arbres binaires.

Pour réaliser ce que nous avons étudié en réalité, nous avons développé une application qui simule l'approche proposée. Afin de mieux contourner toutes les fonctionnalités de chaque entité du système on a fait une conception qui était nécessaire, dans ce cadre nous avons utilisé Java comme un langage de programmation pour ses multiples avantages, nous avons fait des tests qui ont montré des résultats satisfaisants concernant l'accélération de la recherche.

Les résultats de ce travail constituent les bases d'un travail à poursuivre et à améliorer pour une étude beaucoup plus approfondie qui pourra faire l'objet d'une thèse de doctorat.

Ainsi, les perspectives futures sont dans un premier temps d'accélérer la méthode d'indexation proposée et apporter et déployer notre solution posée dans un environnement Cloud afin de toucher mieux aux résultats.

Bibliographie

[1] M. Duclos, “Méthodes pour la vérification des protocoles cryptographiques dans le modèle calculatoire,” Thèse de doctorat, Université Grenoble Alpes, 2016.

[2] J. Fattahi, “Analyse des protocoles cryptographiques par les fonctions témoins,” Thèse de doctorat, Université Laval Québec, Canada, 2016.

[3] S. Alayrangues, S. Peltier, L. Signac, “Informatique débranchée : construire sa pensée informatique sans ordinateur,” Colloque Mathématiques en Cycle 3 IREM de Poitiers, IREM de Poitiers, 2017.

[4] F. Anstett, “Les systèmes dynamiques chaotiques pour le chiffrement : synthèse et cryptanalyse,” Thèse de doctorat, Université Henri Poincaré - Nancy I, 2006.

[5] R. Dumont, “Cryptographie et sécurité informatique,” Thèse de doctorat, Université de liège, 2010.

[6] A. Bouramoul, “Recherche d'information contextuelle et sémantique sur le Web,” Thèse de doctorat, Université Mentouri de Constantine, 2011.

[7] G. Salton, M. J. McGill, “Introduction to modern information retrieval,” McGraw-Hill, New York, 1983

[8] R. Abbes, “Filtrage et agrégation d’informations vitales relatives à des entités,” Thèse de doctorat, Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), 2015.

[9] E. Znaidi, “Contribution à l’analyse et l’évaluation des requêtes expertes : cas du domaine médical,” Thèse de doctorat, Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), 2016.

[10] M. Torjmen, “Approches de recherche multimédia dans des documents semi-structurés : utilisation du contexte textuel et structurel pour la sélection d’objets multimédia,” Thèse de doctorat, Ecole Nationale d'ingénieurs Sfax, 2009.

[11] N. Zemirli, “Modèle d’accès personnalisé à l’information basé sur les Diagrammes d’Influence intégrant un profil utilisateur évolutif,” Thèse de doctorat, Université Paul Sabatier de Toulouse III U.F.R., 2009.

Bibliographie

[12] A. Hannech, “Système de recherche d’information étendue basé sur une projection multi-espaces,” Thèse de doctorat, Université du Quebec à Chicoutimi, 2018.

[13] M. Mitran, “Annotation d’images via leur contexte spatio-temporel et les métadonnées du Web,” Thèse de doctorat, Université Toulouse 3 Paul Sabatier, 2014.

[14] M. Charhad, “Modèles de documents vidéo basés sur le formalisme des graphes conceptuels pour l’indexation et la recherche par le contenu sémantique,” Thèse de doctorat, Université Joseph Fourier-Grenoble, 2005.

[15] G. Salton, “Syntactic approaches to automatic book indexing,” Department of Computer Science Cornell University, 1989.

[16] Pirkola, A, Järvelin, K, “Employing the resolution power of search keys,”Ramo-Wooldridge, Canoga Park, California, 2001.

[17] Maron, M. Earl, KUHNS, J. Larry, “On relevance, probabilistic indexing and information retrieval,” Journal of the ACM (JACM), 1960.

[18] L. Tamine, “Système de recherche d’information approché basé sur l'exploitation de techniques avancées de l’algorithmique génétique,” Thèse de doctorat, Université Paul Sabatier de Toulouse, 2009.

[19] Y. Champclaux, “Un modèle de recherche d’information basé sur les graphes et les

Similarités structurelles pour l’amélioration du processus de recherche d’information,” Thèse de doctorat, Université Toulouse III - Paul Sabatier, 2009.

[20] G. Salton, A. Wong, C. S. Yang, “A vector space model for automatic indexing,” Communications of the ACM, 1975.

[21] M.-F. Sy, “Utilisation d’ontologies comme support à la recherche et à la navigation dans une collection de documents,” Thèse de doctorat, Université de Montpellier II, 2013.

[22] F. Moreau, “Revisiter le couplage traitement automatique des langues et recherche d’information,” Thèse de doctorat, Université de Rennes 1, 2010.

[23] A. Zayed , H. Mostafa, A. Mamouni, "Cloud computing et sécurité : approches et solutions," International Journal of Research in Computer Science 2015.

Bibliographie

[24] P. Mell , T. Grance, "the NIST Definition of Cloud Computing," National Institue of Standards and Technology, 2011.

[25] T. Le vinh, "Security, Trust in Mobile Cloud Computing,"Conservatoire national des arts et metiers - CNAM, 2017.

[26] L. Schubert, K. jeffery, "Advances in Clouds-Research in Future Cloud Computing,"Report from the CLOUD Computing Expert Working Group, 2012.

[27] K. Popović, Ž. Hocenski, "Cloud computing security issues and challenges," POPOVIĆ, Krešimir et HOCENSKI, Željko. Cloud computing security issues and challenges,The 33rd International Convention MIPRO, 29 juillet 2010.

[28] D. Bozzini, "Cryptographie et Surveillance Digitale Cryptographie," Thèse de doctorat, Université de Fribourg, April 2016.

[29] D. Pointcheval, "Le Chiffrement Asymétrique et la Sécurité Prouvée," Thèse de doctorat, Université Paris VII Habilitation, 2002.

[30] P. Jouguet, "Sécurité et performance de dispositifs de distribution quantique de clés à variables continues," Thèse de doctorat, école de l'Institut Mines-Télécom - membre de ParisTech, 2013.

[31] M. Videau, "Critères de sécurité des algorithmes de chiffrement à clé secrète," Thèse de doctorat, Université Pierre et Marie Curie - Paris VI, 2006.

[32] L. Insa, M. Minier, "Cryptographie : de la théorie à la pratique Marine Minier,"Séminaire 5IF, 2005.

[33] https://www.securiteinfo.com/cryptographie/aes.shtml, 12/03/2019.

[34] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, "Enhanced Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," IEEE Transactions on parallel and distributed systems ,2013.

[35] N. Abderrahmane, "Cryptanalyse de RSA," Thèse de doctorat, Université de Caen, France, 2009.

[36] Q. Wang, S. Hu, K. Ren, J. Wang, Z. Wang, M. Du, "Catch me in the dark: Effective

privacy-preserving outsourcing of feature extractions over image data," IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, 2016.

[37] C. Wang, N. Cao, J. Li, K. Ren,W. Lou, "Secure ranked keyword search over encrypted cloud data," International Conference on Distributed Computing Systems, 2010.

[38] J. Xu, W. Zhang, C. Yang, J. Xu, N. Yu, "Two-step-ranking secure multi-keyword search over encrypted cloud data," International Conference on Cloud Computing and Service Computing, 2012.

[39] Z. Xia, X. Wang, X. Sun,Qian Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," IEEE Transactions on parallel & distributed system, vol. 27, 2016.

[40] H. Liang, X. Zhang, H. Cheng, Q. Wei, "Secure and Efficient Image Retrieval over Encrypted Cloud Data," School of Communication and Information Engineering, Shanghai University, Shanghai, China, 2018.

[41] J. Yuan, S. Yu, L. Guo, "SEISA: Secure and efficient encrypted image search with access control," IEEE Conference on Computer Communications (INFOCOM), 2015.

[42] L. Jin, W. Qian, W. Cong, C. Ning, R. Kui , L. Wenjing, "Enabling efficient fuzzy keyword search over encrypted data in cloud computing,"IACR Cryptology ePrint Archive, 2009.

[43] B. Wang, S. Yu, W. Lou, Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," IEEE INFOCOM 2014-IEEE Conference on Computer Communications, 2014.

[44] Z. Xia, N. N. Xiong, A. V. Vasilakos, X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing,"Elsevier Inc, 2017.

[45] B. Ferreira, J. Rodrigues, J. Leitão, H. Domingos,"Towards an image encryption scheme with content-based image retrieval properties," Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, Springer, Cham, 2014.

[46] N. K. Pareek, V. Patidar, K. K. Sud, "Image encryption using chaotic logistic map," Université Udaipur 313002, Rajasthan, India , 2006.

[47] W. K. Wong, D. W. Cheung, B. Kao, N. Mamoulis, "Secure kNN Computation on Encrypted Databases,"Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, 2009.

[48] M. Tebaa,, "Chiffrement Homomorphe appliqué au Cloud Bancaire," Thèse de doctorat Université Mohamed V ,2015.

[49] A. Akavia, D. Feldman, H. Shaul, "Secure Search via Multi-Ring Fully Homomorphic Encryption, Thèse de doctorat," University of Haifa, 2018.