N° d'ordre :.....

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DJILLALI LIABES - SIDI BEL ABBES

FACULTE DES SCIENCES EXACTES
DEPARTEMENT D'INFORMATIQUE

THESE DE DOCTORAT EN SCIENCE

Présentée et soutenue par
Mahmoud Fahsi

Optimisation de la recherche d'information : Protocole de Retrait Privé PIR

Dirigée par : Pr. Sidi Mohamed Benslimane
Ecole Supérieur en Informatique, SBA.

Soutenue le --/--/2017, devant le jury composé de :

Président : Pr. Mohamed Kamel Feraoun	Professeur	Université Djillali Liabes, SBA.
Examinateur : Pr. Zakaria ElBerrichi	Professeur	Université Djillali Liabes, SBA.
Examinateur : Dr. Mohamed Reda Hamou	M.C.A.	Université Taher Moulay, Saida.
Examinateur : Dr. Djelloul Bouchiha	M.C.A.	Centre Universitaire de Naama.
Examinateur : Dr. Djamila Hamdadou	M.C.A.	Université d'Oran 1.

Année universitaire : 2016-2017

À ma chère mère,
À mon père,
À mon défunt frère,
À ma sœur et frère,
À ma femme et à tous qui
me sont chers.

I. Remerciements

Je voudrais tout d'abord exprimer mes plus profonds remerciements à mon encadreur Pr. Benslimane Sidi Mohammed pour son soutien moral et scientifique efficace et constant, pour sa disponibilité et son écoute. Je remercie infiniment mon professeur Lehirech Ahmed pour son encouragement continue, sa disponibilité, ses conseils pendant ces dernières années, et surtout pour son grand cœur et sa bonne foi en moi.

Je remercie aussi Pr. MALKI Mimoun, ancien directeur du laboratoire EEDIS, ainsi que Dr. Boukli Hassen Sofiane, l'actuel directeur pour leurs conseils et leurs encouragements.

Je tiens aussi à remercier tous les membres de jury pour leur disponibilité et acceptation de faire partie de ce jury, d'examiner et de rapporter mon travail.

Je ne serai pas reconnaissant si j'oublie de remercier Dr. ADJOUJ Reda et M. GHAZI Ahmed, pour leurs conseils qui ont été très bénéfiques pour moi.

Je ne saurais oublier de remercier mon oncle FAHSI Habib, et tout la famille.

Sans oublier mes collèges de l'université Djilali Liabes de Sidi Bel Abbes. Ainsi que mes collègues de l'université Taher Moulay de Saida, pour leurs encouragements et précieuses orientations pendant toute la période de l'élaboration de ce travail. Surtout Pr. Amine Abdelmalek qui été à l'origine de mon expérience à Saida.

Enfin, que tous ceux qui directement ou indirectement m'ont apporté leur aide, trouvent ici l'expression de mes sincères remerciements.

II. Sommaire

III. Liste des Figures

IV. Liste des Tableaux

V. Résumé

L'utilisation professionnelle de stockage de données du domaine sanitaire dans les Cloud implique des extensions de recherche d'information. Cependant, ces extensions doivent offrir une protection contre des menaces existantes, par exemple, des pirates informatiques, des administrateurs de serveur et les prestataires de services qui utilisent des données personnelles des gens pour leurs propres buts. En effet, les serveurs Cloud maintiennent les traces d'activités d'utilisateur et des requêtes, ce qui met en péril la sécurité des données utilisateur contre des pirates informatiques de réseau. Ils peuvent même utiliser ces traces pour adapter ou personnaliser leurs plateformes sans accords des utilisateurs.

Dans cette thèse, nous nous intéressons à l'application du chiffrement homomorphe au Cloud Computing, particulièrement au Cloud des données médicales, afin d'assurer la confidentialité des données sensibles des patients stockées dans les serveurs distants, et gérées par les fournisseurs de Cloud. Nous étudions et comparons les durées de chiffrement, de déchiffrement et de traitement des cryptosystèmes homomorphes existants. Nous proposons un Framework de retrait d'information privé, qui implémente des protocoles de chiffrement homomorphes sur le corpus de rapports médicaux destiné à la recherche d'information. Nous étudions l'efficacité de cette solution par une évaluation de temps de recherche d'information. Les résultats expérimentaux montrent que notre approche assure un niveau raisonnable et acceptable de confidentialité pour la récupération de données dans le cloud.

VI. Abstract

Professional use of cloud health storage around the world implies Information-Retrieval extensions. These developments should help users find what they need among thousands or billions of enterprise documents and reports. However, extensions must offer protection against existing threats, for instance, hackers, server administrators and service providers who use people's personal data for their own purposes.
Indeed, cloud servers maintain traces of user activities and queries, which compromise user security against network hackers. Even cloud servers can use those traces to adapt or personalize their platforms without users' agreements.
For this purpose, we suggest implementing Private Information Retrieval (PIR) protocols to ease the retrieval task and secure it from both servers and hackers. We study the effectiveness of this solution through an evaluation of information retrieval time, recall and precision. The experimental results show that our framework ensures a reasonable and acceptable level of confidentiality for retrieval of data through cloud services.

VII. ملخص

الاستخدام المهني لتخزين المعلومات الصحية في السحابة المعلوماتية العالمية يتطلب استحداث لملحقات استرجاعها. وينبغي لهذه التطورات أن تساعد المستخدمين على العثور على ما يحتاجونه من بين الآلاف أو الملايين من وثائق المشاريع والتقارير. ومع ذلك، يجب توفر ملحقات حماية ضد التهديدات القائمة، على سبيل المثال المتسللين، مسؤولي الخادم ومقدمي الخدمات الذين يستخدمون البيانات الشخصية للناس لأغراضهم الخاصة. في الواقع، خوادم السحابة قوم بالحفاظ على آثار للأنشطة المستخدم والاستفسارات، هذا الذي يهدد أمن المستخدم ويعرض حسابه لقراصنة الشبكة. كما يمكن للخوادم السحابية استخدام تلك آثار لتكيف أو تخصيص برامجها دون اتفاقات مع المستخدمين.

لهذا الغرض، نقترح تنفيذ خاصة استرجاع المعلومات باستخدام بروتوكولات PIR لتسهيل مهمة البحث والاسترجاع للوثائق وضمان الحصول عليها بسرية. ندرس فعالية هذا الحل من خلال تقييم وقت استرجاع المعلومات، الحجم والدقة. أظهرت النتائج التجريبية ان نظامنا يضمن مستوى معقول ومقبول من السرية لاسترجاع البيانات من خلال الخدمات السحابية.

Chapitre I.

Introduction générale

Chapitre I. Introduction générale

Le Cloud Computing a été largement reconnue comme une des technologies les plus influentes à cause de ses avantages sans précédent. Avec la virtualisation des ressources, le cloud peut fournir à la demande un self-service, un accès au réseau omniprésent, une élasticité rapide des ressources et des prix à base de taux d'utilisation, qui rendent ainsi les services du Cloud aussi commodes que les besoins de la vie quotidienne telles que l'électricité, l'eau et le gaz. Malgré ses avantages économiques largement reconnus, le Cloud Computing suspend le droit de contrôle direct des clients sur les systèmes qui dirigent leurs données, ce qui soulève des inquiétudes significatives sur la sécurité est la confidentialité considérée comme l'obstacle majeur à son adoption. Ainsi, les approches de cryptographie peuvent perfectionner cette technologie et répondre aux questions de sécurité. Néanmoins, ces derniers introduisent des couts supplémentaires comme le temps excessif de calcul et le stockage excessif, ce qui pourrait réduire de façon significative les avantages économiques du cloud. Ainsi, l'implémentation d'une approche de cryptographie qui renforce la sécurité et la confidentialité des infrastructures du Cloud computing est un grand challenge qui doit prendre en compte la disponibilité ainsi que les performances du système.

1.1 Background

Les avancées actuelles des infrastructures réseaux et le besoin exponentielle en ressources de traitement et de stockage a promu un changement du paradigme de déploiement et de présentation classique des services informatiques et à impliquer l'Outsourcing (externalisation) des ressources informatiques. Ce nouveau modèle appelé Cloud Computing est un modèle de service qui offre à ces clients, un accès distant à un grand ensemble de ressources informatiques partagées, fournies par le Cloud Service Provider (CSP). Ce dernier propose trois types de service de Cloud Computing ;

- Infrastructure as a Service (IaaS), où un client gère l'ensemble des ressources de traitement, de stockage et de communication que le fournisseur CSP offre.
- Plateforme as a Service (PaaS), où les clients gèrent les ressources du CSP uniquement pour déployer leurs applications personnalisées.
- Software as a Service (SaaS), où les clients ne sont que des utilisateurs des applications disponibles dans la plateforme du Cloud.

Ces infrastructures peuvent être privées ou publiques. Dans le cloud privé, l'infrastructure matérielle et logicielle est sous le contrôle du client. Tandis que dans un cloud public, l'infrastructure est une propriété possédée et gérée par le fournisseur de service. Par conséquent, l'infrastructure du cloud publique se trouve à l'extérieur de l'espace géré par le client. À l'heure actuelle, les avantages économiques du cloud computing ont été largement reconnus, surtout les bénéfices d'utilisation des cloud public. L'externalisation des données et des traitements vers un cloud public promet d'offrir des avantages sans précédent comme des

accès réseau omniprésents, une élasticité des ressources à la demande, des frais minimaux de gestion et de maintenance, etc. Pour la plupart des clients, la possibilité d'utiliser une infrastructure et de payer ce service à la demande est une grande motivation pour l'utilisation du cloud public.

En dépit des avantages mentionnés ci-dessus, le cloud public prive le contrôle direct des clients sur les systèmes qui gèrent leurs données et leurs applications (Dimitrios & Dimitrios, 2012) (Jinhui , et al., 2010), ce qui soulève des risques importants en matière de sécurité et de confidentialité. D'une part, bien que les infrastructures cloud soient très puissantes et fiables que les périphériques personnels, un large éventail de menaces internes et externes existe toujours, y compris les pannes matérielles, les bugs logiciels, les coupures de courant, la mauvaise configuration du serveur, etc. D'autre part, le fournisseur de service CSP a ses motivations qui le poussent à se méfier des clients, ou même à augmenter la marge bénéficiaire en réduisant les coûts. Par exemple, le CSP peut délibérément transférer les données des utilisateurs vers un stockage plus lent mais moins cher ou même peut délibérément supprimer certains blocs de données qui sont rarement ou jamais utilisés. De plus, le CSP peut même tenter de cacher des incidents de défaillance afin de maintenir une réputation (Lakshmi, et al., 2007).

À ce jour, bien que des utilisateurs acceptent de négliger l'aspect sécurité et confidentialité de leurs données contre la commodité offerte par les services Cloud computing ou Cloud de stockage tel que Dropbox et Google Drive, ce n'est pas le cas pour les entreprises et les organisations gouvernementales où les données sont nettement plus critiques (Par exemple, les dossiers médicaux, les données commerciales…etc.). Donc, même si le Cloud public est promoteur, de nombreux clients potentiels seront réticents à externaliser leurs données à moins que les préoccupations de sécurité et de vie privée soient bien prises en compte.

Par conséquent, afin d'améliorer l'adoption du cloud computing, il est souhaitable que le cloud public prévoie des garanties de sécurité des données de ces clients. Il existe des approches cryptographiques qui ont été proposées comme solutions pour atteindre les objectifs de sécurité dans le cloud computing. Parce que par rapport aux mécanismes juridiques ou de sécurité physique, les approches cryptographiques peuvent fournir aux clients un certain degré de contrôle sur leurs données externalisées. Plus précisément, les méthodes de cryptographie fournissent les propriétés de sécurité essentielles à un système sécurisé, à savoir la confidentialité, l'intégrité, l'authentification et la non-répudiation :

- La confidentialité garantit que l'information est inaccessible aux utilisateurs non autorisés ou aux systèmes.
- L'Intégrité signifie que les données en transit ou stockées ne peuvent être modifiées par des utilisateurs ou des systèmes non autorisés.
- Authentification est utilisée pour s'assurer que les parties impliquées dans une communication sont vraiment ceux qu'ils prétendent être.
- La non-répudiation assure qu'un utilisateur ne peut pas nier à un stade ultérieur ses opérations précédentes sur n'importe quelles données.

1.2 Position de problème :

Bien que les approches cryptographiques puissent assurer les objectifs de sécurité du cloud computing, cela pourrait réduire considérablement leurs performances et donc rendre difficile le déploiement des services traditionnels qui traitent des données en claire. Par exemple, le chiffrement traditionnel des données dans le cloud rend inefficace l'exploitation des répliques des données lorsque le serveur effectue la déduplication pour économiser de l'espace de stockage. De plus, les méthodes classiques de chiffrement ne permettent pas d'effectuer des recherches sur des données chiffrées en raison de la protection de la confidentialité des données et entraînent donc des coûts supplémentaires pour l'utilisateur et le serveur. Par conséquent, il est souhaitable d'utiliser des approches cryptographiques qui assurent les objectifs de sécurité sans causer de surcharge significative aux infrastructures de Cloud Computing existantes. Dans cette thèse, nous nous concentrons principalement sur la mise en place d'un Framework de recherche d'informations stockées dans le cloud afin d'assurer la confidentialité des requêtes et des résultats dans un cloud de stockage.

1.2.1. Stockage de données dans le Cloud

Selon les prévisions de l'International Data Corporation (IDC), le volume de données dans le monde atteindra 40 billions de giga-octets en 2020 (Gantz & Reinsel, 2012). D'autre part, l'application cloud la plus sollicitée est l'externalisation du stockage de données[1,2,3], où les individus et les entreprises stockent leurs données à distance sur le cloud pour alléger le fardeau de gestion du stockage et obtenir une utilisation de données beaucoup plus flexible.

En dépit des avantages attrayants apportés par ce service, il est sujet d'une grande hésitation. La principale raison est la préoccupation de sécurité des données car les utilisateurs ne possèdent plus physiquement leurs données (Lakshmi, et al., 2007) (Miller, 2010). Par conséquent, même si l'externalisation du stockage de données est économiquement intéressante pour une gestion à long terme des données à grande échelle, l'adoption généralisée des cloud de stockage peut être restreinte sans des garanties de la confidentialité et de la disponibilité des données.

1.2.2. Recherche d'information dans le Cloud

Outre l'élimination du fardeau de gestion des supports de stockage local, le stockage des données dans le cloud ne sert à rien, à moins qu'elles ne puissent être récupérées efficacement et en toute sécurité pour être utilisées. Dans cette thèse, nous visons également la mise en place de système de recherche d'information et de transmission des résultats sécurisés et confidentiels.

En réalité, les utilisateurs finaux ne peuvent pas entièrement faire confiance aux fournisseurs CSP et préfèrent crypter leurs données avant de les charger sur les serveurs

[1] Dropbox. http://www.dropbox.com/.
[2] GoogleDrive. http://drive.google.com/.
[3] Bitcasa. http://www.bitcasa.com/.

cloud afin d'assurer la confidentialité des données. Cela rend généralement l'utilisation des données plus difficile par rapport au stockage traditionnel où les données sont stockées en claire. Une solution triviale consiste à télécharger toutes les données et à les déchiffrer localement. Néanmoins, cette solution est coûteuse en termes de bande passante et d'espace de stockage nécessaire pour chaque requête et est donc clairement impraticable. Une autre solution typique est de prédéfinir des mots-clés pour chaque document chiffré et un utilisateur peut rechercher les documents chiffrés avec un jeu mots-clés. Toutefois, les mots-clés prédéfinis ont plus de résultats et donc ont besoin de plus d'espace de stockage. En plus, les problèmes de confidentialité seront partiellement résolus car les mots-clés qui annotent les données cryptées peuvent divulguer des informations et compromettre la confidentialité des données. Par conséquent, les mots clés doivent être chiffrés pour protéger confidentialité.

De la même manière, quand l'utilisateur effectue une recherche sur des données dans son espace de stockage Cloud, la requête peut divulguer des informations aux gestionnaires du cloud ou à d'autres parties non fiables. En outre, la réponse à une requête de recherche ne devrait pas être longue. C'est-à-dire que le traitement de l'appariement (matching) de la requête et les données chiffrées ne doit pas affecter la disponibilité des serveurs CSP. Cela implique le besoin de développer des techniques de recherche efficaces et sécurisées sur des données chiffrées. De telles techniques devraient prendre en compte les fonctionnalités et les modèles de recherche d'information de la littérature des systèmes de recherche d'information moderne comme Google, Bing, etc. L'adaptation de ces techniques est importante pour assurer le succès des cloud de stockage qui préserve la vie privée des individus et des organisations.

De plus, dans le cloud, les propriétaires de données et les serveurs cloud sont probablement dans deux domaines différents. Dans le cloud public en particulier, les ressources de données ne sont pas physiquement sous le contrôle total du propriétaire. Par conséquent, le CSP doit transmettre les données via le réseau. Ce dernier est habituellement un réseau public car il est beaucoup moins cher et plus commode qu'un réseau privé ou qu'une ligne louée qui offre de meilleures garanties de sécurité. Cependant, le canal de communication public est vulnérable et par conséquent instable par défaut pour la transmission de message. Dans ce cas, un chiffrement des données à transmettre peut jouer le rôle de réseau privé virtuel VPN établis entre l'utilisateur et l'infrastructure du cloud.

1.3 Contributions

Dans cette thèse, nous nous intéressons à l'application du chiffrement homomorphe au Cloud Computing, particulièrement au Cloud des données médicales, afin d'assurer la confidentialité des données sensibles des patients stockées dans les serveurs distants, et gérées par les fournisseurs de Cloud. Pour le faire, nous avons proposé un Framework de retrait d'information privé qui implémente des protocoles de chiffrement homomorphes sur le corpus Medline. Ce dernier étant un ensemble de rapports médicaux destinés à la recherche d'information. Nous étudions les cryptosystèmes homomorphes existants, et nous comparons les durées de chiffrement, de déchiffrement et de traitement des cryptosystèmes

homomorphes choisis, afin de choisir l'algorithme adéquat à notre application, avec les tailles de clés optimales permettant un temps de traitement convenable.

Le travail a fait objet des productions scientifiques (Fahsi, et al., 2015), (Fahsi & Benslimane, 2014a) et (Fahsi & Benslimane, 2014b) suivantes :

- Mahmoud Fahsi, Sidi Mohamed Benslimane, Amine Rahmani. A Framework for Homomorphic, Private Information Retrieval Protocols in the Cloud. International Journal of Modern Education and Computer Science (IJMECS). pp. 16-23, 7(5), 2015. ISSN : 2075-0161 (Print), ISSN : 2075-017X (Online).
- Mahmoud Fahsi, Sidi Mohamed Benslimane, Homomorphic Private Information Retrieval Protocol for secure Data Warehouse Access. The First International Symposium on Informatics and its Applications (ISIA2014), February 25-26, 2014, M'sila, Algeria.
- Mahmoud Fahsi, Sidi Mohamed Benslimane. Studying the effects of conflicting Tokenisation on LSA dimension reduction. The 3rd International Conference on Multimedia Computing and Systems (ICMCS'14), April 14-16, 2014, Marrakesh, Morocco.

1.4 Organisation de la thèse :

Le reste de cette thèse est organisé comme suit :

Dans le chapitre (1) et (2), nous présentons les notions préliminaires sur la recherche d'information et le cloud computing. Ces notions seront utilisées tout au long de cette thèse.

Dans le chapitre 3, nous donnons une description approfondie des protocoles de retrait d'information privé, largement utilisés afin d'assurer la confidentialité des requêtes et des résultats de recherche sur des données chiffrées. Nous parlons aussi des mécanismes de sécurité dans le Cloud qui est un sous domaine du cloud computing (informatique dans les nuages). Cette sécurité engendre les notions de la sécurité des réseaux, sécurité du matériel et les stratégies de contrôle déployées pour protéger les données, ainsi que les applications et l'infrastructure liée au cloud computing.

Dans le chapitre 4, un état de l'art des algorithmes du Chiffrement Homomorphes appliqué dans le cloud est exposé. En plus, une comparaison entre les différentes approches PIR est donnée afin de mieux positionner notre contribution.

Dans le chapitre 5, nous proposons un Framework PIR que nous avons implémenté dans un simulateur cloud pour assurer la confidentialité de retrait d'information. Une section de discussion de résultats nous permettra d'évaluer les techniques par rapport au temps d'exécution.

Enfin, nous concluons ce manuscrit en présentant le bilan de nos contributions, ainsi que les perspectives futures envisageables.

Chapitre II

Indexation et recherche D'information

Chapitre II. Indexation et recherche D'information

2.1 Introduction

La Recherche d'Information (RI) n'est pas un domaine récent, il date des années 60. Une des premières définitions de la RI a été donnée par Salton : « la recherche d'information est un domaine qui a pour objectif, la représentation, l'analyse, l'organisation, le stockage et l'accès à l'information » (Salton, 1983).

Plusieurs tâches se regroupent sous le vocable de la RI, la plus ancienne est la recherche documentaire, nous y trouverons également d'autres tâches plus au moins récentes comme : le filtrage d'information, l'extraction d'information, la recherche d'information multilingue, les questions réponses, la recherche d'information sur le web, etc.

Dans ce chapitre, une présentation du domaine de la RI sera fournis. Dans sa première partie, 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, de recherche et de reformulation de requêtes ; ainsi que, les modèles de RI. Dans la seconde partie, nous passerons en revue les techniques de recherche d'information sémantique. Dans la dernière partie de ce chapitre est discutée l'évaluation des systèmes de recherche d'information.

2.2 Indexation

L'indexation consiste à représenter un ou des documents sous une forme plus simple à manipuler pour les rendre facilement exploitables par rapport à un domaine donné. L'indexation est une méthode habituelle qui précède la recherche d'information dans les documents textuels ou « chaque document sera décrit par une séquence structurée ou non structurée de mots clefs et/ou descripteurs ». L'index du document est le résultat de ce traitement. Par la suite, une recherche est réalisée à travers une requête sous la forme d'une séquence de descripteurs empruntés au même vocabulaire que celui utilisé pour l'indexation, par la comparaison du contenu de la requête et le contenu de l'index des documents afin de proposer les documents dont l'index coïncide totalement ou partiellement avec la requête.

L'indexation est habituellement globale, car les données cibles d'indexation étaient des livres ou documents entiers, faisant partie d'un rayon dans une bibliothèque, et qu'il n'était pas possible de manipuler une partie du document sans avoir à en manipuler le tout : le bibliothécaire ou le documentaliste utilise l'index pour sortir le livre du rayonnage.

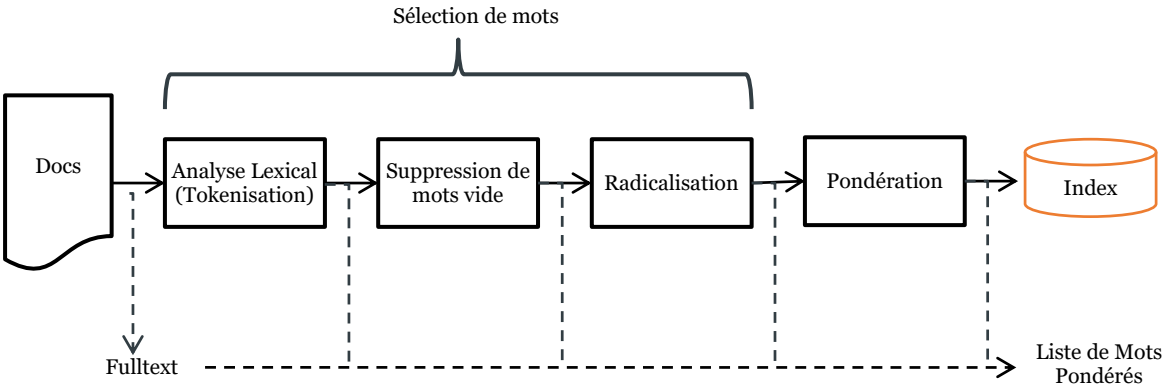

Figure 2-1: Processus d'indexation

Comme l'indique la figure 2.1, les mécanismes d'indexation s'intéressée à chaque document dans son intégralité, pour on formaliser son contenu global : en premier temps, une analyse de contenu selon l'usage visé est effectuée, ensuite la taille du résultat de l'analyse est réduite à l'aide de vocabulaire contrôlé, pour obtenir l'index du document. C'est un processus formalisé qui a comme but « l'extraction du sens des documents » (Faloutsos, 1985). Autrement dit aboutir à une représentation formelle du contenu global d'un document.

En réalité, une telle définition paraît optimiste et même ambiguë car elle sous-entend une définition du "sens" d'un document. Alors qu'on ne peut pas "extraire" automatiquement le sens exact d'un document. Ainsi, la plupart des experts définissent en fonction de leurs objectifs particuliers ce qu'ils perçoivent par représentation du contenu d'un document (Salton, 1983).

Pour avoir une idée sur la notion d'indexation, nous présentons dans ce qui suit quelques définitions terminologiques les plus fréquentes.

2.2.1. Étapes d'indexation

L'analyse lexicale : La première étape est l'analyse lexicale qui permet de transformer un contenu textuel d'un document en un ensemble des termes nommés aussi « lexème ». Durant cette étape, la ponctuation, la casse, et la mise en page sont supprimées.

Le nettoyage de collection par élimination de mots vides : Afin de supprimer les termes sans importance, plusieurs techniques peuvent être utilisé parmi celles-ci, nous utilisions souvent les anti-dictionnaires (Stop List) qui permettent de ne pas conserver les mots sans sens (mots vides) c'est-à-dire sans contenu informationnel des documents. La liste de mots vides contient souvent les mots prépositions, les mots outils, articles, pronoms, ainsi que les mots athématiques. Un mot athématique fait partie des mots présents dans le document pour le présenter ou l'introduire mais sans rapports avec le sujet traité comme le contenu de la balise HTML « head ». L'utilisation des anti-dictionnaires est très simple.

Quand un mot-vide est rencontré dans un texte à indexer, il n'est pas considéré comme un mot a index. Tous de même, la qualité de la recherche est influencée par la suppression des mots vides, donc elle doit être contrôlée. Il est aussi clair que le rôle d'un mot dans un document découle du contexte dans lequel il est employé, et qu'il peut avoir un pouvoir d'information différent dans un autre contexte. Ainsi, un anti dictionnaire devrait être dépendant du domaine d'application. Néanmoins, en pratique nous utilisions souvent une liste de mots vide relative à la langue d'application.

Le nettoyage des mots par élimination des suffixes/préfixes : La plupart du temps, les variantes morphologiques des mots ont un sens très proche. On plus, un mot peut avoir deux variantes possibles ; une variante morphologiques (Frakes & Baeza-Yates, 1992) ou sémantiques (Paice, 1996). Pour retrouver des documents contenant le mot recherché ou une ou plusieurs de ces variantes, il est intéressant d'éliminer les préfixes et suffixes considérés non significatives et de garder la racine (le lemme ou radicale) qui représente la partie commune. En effet le radical d'un terme est une simple diminution du nombre de lettres. Par exemple, le mot « calculateurs » peut être représenté par plusieurs radicaux « calcula », « calculer », « calculateur », sa racine linguistiquement correcte étant « calcule ».

Ajouter une pondération formelle : La pondération d'un mot de l'index est une association de valeurs numériques à ce mot. Cette pondération exprime la valeur informative du terme pour un document donné. Elle ajoute une information sur la valeur du mot dans un document par rapport aux autres mots du même document ainsi que par rapport aux autres mots des autres documents. Au sens commun c'est « mettre au courant de quelque chose, donner connaissance d'un fait » (Guiraud, 1967). Salton et autre (Salton, 1987) explique et compare comme suit les différentes concepts d'une pondération basiques en RI :

- 0 ou 1 : exprime la présence (1) ou l'absence (0) d'un terme dans le document représenté par un modèle probabiliste.
- 0 ou x : exprime la présence de (x) ou l'absence (0) d'une occurrence d'un terme dans le document représenté par un modèle statistique.
- *tf* : term-frequency est la fréquence du terme dans le document c'est-à-dire le nombre d'occurrences d'un terme dans le document.
- *idf* : Inverse of Document Frequency est la fréquence absolue inverse. C'est un facteur qui varie inversement proportionnel au nombre n de documents où un terme apparaît dans une collection de N documents.

De la même manière, Salton exprime la fréquence absolue inverse comme suit :

$$\mathrm{idf} = \log(N/n) \tag{1}$$

Avec N : nombre total de documents dans une collection et n : nombre de documents où le terme parait.

Par la suite, Le poids d'un terme j dans le document i s'écrit suivant comme suit (Jones, 1972):

$$\text{poids}(j) = tf_{ij} \times idf_j \ (4) \qquad (2)$$

Avec tf_{ij} : la fréquence d'apparition du terme j dans le document i et idf_j: la fréquence absolue inverse du terme j dans la collection.

Salton explique que le poids d'un terme va augmenter si celui-ci est fréquent dans le document et décroître si celui-ci est fréquent dans la collection. La formule (2) est appelé $tf \times idf$ et elle fournit une bonne représentation du poids pour les corpus dont les documents sont de taille homogène c'est-à-dire composés de documents de tailles similaires (Salton, 1987).

2.2.2. Types indexations

L'ensemble des mots de l'index forme le vocabulaire de l'index, appelé aussi langage d'indexation. Généralement, le langage d'indexation se compose d'un vocabulaire et d'une syntaxe. Au côté du vocabulaire, la syntaxe est l'ensemble des règles qui gèrent la construction correcte des expressions à partir de plusieurs éléments du vocabulaire.

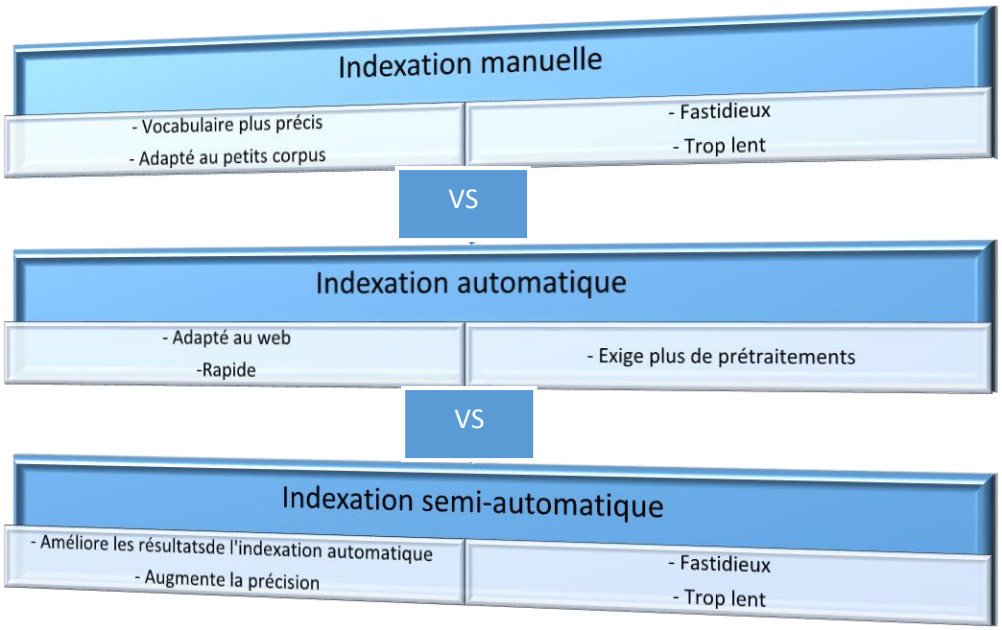

Figure 2-2: résume les trois types d'indexation.

La génération d'un langage d'indexation est une opération manuelle, automatique ou semi-automatique. La figure 2.2 explique en détail ce qui suit :

- Dans l'indexation manuelle, une équipe d'experts désignent préalablement les termes à indexer relatives à chaque document.

- Dans l'indexation semi-automatique (ou supervisée), un algorithme identifie, pour chaque document de la collection, des descripteurs qui sont proposés à l'équipe des experts. Ces derniers peuvent valider, supprimer ou, parfois, ajouter des entrées dans l'index.
- Finalement, l'indexation automatique qui est la plus fréquemment utilisée. Appelée aussi « non-supervisée » du fait que ce type d'indexation fonctionne sans intervention humaine. Ce qui rend le traitement plus rapide.

2.2.3. Critères d'évaluation d'une indexation

Les résultats des systèmes de recherche d'information dépondent de la qualité de leur indexes. Cette dernière est évaluée sur la base de deux critères : l'adaptation et la cohérence entre représentations des requêtes et représentation des documents.

a) **La cohérence :** Il est clair que l'index doit être cohérent, c'est-à-dire que si deux textes considèrent le même sujet, et utilisent deux différents vocabulaires, seront indexés avec les mêmes mots-clés. Mais généralement deux personnes différentes ont moins de 20% de chance de choisir instinctivement le même mot pour décrire un concept. De ce fait, il est difficilement possible d'obtenir un index cohérent d'une grande collection par l'indexation humaine, sans un contrôle minimal. Autrement, l'indexation automatique n'est pas non plus cohérente, et ne sont pas capables de prendre en compte les ambiguïtés des termes comme la synonymie, l'homonymie ou la polysémie.

b) **La concordance entre représentations :** il s'agit de vérifier le critère d'adéquation entre la représentation de la requête et le format de l'index du corpus. Si nous voudrions retrouver un besoin exprimé sous forme de requête, il faut que ses descripteurs appartiennent au même vocabulaire utilisé pour décrire la collection.

C'est donc plus facile de vérifier le critère de cohérence et d'adéquation si en indexe une collection à l'aide d'un langage contrôlé. Mais ce dernier ne prend pas en compte l'évolution du vocabulaire dans le temps. Ce qui représente un point faible pour les systèmes de recherche d'information qui index automatiquement une collection de document, car le vocabulaire de la collection à de grande chance d'être moins actuel que celui de l'utilisateur du système. Il est d'ailleurs conseillé à l'utilisateur de SRI de se conformer au vocabulaire des documents plutôt qu'au sien, quand il compose sa requête. Heureusement, les méthodes de retour de pertinence permettent à l'utilisateur d'effectuer en partie cette tâche (FURNAS, et al., 1987).

2.3 Systèmes de Recherche d'Information SRI

On remarque bien que l'indexation et la recherche d'Information sont deux domaines d'application très liés. Les Systèmes de Recherche d'Information font la liaison entre ces deux domaines.

Tous système de recherche d'information (SRI) implémente les tâches de RI expliqués dans la figure 2.3. Son objectif principal est de retourner un ensemble de documents, qui répondent au besoin en information d'un utilisateur, exprimé par une requête qu'il a soumise. Afin de réaliser cela, un SRI utilise un processus en "U" pour identifier les documents pertinents, depuis une collection de documents, en réponse à un besoin spécifique.

Comme l'illustre la figure 2.3, le processus de recherche d'information est composé de trois fonctions principales :

(1) Indexation des documents de la collection et de la requête utilisateur ;

(2) l'appariement (Matching) de requête-documents, qui permet de comparer la requête et chaque document de la collection ;

(3) Reformulation de la requête ou expansion de requête, qui est la fonction de modification. L'expansion intervient en réponse aux mauvais résultats. Elle a comme but l'amélioration de la qualité de la requête.

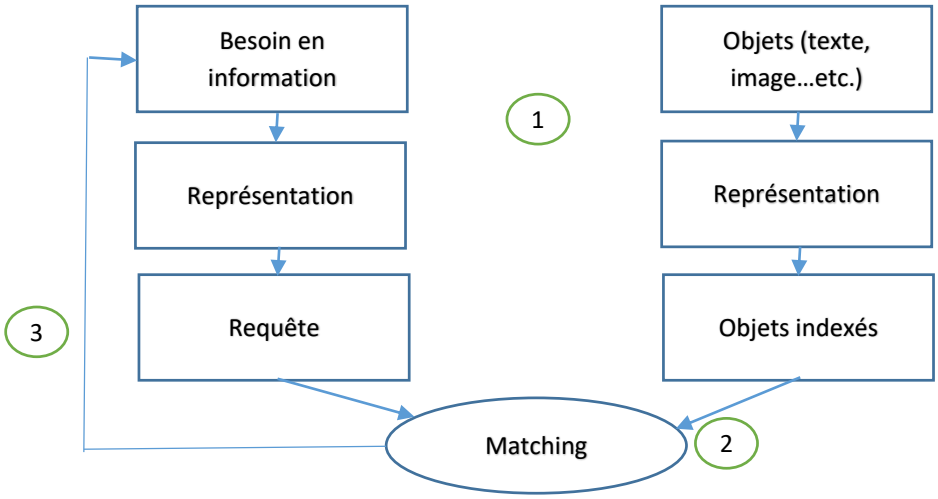

Figure 2-3: processus de Recherche d'information

Dans ce qui suit, nous introduisons dans un premier temps les éléments de base de la mise en œuvre du système de RI, à savoir l'indexation et l'interrogation.

2.4 Les modèles de recherche

Le modèle de recherche d'information est responsable de la représentation des documents, des requêtes d'un utilisateur, et surtout de la fonction d'appariement documents/requêtes. Les modèles de recherche d'information fournissent un outil d'interprétation de la notion de pertinence vis-à-vis le besoin en information. Les modèles de RI peuvent être classés en trois catégories principales, à savoir : les modèles booléens, les modèles vectoriels et les modèles probabilistes. La figure 2.4 présente une classification des différents modèles de RI, proposée dans (Baeza-Yates, 2011). Nous distinguons les modèles suivants :

Les modèles booléens dans (Salton, 1969) modélisent l'appariement document-requête nous nous basant sur la théorie des ensembles et l'algèbre de Boole. Leur principe est relativement simple : (1) des requêtes formulées par des expressions booléennes, (2) des poids binaires (présence/absence) ; et (3) une notion de pertinence. Il existe trois variations principales : le modèle booléen classique, le modèle booléen étendu et le modèle booléen flou.

La deuxième classe introduite par Salton et autres, est celle des modèles vectoriels (Salton & Yang, 1975) qui reposent sur une représentation vectorielle des documents et des requêtes. Cette classe résout quelques limites du modèle booléen en proposant, à l'instar des poids binaires de chaque terme pour les documents et les requêtes, un poids positif plus significatif. Les modèles vectoriels comprennent le modèle vectoriel généralisé, le modèle connexionniste et le modèle LSI (Latent Semantic Indexing).

La dernière classe est la classe des modèles probabilistes adoptés par (Maron, 1960), (Robertson, 1988) et (Salton, 1986) pour mieux modéliser le degré de pertinence des résultats par rapport à la requête utilisateur. Cette catégorie englobe les modèles de langue, le modèle probabiliste général et le modèle de réseau inférentiel.

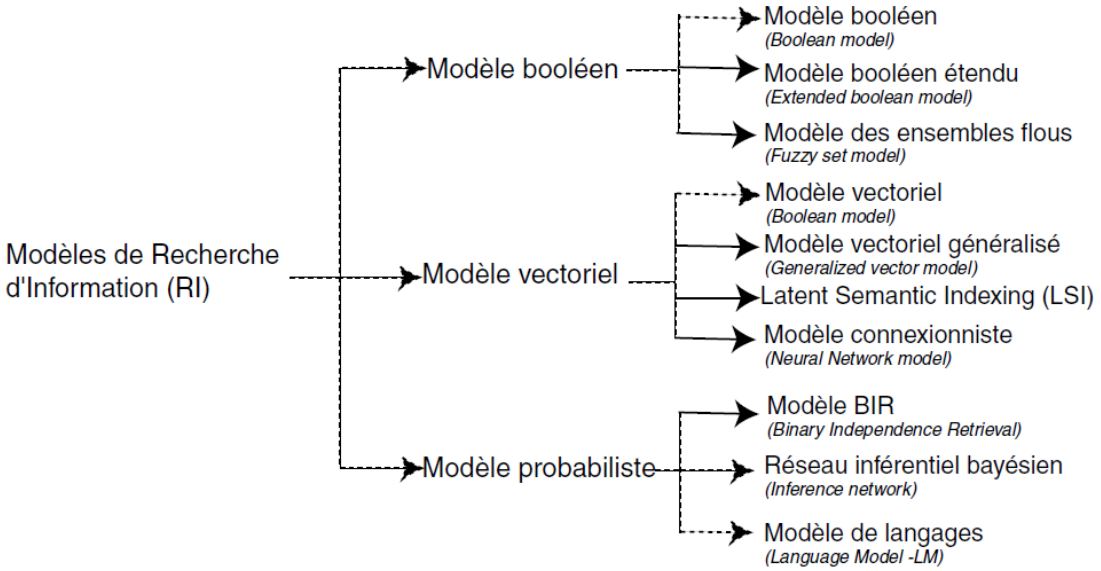

Figure 2-4 : Taxonomie des modèles en RI (Baeza-Yates, 2011).

Contrairement à l'indexation qui choisit les termes pour exprimer le contenu d'un document ou d'une requête, le modèle donne une interprétation des termes choisis. Le modèle remplit deux fonctions quel que soit l'ensemble de termes pondérés issus de l'indexation : La première fonction est de permettre une représentation interne pour le document et pour la requête basée sur les termes dedans l'index, la seconde fonction est de fournir une fonction de comparaison entre la représentation des documents et la représentation des requêtes afin de déterminer leur degré de similarité (ou ressemblance). Ainsi, le modèle de recherche d'information détermine le comportement clé d'un SRI (Système de Recherche d'Information).

De nombreux modèles de RI existent. Nous présenterons en premier lieu le modèle booléen qui est historiquement un des premiers modèles et qui a servi de base aux recherches du domaine. Ensuite, nous décrirons le modèle vectoriel (appelé aussi algébrique) qui est le plus utilisé. Et enfin, le modèle probabiliste qui est à la base de model de langue. Pour chacune des modèles précédents, deux modules importants seront illustrés : le module représentation et le module comparaison.

2.4.1. Modèle Booléen

Dans le modèle booléen, une recherche consiste à trouver tous les documents qui contiennent les mêmes termes que la requête construite à base de mots clés. Dans ce sens, un document se rapportant à un mot-clé « professeurs » ne sera pas récupéré pour répondre à une requête qui concerne le mot-clé « chercheurs » lorsque le document ne contient pas le terme « chercheurs », même s'il est évident que « professeurs » est un type de « chercheurs ».

Dans le modèle booléen, les requêtes peuvent être formulées à l'aide des termes reliés par les opérateurs logiques comme « AND », « OR » et « NOT ». Comme indiqué dans les sections précédentes, le document est représenté par un vecteur index, c'est-à-dire : $d = t_1, t_2, \dots t_n$. La requête est par contre représentée par une expression logique de termes avec des opérateurs logiques.

L'appariement (Matching) dans le modèle booléen entre une requête et les documents un par un est un appariement exact. Autrement dit, si un document de la collection implique au sens logique la requête, alors le document est pertinent pour la requête. Sinon, il est considéré non pertinent. La correspondance entre document et requête est déterminée par la fonction M comme suit :

$$M(d,q) = \begin{cases} 1 \ si \ d \ appartient \ à \ l'nsembledécrit \ par \ q \\ 0 \ sinon \end{cases} \quad (3)$$

Le modèle probabiliste présente un certain nombre de faiblesses :

- Le manque d'ordre (pas de Ranking) des documents retournés selon leur pertinence.

- La représentation binaire du modèle est peu informative, car elle ne renseigne ni sur la longueur de document ni sur la fréquence du terme dedans, importantes pour la RI.
- La difficulté de formuler de bonnes requêtes implique que l'ensemble des documents retournés est souvent grand, pour la plupart des requêtes courtes, ou complètement vide dans le cas de requêtes longues.
- Ce modèle ne supporte pas la réutilisation des résultats initiales d'une recherche pour les améliorés.
- Les systèmes booléens présentent une efficacité de recherche inférieure aux autres systèmes.

Pour cela, plusieurs améliorations ont été apportés afin de remédier à certains problèmes de ce modèle, parmi elles nous trouverons : le modèle booléen basé sur la théorie des ensembles flous (Zadrozny & Kacprzyk, 2005), le modèle booléen étendu (Fox, et al., 1992). Ainsi, Les premiers SRI développés étaient basés sur le modèle booléen, même aujourd'hui beaucoup de systèmes de recherche d'information commerciaux utilisent le modèle booléen. Cela est dû à la simplicité et à la rapidité de sa mise en œuvre.

2.4.2. Modèle Vectoriel

Le modèle vectoriel est un modèle mathématique proposé par Salton base sur une formalisation géométrique dans le cadre du système SMART (Salton, 1971). Par défaut, les documents ainsi que les requêtes sont représentés dans un espace géométrique, dite vectoriel, à plusieurs dimensions, ou chaque dimension représente une pondération d'un terme d'indexation. Ce qui veut dire que chaque document et chaque requête est représenté par un vecteur de « n » dimensions (n : nombre de termes). Formellement, si nous supposons un espace de termes d'indexation $T= \{t_1, t_2, \ldots, t_n\}$ de dimension n, un vecteur document d_i $(w_{i1}, w_{i2}, \ldots, w_{ij}, w_{in})$ et une requête q $(w_{q1}, w_{q2}, w_{qj}, w_{qn})$ de même dimension. Le modèle vectoriel propose des fonctions de pondération de terme dans le document ; w_{ij} (resp. w_{qj}) qui représente le poids du terme t_j dans le document d_i (respectivement dans la requête q). Dans la littérature, la plus part des fonctions de pondération proposés prennent en compte une pondération locale et une pondération globale (Ronan & Colm, 2006).

La pondération locale permet de mesurer l'importance du terme dans le document. En général, elle correspond à une fonction de fréquence d'occurrence des termes dans le document (noté tf abréviation de term frequency), exprimée comme suite :

$$tf = 1 + (f(t_i, d_j)) \quad (4)$$

Où $f(t_i, d_j)$ représente la fréquence du terme t_i dans le document d_j.

Puisque les termes qui surgissent dans plusieurs documents de la collection ne permettent pas la distinction des documents pertinents des documents non pertinents, un facteur *idf* (inverted document frequency) de pondération globale est alors introduit. La pondération globale représente le poids d'un terme dans la collection. Ce poids doit être plus important quand le terme apparaît moins fréquemment dans la collection, et doit dépondre d'une manière inverse à la fréquence du terme en document. *Idf s'*exprime comme suit :

$$idf = log\left(\frac{N}{n_i}\right) \tag{5}$$

n_i : la fréquence en document du terme considéré,

N : le nombre total de documents dans la collection.

Les deux fonctions (4) et (5) sont référencées sous le nom de la mesure *tfidf* qui donne une très bonne approximation de l'importance d'un terme dans une collection de documents de taille homogène.

$$tfidf = tf.idf \tag{6}$$

Cependant, la taille du document est ignorée dans la mesure *tfidf* ainsi définie dans (6). En effet, la fonction *tfidf* favorise les documents longs, car ils ont tendance à contenir plus de répétitions d'un même terme. En conséquent, *tfidf* augmentent la similarité des documents longs par rapport à la requête. Pour résoudre ce problème, des travaux ont proposé d'intégrer la taille du document dans les formules de pondération, comme facteur de normalisation (Robertson & Walker, 1997) (Singhal, et al., 1996).

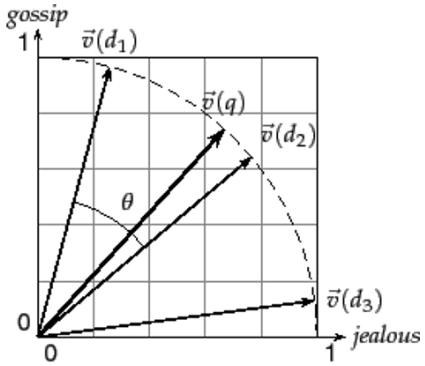

Figure 2-5 : similarité cosinus (Christopher, et al., 2008)

L'appariement (matching) dans le modèle vectoriel consiste dans ce cas à trouver les vecteurs documents qui s'approchent le plus de vecteur de la requête. Dans ce cas plus proche veut dire distance entre les deux vecteurs la plus petite. Van Rijsbergen défini plusieurs mesures de similarité, dont les plus courantes sont le Produit scalaire, Coefficient de Dice, Mesure de Jaccard, Mesure de recouvrement et Cosinus illustrées dans la figure 2-5 (Van Rijsbergen, 1979).

Parmi les avantages du modèle vectoriel, nous trouverons sa prise en compte du poids des termes dans les documents, ce qui permet de retrouver des documents qui répondent partiellement à une requête. En plus, il offre la possibilité de classer (Rank) les résultats

d'une recherche basée sur la similarité potentielle entre documents et requête. Autrement, l'inconvénient majeur de modèle vectoriel est principalement qu'il repose sur le fait qu'il suppose l'indépendance des termes d'indexation, tandis que ces termes dans les documents sont souvent sémantiquement liés.

Dans la littérature, plusieurs variantes du modèle vectoriel ont été adoptées, pour remédier aux limitations précédemment cités. Parmi elles, nous citons particulièrement le modèle vectoriel généralisé (Wong, et al., 1985), Latent Semantic Indexing (Foltz, 1990), (Dumais, 1994), (Metzler & Croft, 2007), (Lang, et al., 2010) et(Fahsi & Benslimane, 2014b), et le modèle connexionniste (Kwok, 1989).

2.4.3. Modèle Probabiliste

Les premiers modèles probabilistes ont été basés sur les travaux de (Robertson, 1977) (Bookstein, 1983) et (Fuhr, 1989) développés sur la base de la théorie des probabilités. Ainsi, elle considère que les termes d'indexation sont indépendants, et que la probabilité d'apparition de chaque terme est la même avec ou sans la présence des autres termes. Sous la base de cette hypothèse, le problème de recherche d'information revient donc à estimer la probabilité qu'un document soit pertinent par rapport à la requête.

La probabilité de pertinence *P(R)* désigne la probabilité d'apparition d'un évènement est formalisée à travers l'expérimentation qui est le procédé par lequel l'observation est faite. L'espace de départ {pertinent, non-pertinent} est l'ensemble des valeurs que peut prendre un fait. Dans le modèle probabiliste les termes d'indexation sont considérés indépendants, c'est-à-dire que la probabilité d'apparition d'un terme est la même avec ou sans la présence des autres. A partir de ces hypothèses, le SRI cherche à estimer la probabilité qu'un document soit pertinent pour une requête. Ce qui revient à estimer la probabilité de la pertinence notée *P(PERT/D)* et non pertinence *P (NPERT/D)*. Seules l'existence et le manque de termes dans les documents et dans les requêtes sont considérées comme des évènements observables, 0 (absent) ou 1 (présent). L'ensemble de documents pertinents est appelé *PERT* et l'ensemble de documents non pertinents *NPERT*.

Dans le modèle probabiliste, la similarité d'une requête q et un document d est calculée par :

$$\text{Similarite}(d, q) = \frac{P(\text{Pert}/d, q)}{P(\text{NPert}/d, q)} \quad (7)$$

Ainsi, quand la probabilité d'un document est élevée, ce document est pertinent pour la requête. Pour le calcul de ces probabilités qui ne sont pas directement calculables en utilisant les règles de Bayes suivantes :

$$P(PERT|D,Q) = \frac{P(D,Q|PERT) * P(PERT)}{P(D,Q)} \quad (8)$$

$$P(NPERT|D,Q) = \frac{P(D,Q|NPERT) * P(NPERT)}{P(D,Q)} \quad (9)$$

$$O(D) = \frac{P(PERT|D,Q)}{P(NPERT|D,Q)} = \frac{P(D,Q|PERT) * P(PERT)}{P(D,Q|NPERT) * P(NPERT)} = Const * \frac{P(D|PERT,Q)}{P(D|NPERT,Q)} \quad (10)$$

Avec :

- *P (PERT)* la probabilité qu'un document choisi au hasard soit pertinent (*P(PERT)* est une constante qui dépend de la requête).
- *P (D/PERT, Q)* la probabilité d'observer *D* sachant que la présence de *Q*.
- *P (D, Q)* la probabilité appariée du couple *D, Q*.
- *O(D)* permet de classer les documents en fonction de leur estimation de pertinence.

L'état de l'art présenté par Blair et Maron suggère que *P(PERT)* pourrait être défini par les statistiques sur l'usage du document. C'est-à-dire par proportion du nombre d'utilisations du document courant sur le nombre d'utilisations total. Ce qui induit à un classement par popularité, très à la mode sur internet (Blair & Maron, 1990).

Selon (SAVOY, 2006), Le modèle de recherche probabiliste et le model vectoriel sont presque équivalents, avec plus efficacité (précision) pour le modèle de recherche vectoriel, et plus de performance (rappel et temps d'exécution) pour le modèle de recherche probabiliste.

2.5 Évaluation des SRI

L'évaluation des modèles et des méthodes de recherche d'information a toujours été un enjeu très important depuis la naissance du domaine de la RI (Lancaster, 1979), Pour cela deux classe de mesures de qualité des systèmes de recherche ainsi que des ensembles de données ont été développés afin de tester ces systèmes sur une base commune. La première classe est dite « mesures système », vise à évaluer les performances du système en termes de qualité des documents retournés par le système, c'est-à-dire leur pertinence vis-à-vis le besoin des utilisateurs. L'autre est dite « mesures usager », centré sur le critère satisfaction de l'utilisateur, et non sur les performances système. la deuxième classe s'intéresse à la modélisation du comportement des utilisateurs en situation de recherche.

Dans la littérature, les mesures système sont les mesures les plus utilisées dans le domaine de la RI. Ces mesures se basent sur deux éléments principaux à savoir : les collections de test et des mesures d'évaluation. Ainsi, pour réaliser une évaluation, une expérimentation qui utilise les éléments suivants doit être établie :

- Une collection de test composée de :
    1. Un ensemble de documents
    2. Un ensemble de requêtes.
    3. Une évaluation préétablie de la pertinence des documents pour chaque requête.
- Des mesures et des critères quantifiables.

2.5.1. Les corpus/collections de test

Un corpus (ou Une collection) de test est généralement composée d'un ensemble de documents, d'un ensemble de requêtes et d'un fichier jugements de pertinence. L'évaluation d'un SRI à l'aide d'un corpus consiste à comparer les résultats retournés par l'approche implémentée avec ceux des jugements de pertinence.

Les premières collections de test comme LISA, NLP, CACM, CISI, Cranfield, Time, Medline, ADI[4] étaient réalisées par de petits groupes de recherche et ne contenaient pas plus de 12000 documents. Tandis que la collections CLEF[5] et la campagne TREC[6] constituent à ce jour les collections de référence dans le domaine d'évaluation des systèmes de recherche d'information (Voorhees & Harman, 2005). Leurs objectifs est de proposer des collections de test spécifiques à des tâches comme la recherche d'information sur le web, recherche d'information médical, recherche d'information dans les micros blogs, recherche d'information contextuelle, etc. en plus, ils proposent des protocoles d'évaluation spécifiques à chaque domaine (Kanoulas, 2016). Les documents CLEF et TREC sont codés à l'aide de SGML dans un format spécifique TREC. Leurs nombres au sein d'une collection varient entre des milliers à des millions de documents avec une taille entre 2Go à 25 To.

2.5.2. Les notions de bases

Dans une collection de test (corpus), les documents forment quatre partitions selon deux caractéristiques :

1. Les documents retournés et les documents non retournés.
2. Les documents pertinents et les documents non pertinents.

[4] http://ir.dcs.gla.ac.uk/resources/test_collections/
[5] www.clef-campaign.org
[6] http://trec.nist.gov/

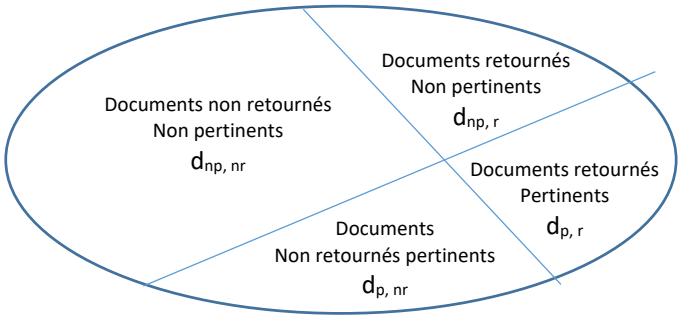

Figure 2-6 : Représentation des partitions de la collection de test

Il existe à cet effet de nombreuses mesures (Sanderson, 2010), parmi lesquelles, nous avons retenu les suivantes :

- La Précision/Rappel à n documents restitués,
- La F mesure,
- La Précision exacte,
- La Précision interpolée,
- La Précision moyenne MAP.

2.5.3. Rappel/ précision

Ces deux mesures sont calculées à partir des sous-groupes de documents (pertinents, non-pertinents) de la figure 2.6 par :

3. La Précision : c'est la capacité du système à rejeter les documents non pertinents obtenus par le rapport entre l'ensemble de documents sélectionnés pertinents et l'ensemble de tous les documents sélectionnés.

$$Précision = \frac{\#documents\ sélectionnés\ pertinents}{\#tous\ les\ documents\ sélectionnés} \quad (11)$$

4. Le Rappel : c'est la capacité du système à trouver les documents pertinents de tous les documents. Autrement dit, c'est le rapport entre le nombre de documents pertinents sélectionnés et le nombre de tous les documents pertinents dans le corpus.

$$Rappel = \frac{\#documents\ pertinents\ sélectionnés}{\#tous\ les\ documents\ sélectionnés} \quad (12)$$

Un système idéal devrait retourner tous les documents pertinents, c'est à dire la précision et le rappel de ce système est égal à 100%. Mais c'est impossible car le facteur précision et le facteur rappel sont antagonistes. En effet, si la précision augmente, le rappel diminue et vice versa.

5. Précision/Rappel à n documents : appelée aussi précision exacte, signifie que si la requête admet n documents pertinents alors la précision exacte est calculée sur les n premiers documents résultants du calcul.

$$Précision_n = \frac{\#documents\ pertinents\ sélectionnés}{n} \tag{13}$$

La précision/rappel moyenne : c'est la moyenne des valeurs de précisions de chaque document pertinent (sélectionné ou non) dont la précision de document pertinent non sélectionné est nulle.

$$Précision_n = \frac{\sum_{i=1}^{m} précision\ de\ documents}{\#Précision(\#documents\ pertinents)} \tag{14}$$

À côté des mesures précédentes, nous trouverons d'autres métriques permettant de définir plus la performance du système tel que le temps de réponse du système, la souplesse et faciliter d'utilisation, le nombre total des documents retournés (couverture), le rang du premier document et la longueur de recherche.

2.6 Conclusion

Nous avons passé en revue les principaux concepts et modèles de la RI dans ce chapitre. Nous avons introduit des notions de base comme le besoin en information exprimé dans la requête utilisateur, le document, la collection de test et la notion de pertinence. Par la suite, nous avons présenté aussi les étapes du processus de base de la recherche d'information, tel que : l'indexation, le matching requête-document et le poste traitement des résultats tel que l'expansion et la reformulation de requêtes. Ensuite, nous avons étudié les différents modèles de la RI. Ces modèles bien que simples n'emploient aucun mécanisme de sécurité pour assurer la confidentialité et ne permettent pas de sécuriser la communication entre les serveurs des systèmes de recherche d'information. En plus, peut de modèles adaptent leurs implémentations aux nouvelles technologies tel que les Cloud de données. Enfin, les mesures et les collections de tests dédiées à l'évaluation des systèmes de recherche d'information ont été traitées.

Le chapitre (2) sera consacré à la description du Cloud considéré comme la nouvelle dimension de stockage et traitement de données. Le trait sera fait autour de l'historique de son implémentation, les différentes architectures ainsi que les avantages et inconvénients de chacune.

Chapitre III.

Le Cloud :
Concepts de base et Sécurité

Chapitre III. Le Cloud : Concepts de base et Sécurité

3.1 Introduction et définitions

Le Cloud Computing est une technologie similaire à Internet et qui est utilisée quotidiennement depuis longtemps sans le savoir, pour stocker et traiter nos boites de messagerie électronique, nos fichiers distants, nos données des réseaux sociaux, etc. Bien qu'assez similaire à l'Internet dans leurs fonctionnements, leurs services sont assez différents.

Premièrement, nous pouvions utiliser Internet sans avoir besoin du Cloud Computing, mais nous ne pouvions pas utiliser les Cloud Computing sans utiliser Internet car c'est le moyen le plus utilisé par les fournisseurs de solution de Cloud Computing pour proposer leur service.

Deuxièmement, le Cloud Computing fournis des services d'hébergement et d'exploitation des applications et des données utilisateur qui seront stockés et traités non plus sur l'ordinateur local de ce dernier, mais sur les serveurs distants du cloud, accessibles par le biais d'une excellente bande passante, indispensable à la fluidité du système.

En partant de ce principe, Le NIST[7] propose la définition suivante du Cloud Computing:

« Le Cloud Computing est un modèle qui permet d'offrir, à la demande, un accès réseau commode à un ensemble de ressources informatiques configurables partagées (par exemple des réseaux, des serveurs, des systèmes de stockage, des applications et des services) qui peuvent d'être rapidement mises à disposition et libérées avec un effort de gestion ou une intervention du fournisseur de services minimale. Ce modèle de Cloud met en avant la disponibilité et présente cinq caractéristiques essentielles, trois modèles de services et quatre modèles de déploiement» (Mell & Grance, 2009).

Ainsi définit, la technologie du Cloud Computing n'a pas une date d'innovation précise, mais la notion de consommation de services informatiques a été proposée dans les années soixante par John McCarthy, l'informaticien brillant et l'inventeur entre autres du langage LISP. Le but était de rendre disponible la puissance de calcul des superordinateurs et leurs capacités de stockage à tout le monde, à tout moment, n'importe où via une très grande bande passante fournis par Internet. Cependant, ce n'est qu'en 2006 qu'Amazon comprit qu'un nouveau mode de consommation de l'informatique et d'Internet faisait son apparition (Papadimitriou, et al., 2008).

Mais après l'adoption de la technologie du Cloud Computing, Trois concepts fondamentaux ont apparu :

[7] Institut National des Standards et de la Technologie

1. La virtualisation :

Consiste à faire fonctionner un ou plusieurs systèmes d'exploitation sur un ou plusieurs ordinateurs. Cela à de nombreux avantages tels que l'utilisation optimale des ressources d'un parc de machines, l'économie sur le matériel ainsi que l'installation, tests, développements sans endommager le système hôte.

2. Datacenter :

Un Datacenter (centre de traitement de données) est un site dans lequel se trouvent les équipements qui constituent le système d'information de l'entreprise (serveurs, supports de stockage, équipements et câbles réseaux, etc.).

Le Datacenter peut être interne à l'entreprise comme il peut être externe à l'entreprise. Il est exploité par les utilisateurs de service avec ou sans le soutien de prestataires de service. Cela dépend si cette infrastructure est propre à une entreprise et utilisée par elle seule ou à des fins commerciales. Dans ce cas, des particuliers et/ou des entreprises peuvent le louer pour stockage ou traitement.

3. Plateforme collaborative :

La plateforme collaborative est un ensemble d'outils mis à la disposition des utilisateurs du service Cloud pour faciliter et optimiser la communication entre les acteurs dans le cadre d'un projet de travail ou d'une tâche au sein de ce dernier. Une plateforme collaboratives incorporent généralement les outils suivant : Un service de messagerie, Des outils de type forum ou pages de discussions, Un système de partage de ressources et de fichiers, Un annuaire des profils des utilisateurs, Un calendrier, etc.

3.2 Modèles de services du Cloud Computing

Selon NIST, les fournisseurs du Cloud proposent trois modèles principaux selon les types de fonctionnalités et les besoins des entreprises :

- IaaS : Infrastructure as a service
- Pass : Platform as a service
- SaaS : Software as a service

Cette classification par modèles est due à la couverture de chaque modèle résumé dans la figure 3-1.

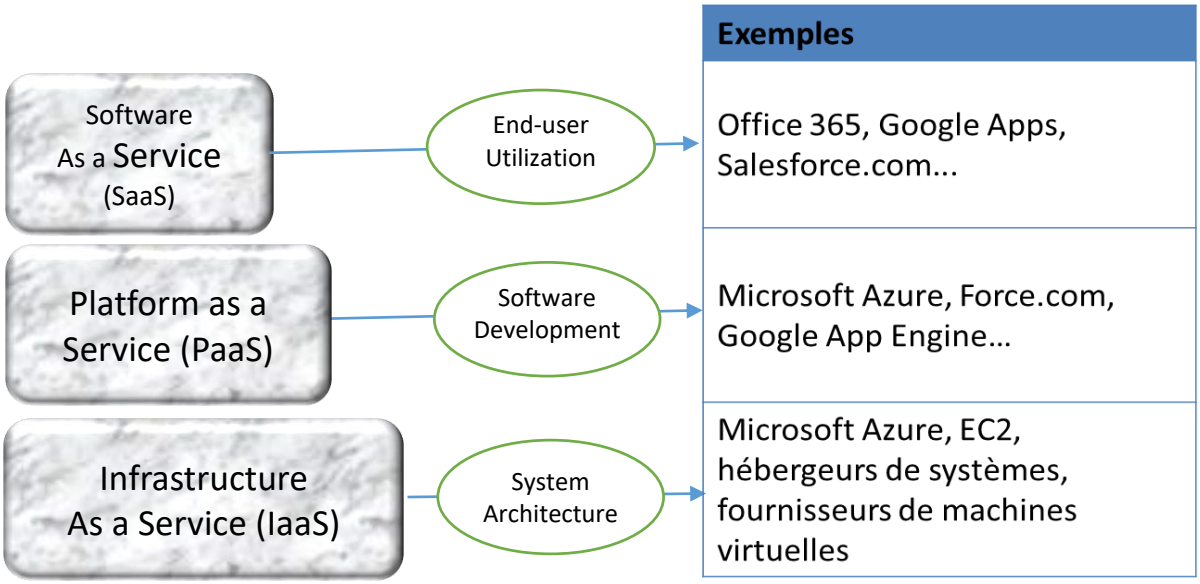

Figure 3-1 : Répartition des charges de l'utilisateur en fonction du modèle de Cloud

Une organisation en modèles permet de séparer les domaines de compétences, ainsi l'utilisation d'une couche est complètement non soumis aux couches inférieures ou supérieures.

3.2.1. PaaS: Platform as a service

PaaS ou plateforme en tant que service est un modèle où le client a la capacité de déployer dans l'infrastructure du Cloud ses applications développées à l'aide des langages de programmation et des outils pris en charge par le fournisseur ou achetées depuis d'autres fournisseurs. Dans ce cas, le client ne gère et ne contrôle pas l'infrastructure de Cloud sous-jacente (administration réseau, serveurs, systèmes d'exploitation et stockage), mais contrôle les applications déployées et l'environnement qui les héberge.

Cette plateforme offre un environnement de développement géré, hébergé et entretenu par un fournisseur Cloud, basé sur une infrastructure matérielle externe à l'entreprise. Il aura donc la possibilité de développer des applications web uniques pour son activité à l'aide d'une interconnexion collective de plusieurs intervenants internes ou externes tels que :

- Les plateformes de développement
- Les outils de gestion des bases de données
- Les outils de gestion de la sécurité, de la capacité

Plusieurs solutions complètes destinées aux développeurs sont disponibles via Internet. Les plus importants sont :

- Microsoft avec Windows AZURE
- Google avec Google App Engine
- Orange Business Services.

3.2.2. SaaS: Software as a service

SaaS ou application en tant que service est le premier modèle des coud à apparaitre. Il est aussi le modèle le plus utilisé parmi les modèles de Cloud du marché (Armbrust, et al., 2010). Ce modèle déploie et offre des applications sur une infrastructure gérée par le fournisseur. Dans ce cas, le fournisseur loue ces applications clé en main à ses clients en tant que service à la demande au lieu de leur facturer la licence du logiciel. Ainsi, l'utilisateur final n'a pas besoin d'installer les logiciels, de les maintenir, ou de les mettre à jour.

Puisque les applications sont clés en main, le client n'a pas à gérer ou contrôler l'infrastructure SaaS. Il ne peut pas aussi déployer ses propres applications, mais il peut seulement configurer certains paramètres de ces applications. Le service SaaS est généralement un service de messagerie électronique, CRM (Customer Relationship Management), GED (Gestion électronique de documents), collaboration en ligne, logiciels de gestion de paie et ressources humaines, ou des services de sauvegardes en ligne.

Les offres SaaS les plus utilisées sont (Marinos & Briscoe, 2009):

- Google Apps (messagerie et bureautique)
- SalesForce offre CRM (Customer Relationship Management)
- Office 365 (messagerie, outils collaboratifs, bureautique)

3.2.3. IaaS : Infrastructure as a service

IaaS ou l'infrastructure en tant que service est un modèle où l'entreprise dispose d'une infrastructure informatique formée de serveurs de traitement et de stockage, et d'interconnexion réseau qui se trouve en réalité chez le fournisseur de service. Il est en mesure aussi de déployer et d'exécuter des logiciels quelconques, comme des systèmes d'exploitation et des applications. Ce qui permet à l'utilisateur du service IaaS d'éviter l'achat et de la gestion du matériel. L'entreprise exploite le matériel comme un service à distance. Ce service permet à l'entreprise de se focaliser sur ses processus métiers sans avoir à contrôler l'infrastructure matérielle du Cloud.

Dans ce cas, l'entreprise qui s'approprie le service IaaS dispose d'un contrôle sur les systèmes d'exploitation, les applications déployées le stockage, et l'ensemble des composants réseau sélectionnés (par exemple le pare-feu de l'hôte). Ce qui inclut des un espace serveur, un nombre de connexions réseau, une bande passante, des adresses IP et des load balancers. Ces ressources proviennent réellement de différents Datacenter, dont le fournisseur IaaS a la responsabilité d'entretenir. Les principales offres IaaS proposées sont :

- Le service Elastic Compute Cloud (EC2) de Amazon Web Services (AWS)
- Le service Azure de Microsoft

3.3 Déploiement des Cloud

L'utilisation de la solution Cloud Computing permet d'avoir un espace virtuel de mise en place des infrastructures serveur ou réseau, des plateformes de développement ou d'exécution, et tout cela peut être déployé sur trois types de Cloud Computing selon les besoins des entreprises ainsi que des utilisateurs. Chaque mise en place est appelée modèle de déploiement du Cloud Computing de la figure 3-2.

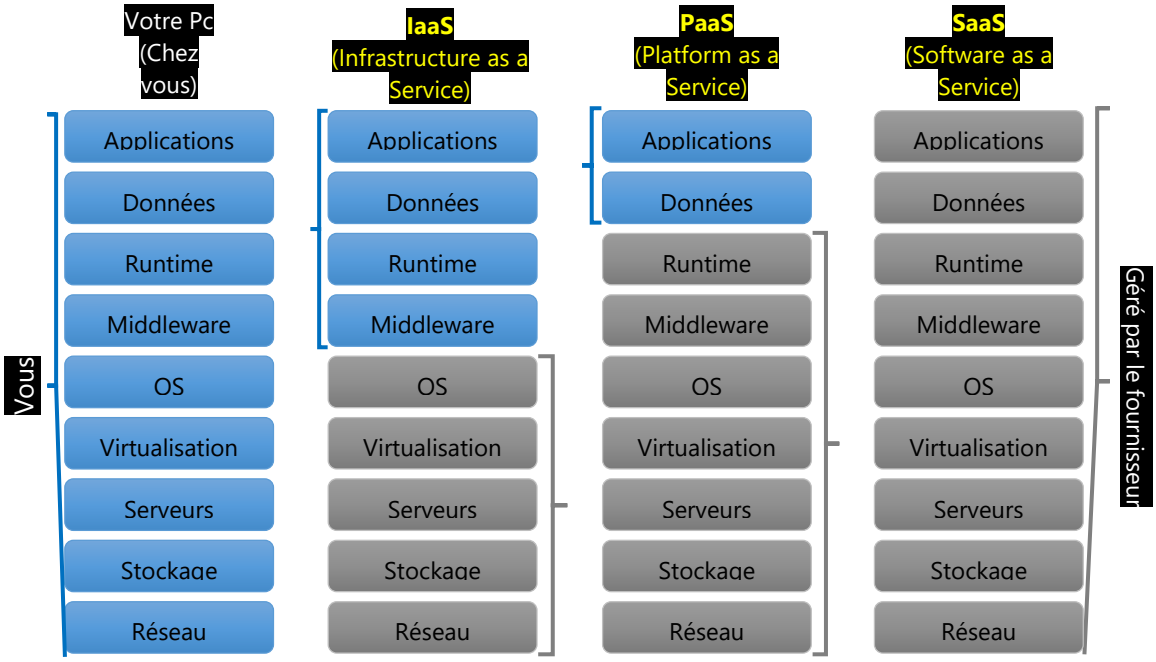

Figure 3-2 SaaS, PaaS et IaaS qui gère quoi ?

Une explication des trois modèles ci-dessous sera fournie dans les sections qui suivent. Mais il est important de dire que le client des services cloud obtiendra un meilleur contrôle sur la sécurité des ressources lorsqu'on passe de SaaS au PaaS et plus encore de PaaS à IaaS, de même que lorsqu'une entreprise passe d'un Cloud public ou d'un Cloud communautaire à un Cloud privé comme exprime NIST[8][9].

3.3.1. Cloud Privés

Dans le cas du cloud privé, l'infrastructure matérielle est réservée à une entreprise. Donc, seuls l'entreprise et ses employés ou un tiers peuvent le gérer. L'infrastructure dans ce cas se trouve dans les locaux de l'entreprise ou ailleurs.

Le Cloud privé est caractérisé par la délimitation de l'utilisation à une seule organisation ainsi qu'un degré plus élevé de sécurité du réseau. Les ressources de fonctionnement proviennent d'un ensemble distinct de serveurs physiques interne ou externe à l'entreprise,

[8] ISO/IEC 17788:2014/ ITU-T Y.3500
[9] ISO/IEC 17789:2014/ ITU-T Y.3502

et accessibles par des connexions réseaux privées ou des connexions réseaux publics sécurisées. Les mécanismes de sécurité supplémentaires sont fournis par le modèle de déploiement du Cloud privé sont idéales pour tout type d'entreprise ayant besoin de stocker et traiter les taches et données privées sensibles. Dans Ce type de déploiement, les serveurs qui hébergent les services sont localisés dans le même site que l'entreprise propriétaire accessible à l'aide d'un réseau fermé et sécurisé, administrés par la direction du service informatique. Ce qui fait que l'entreprise met en place ça propre politique de gestion de son Cloud.

Dans le Cloud privé, les ressources informatiques sont flexibles, et les processus ainsi que le personnel de l'entreprise propriétaire utilisent en fonction de leurs besoins quand ils en ont besoin.

3.3.2. Cloud Communautaire

Ce mode de déploiement est un modèle dédié à une communauté de professionnels qui leurs permet de travailler en collaboration sur un même projet. L'exemple le plus adéquat est celui d'un Cloud gouvernemental dédié aux institutions étatiques.

Le Cloud communautaire à une infrastructure partagée par plusieurs entreprises mais destinée à une communauté précise avec des préoccupations communes des missions communes ou des exigences de sécurité communes.

Pour ce qui concerne l'infrastructure, le Cloud communautaire peut être géré par des entreprises ou un tiers et peut se trouver dans leurs locaux ou ailleurs.

3.3.3. Cloud Publics

Le principe du Cloud public est l'inverse du Cloud privé, car il fournit des services à des clients multiples en utilisant la même infrastructure partagée. Pour le déploiement des Cloud publiques, l'infrastructure est disponible au grand public ou à un groupe d'entreprises mais elle appartient à une seule entreprise qui vend des services en nuage.

Le Cloud publique est le modèle le plus connu par les utilisateurs des Cloud. Comme le cloud privé, les services du cloud public sont offerts à travers un environnement virtualisé. Sauf que cette fois, il est construit en utilisant à l'aide de ressources physiques partagées et accessibles via Internet (non sécurisé).

Les offres SaaS de stockage Cloud ainsi que les applications office, sont les offres Cloud public les plus réputées, mais il existe des offres IaaS et PaaS comme l'hébergement web et les environnements de développement comme service Cloud, correspond également à ce modèle (bien que les meilleures offres existent au sein de Clouds privés).

Ainsi définis, les Clouds publics accessibles via des offres destinées aux grand public, non gourment en infrastructure et en sécurité comme les Clouds privés. Mais les entreprises aussi peuvent avoir recours au Cloud public pour le stockage de contenus non-sensibles ou autres.

3.3.4. Les Clouds hybrides

Dans le cas d'un cloud hybride, L'infrastructure est constituée de deux Clouds ou plus (privés, communautaires ou publics) qui restent indépendants à l'intérieur mais reliés par une technologie standardisée ou propriétaires afin d'autoriser la portabilité des données et des applications. Exemple : « Cloud bursting » qui gère la répartition de charge entre les différents Clouds.

Le modèle hybride permet la cohabitation de deux ou plusieurs Clouds privés, Cloud publics et des Clouds communautaires pour remplir différentes fonctions au sein de la même entreprise.

Une organisation peut maximiser son efficacité en utilisant des services d'un Cloud public pour ses opérations non-sensibles et utiliser des services d'un Cloud privé si elle en a besoin

La mise en œuvre des modèles du Cloud hybride peut être par plusieurs manières :

- Plusieurs fournisseurs de Cloud s'unissent afin de fournir des services intégrés en Cloud privé et public ;
- Plusieurs fournisseurs de Cloud individuels proposent un pack hybride complet ;
- Des organisations qui gèrent leur propre Cloud privé s'inscrivent à un service de Cloud public puis l'intègrent ensuite dans leur infrastructure.

3.4 Avantages et inconvénients du Cloud Computing

En fonction du type de mode de déploiement du Cloud (public, privé ou hybride), un nombre d'avantages, mais aussi des inconvénients seront offertes.

Type	Avantages	Inconvénients
Cloud public	- Aucun investissement préalable - Aucun pré requis demandé - Un service d'une grande flexibilité - Un service d'une grande disponibilité - Un payement sur mesure	- Budget assez grand - Le cadre légal - La pérennité du service - Confidentialité et sécurité des données
Cloud privé	- Sécurité et confidentialité - Une architecture sur mesure	- Budget très grand

Tableau 3-1 : Avantages et inconvénients des modèles de déploiements.

Même si le Cloud public propose des ressources informatiques hébergées distantes et mutualisées, les offres de Cloud privées se distinguent par leur aspect dédié, car leur usage est réservé à une seule entreprise dans le but de répondre à un besoin personnalisé. Dans ce cas, toute la charge financière et technique comme le cout des serveurs, des personnels qualifiés ou des logiciels sont supportés par l'entreprise.

Suite aux problèmes et inconvénients des Cloud public cités plus haut, de nombreuses entreprises se tournent vers le Cloud Computing privé bien sûr si l'entreprise support les charges résultantes.

3.5 Sécurité dans le Cloud Computing

La "sécurité" est le frein et la préoccupation principal des entreprises et utilisateurs qui s'engent à l'adoption des services Cloud spécialement le Cloud publics. La sécurité couvre la confidentialité, l'intégrité, l'authenticité et la disponibilité des informations. Ainsi, certaines questions légitimes reviennent sans cesse :

- Où seront stockées les données ?
- Sont-elles en sécurité ?
- Aurais-je un accès à n'importe quel moment ?
- Qui aura un accès à nos données ?

3.5.1. Risques Associées au Cloud Computing

L'inventaire des menaces (risques) permet de sécuriser d'utilisation cloud computing. Si le gain obtenu par l'utilisation du Cloud est indéniable (flexibilité, disponibilité, paiement sur mesure, etc.), l'emplacement des données et la délégation de certaines responsabilités induit de nouveaux risques comme (Grange, et al., 2010):

1) Risques spécifiques au cloud public :

- Perte de contrôle et/ou de gouvernance
- Conformité(s) et maintien de la conformité
- Décentralisation et localisation des données
- Perte et destruction de données

2) Risques récurrents pour le cloud privé et publique en même temps :

- Les problèmes de récupération des données
- Les problèmes des APIs et des interfaces de programmation
- La Malveillance dans l'utilisation
- L'Usurpation d'identité

La mise au point d'une solution pour ces menaces implique des problèmes de sécurité subjacents à la solution elle-même. Afin de tester l'efficacité de ces solutions, l'organisation Cloud Security Alliance CSA a mis en place des normes de sécurité dans le Cloud Computing and Storage (Wayne & Timothy, 2011). Cette dernière a instauré deux domaines de sécurité de Cloud :

- **Sécurité physique** : permet de mettre en place une panoplie de standards relatifs aux infrastructures physique et leur sécurité.
- **Sécurité logique** : s'intéresse aux mécanismes de protection des applications de virtualisation et de déploiement de Cloud.
- **Sécurité des données :** se concentre sur les mécanismes de sécurité des données.

Tout de même, la sécurité absolue ne peut pas exister, et donc le problème de sécurité restera un problème de confiance entre le fournisseur de service et le consommateur de service.

3.5.2. Solutions physique:

Avec la "dématérialisation" impliquée par l'utilisation des technologies Cloud Computing qui implique un hébergement multiple/réparti sur plusieurs Datacenters, le client ne connaît pas le lieu d'hébergement de son infrastructure virtuelle, ces applications ou ces données.

Pour son assurer, le client de service Cloud peut faire une visite de Datacenter. Mais cela est insuffisant, le fournisseur Cloud adopte alors un certain nombre de certifications et/ou de classifications reconnues. Exemple : la classification Tier[10].

Les garanties qui concernent l'aspect physique du fournisseur de service Cloud comporte les quatre (04) axes non exhaustifs suivants :

a) **Accès physique :** Une personne mal intentionnée qui possède une connaissance parfaite de l'implémentation physique du Centre de Calcule peut mettre le Cloud hors service, ce qui provoque une rupture de service.

b) **Traçabilité des accès physiques :** Le contrôle d'accès doit être fortement considérer pour assurer une sécurité physique accrue, soit pour les Cloud privés, publics ou hybrides. La traçabilité examine le va-et-vient du personnel (informaticiens, agents, etc.) susceptibles d'être une source de dysfonctionnements volontaires ou non.

c) **Redondance matérielle :** Cette mesure est importante pour garantir la disponibilité des données et les services du Cloud avec des performances idéales. La redondance est une réplique des configurations sur des répliques des équipements ce qui permet de se protéger si un problème survient à un équipement donné.

d) **Résilience :** C'est la mise en place d'une architecture de secours sur un site géographiquement éloigné, avec des équipements redondants pour éviter les répercutions d'un désastre d'origine naturelle ou humaine qui engendre la destruction de l'infrastructure du Cloud.

[10] http://www.uptimeinstitute.org/

3.5.3. Sécurité logique

Le Cloud Computing se base sur le déploiement de plusieurs services dans un environnement virtuel ou physique. D'où le besoin de mécanismes logique qui assure la conformité des services et les plateformes de déploiement du Cloud. Dans ce cadre, les sept (07) points non exhaustifs suivant sont expliqués :

a) **Sécurité des serveurs virtuels :** il s'agit de sécuriser les machines virtuelles (VM) spécialement lors des mises à jour de sécurité, ainsi réduire la surface d'attaque par l'isolation des flux réseaux, l'affectation de quotas d'usage des ressources par les VMs, etc.

b) **Colocation sécurisée :** il s'agit d'héberger les applications et les données de plusieurs utilisateurs (entreprise, organisations, etc.) dans la même infrastructure physique, tout en assurant la confidentialité, l'intégrité, etc. la sécurité exige une gestion des droits d'accès, des privilèges, des mots de passe, etc.

c) **Le chiffrement :** les algorithmes de chiffrement asymétriques sont plus intéressants dans le contexte de Cloud, puisque le propriétaire de l'information est en mesure de la déchiffrer avec la clé privée. La sécurité dans ce cas est relative à la taille de la clé, et sélective car nous pouvions choisir de ne chiffrer que ce qui le nécessite. Mais en cas de besoin de traitement tel que l'indexation, le calcule ou la sauvegarde, le chiffrement va perdre son utilité.

d) **Solutions de déploiement logiques :** l'application de plusieurs méthodes de déploiement de même service sur les VMs permet de fournir des ressources aux différents utilisateurs en toute sécurité. Le but est donc de simplifier les déploiements de solutions mutuelles et améliorer la confidentialité.

e) **Segmentation réseau :** cette solution permet de mettre fin aux risques classiques lié au serveur, ainsi que ceux relatives à la colocation dans les Cloud public. La segmentation est faisable à l'aide de différents VLAN reliant l'infrastructure physique du Cloud et l'infrastructure du client. Des équipements comme les pare-feu, proxy, etc. assurent le routage sécurisé entre les VLAN.

f) **Sécurité de l'outil d'administration :** une seule vulnérabilité dans les outils d'administration peut causer une rupture de service ou une perte des données. La solution préventive consiste à mettre en place des outils de contrôle d'accès aux l'interfaces comme les pare-feu ou les proxys en plus d'antivirus.

g) **Sécurité des accès logiques :** ce point est possible si l'on accompagne l'authentification par la journalisation des accès réussies ou échouées, le changement périodique des mots de passe, formation du personnel, etc.

3.6 Sécurité des données

Le propriétaire de service Cloud s'engage à assurer la confidentialité et l'intégrité des données de ses utilisateurs la perte, l'altération ou la destruction. Il empêche tout accès ou utilisation fraudeuse par l'application de directives et/ou normes techniques et organisationnelles strictes. Dans ce contexte, nous identifions les points suivants :

3.6.1. Responsabilité juridique sur les données

Du point de vue juridique, Le client est responsable du contenu de ses données (mots de passe, certificats, etc.) et de leur utilisation. Ce qui veut dire que les résultats de sa négligence ne pourraient pas être reprochés au fournisseur. Mais la responsabilité est aussi partagée avec ce dernier, proportionnellement à l'infrastructure qui lui est confiée. Par exemple, le client contrôle seulement ses données dans une infrastructure PaaS, mais ce partage la responsabilité avec le fournisseur dans une infrastructure Saas. Donc, c'est en fonction de la plateforme et de l'architecture que le fournisseur peut être responsable de la sauvegarde et de la confidentialité des données.

Le contrat entre le fournisseur et le client doit expliciter les responsabilités de chaque partie suivant une norme ISO 17789-2014[11]. Ce qui veut dire que chacun maitrise sa propre sécurité, mais aussi contrôle le domaine tiers. Un contrat mal formuler, avec une mauvaise description des responsabilités, engendre des procédures longues et laborieuses en cas de litiges.

3.6.2. Sauvegarde et récupération des données

L'efficacité d'un processus de protection des données est mesurée par le RTO (Recovery Time Objective) qui donne le temps de reprise du service après une panne, et le RPO (Recovery Point Objective) qui représente la quantité maximale de données qu'une opération de restauration peut perdre après une panne (Kokkinos, et al., 2016). Proportionnellement à leur importance, les données critiques pour l'entreprise sont plus fréquemment sauvegarder.

Dans la pratique, c'est le contrat SLA (Service Level Agreements) qui contient une définition des niveaux de services attendus par les utilisateurs er fournis par le fournisseur de service. Il définit aussi les paramètres du processus de protection des données.

[11] DANSK Standard: Information technology, Cloud computing, Reference architecture

3.6.3. Intégrité des Données

L'intégrité dans le contexte du Cloud computing est assurée par le contrôle des droits d'accès. Les systèmes d'exploitation comme le VM implémentent RBAC (Role Based Access Control) pour assurer ce point. RBAC définis un nombre d'actions (permissions et privilèges de lecture, écriture ou exécution) que les différents utilisateurs peuvent réaliser au sein du Cloud. En effet, plusieurs rôles sont associés aux groupes d'utilisateurs selon leurs fonctions ou rôles (collaborateurs, clients ou partenaires, etc.).

3.6.4. Confidentialité des donnée

La confidentialité est le but le plus ardu à assurer dans le contexte de traitement et d'hébergement des données sensibles dans le Cloud. Ce besoin est atteint par le biais de la cryptographie, qui rencontre des défis complexes dans les architectures cloud computing et stockage, particulièrement s'il s'agit de protéger les données de l'entreprise hébergés dans le Cloud contre des accès non autorisés de la part de du propriétaire de service Cloud (hébergeur). Cela est dû au fait que les clés de chiffrement sont stockées en clair dans le serveur Cloud qui les utilisent afin de répondre aux requêtes du client. Pour interdire l'accès des administrateurs du Cloud aux clés et aux données chiffrées, plusieurs travaux ont portés sur les concepts de confidentialité et de respect de la vie privée dans le Cloud, pour la mise en œuvre d'un chiffrement ajusté à ces dispositions et qui fournisse un équilibre acceptable entre sécurité, efficacité et fonctionnalité.

Dans le contexte de stockage dans le Cloud, les algorithmes de chiffrement requêtables, c'est-à-dire qui assurent le matching requêtes-base de données, permettent de traiter des requêtes et de réaliser des recherches dans des bases de données chiffrées sans les déchiffrés dans le coté serveur, ainsi assurer la confidentialité. Cette classe d'algorithme sera détaillée dans la section état de l'art suivante.

3.6.4.1. Fondements sur la Cryptographie

Pendant plusieurs années, la cryptographie était le domaine exclusif de services secrets militaires, diplomatiques et gouvernementaux et a été utilisée pour principalement assurer les butes de la sécurité, telles que la confidentialité de données, l'intégrité de données et l'authentification d'origine de données.

La cryptographie ou le chiffrement est définis comme l'art de codifier le contenu de messages en secrets, puis permettre aux récepteurs de récupérer leur contenu original. En 1976, Diffie Hellman a instauré des changements radicaux dans la théorie de cryptographie, en présentant le premier algorithme de chiffrement asymétrique (Diffie & Hellman, 1976). En 1978, Rivest, Shamir et Adelman ont défini leur célèbre algorithme RSA (Rivest, et al., 1978). Depuis, Shamir a continué à publier des idées révolutionnaires, à savoir, les systèmes basés sur ID, sur l'homomorphisme et autres. Concurremment, les chercheurs (Koblitz, 1987) et (Miller, 1986) ont proposé de façon indépendante des plans cryptographiques originaux basés sur les structures de courbe elliptiques. Récemment, la cryptographie quantique

apparaît comme la cryptographie de l'avenir, puisqu'elle ne se base pas sur l'algèbre abstraite et la théorie de groupes, mais sur des théories où chaque bit est représenté par la polarisation d'un photon.

3.6.4.2. Système Cryptographique

Le système cryptographique est un ensemble d'algorithmes, qui s'appliquent sur des textes clairs, textes chiffrés et a comme paramètres des clés de chiffrement et de déchiffrement. Généralement, un système cryptographique est considéré meilleur s'il a un minimum de paramètres confidentiels et assure au maximum la sécurité des données chiffrées.

a) Cryptographie symétrique

Une distinction fondamentale entre les schémas de cryptographie fait allusion à la relation entre la paire de clés, impliquées dans les algorithmes de chiffrement et de déchiffrement de message. La cryptographie symétrique consiste à un partage de clé secrète entre deux entités Alice et Bob qui communiquent entre eux. La cryptographie symétrique, aussi bien que la cryptographie asymétrique, se base sur l'utilisation de deux algorithmes collatéraux pour le chiffrement et le déchiffrement de message. Formellement, le système cryptographique est un ensemble (P, C, K, E, D), avec :

- P : ensemble de textes en clairs ;
- C : ensemble de textes chiffrés ;
- K : ensemble de clés de chiffrement/déchiffrement ;
- E $= \{E_k : k \in \mathrm{K}\}$: ensemble de méthodes de chiffrement

$$E_k : \mathrm{P} \rightarrow \mathrm{C} \tag{15}$$

- D $= \{D_k : k \in \mathrm{K}\}$: ensemble de méthodes de déchiffrement

$$D_k : \mathrm{C} \rightarrow \mathrm{P} \tag{16}$$

Ainsi, un chiffrement symétrique revient à la vérification de la relation suivante :

$$\forall m \in M, k \in K;\ \mathrm{D}(\mathrm{E}(\mathrm{m},\mathrm{k}),\mathrm{k}) = m \tag{17}$$

L'algorithme de Vernam (Vernam, 1926) est un des algorithmes symétriques bien connus. Il a été proposé par Gilbert Vernam qui suppose que la clé secrète k est une chaîne de bits aléatoires aussi longue que le message m. Comme représenté dans la figure 3.3, quand Alice veut chiffrer un message m, en utilisant une clé secrète k partagée avec Bob, elle calcule $c = E\ (m,\ k) = m \oplus k$. Ainsi, Bob utilise la même clé k pour déchiffrer le message c comme suit : $m = D\ (c,\ k) = c \oplus k$.

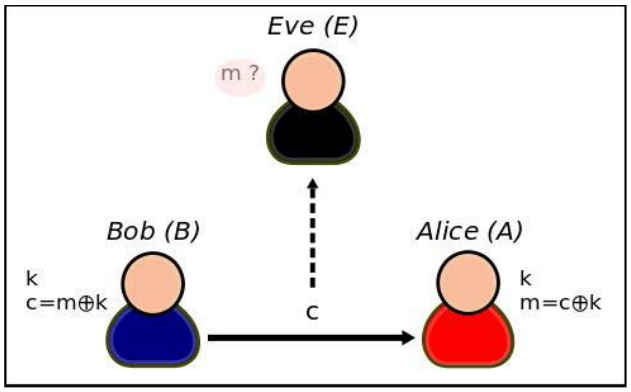

Figure 3-3 : Vernam plan de cryptage

L'algorithme de Vernam est un schéma de chiffrement à clé secrète unique k car elle est utilisée une fois pour chiffrer un message unique. Donc, la clé doit être renouvelée pour chaque message. Cet algorithme est tout à fait sûr puisque le message chiffré ne fournit aucun renseignement sur le message original aux crypte-analyste. C'est une très forte notion de sécurité d'abord développée par Claude Shannon (Shannon, 1949).

Toutefois, Shannon a prouvé que l'algorithme de Vernam révèle beaucoup d'inconvenants, tel que la longueur de la clé secrète |k| qui doit être égal ou plus grand que la longueur du message |m|. En plus, puisque la clé doit être renouvelée pour chaque message, Alice et Bob doivent maintenir un canal de communication sécurisé pour échanger la nouvelle clé secrète pour chaque message transmis. Ce n'est pas réalisable en pratique parce que Alice et Bob gaspilleront la moitié de leur temps de communication dans l'échanger des clés.

Afin de remédier aux problèmes de l'algorithme de Vernam, de nouveaux schémas de chiffrement symétriques ont apparu, appelés des chiffrements par bloc. Ces algorithmes chiffrent des blocs de données, en utilisant de petites clés de longueurs préfixées. Les algorithmes de chiffrement par bloc se basent principalement sur la permutation. Les algorithmes les plus célèbres sont le Data Encryption Standard "DES" (Smid & Branstad, 1992.), et l'Advanced Encryption Standard "AES" (Rijndael, 2001). Ce dernier est largement déployé par plusieurs fournisseurs de service Cloud, tel qu'Amazon Simple Storage Service (Sachdev & Bhansali, 2013).

Les chercheurs dans le domaine des Cloud de stockage donnent plus d'importance à la sécurité des données, toute en considérant l'impact des algorithmes adoptés sur les performances de cloud. Ainsi, les algorithmes de chiffrement symétriques modernes répondent à plusieurs exigences de sécurité de cloud, à savoir, la disponibilité car ils sont rapides et considérablement moins complexes que les algorithmes asymétriques. Donc, ils sont convenables pour traiter de grands flux de données.

Cependant, les algorithmes de chiffrement symétriques supposent qu'Alice et Bob sont capables d'échanger la clé d'une manière sûre avant chaque communication. En tant que tel, la gestion des clés est un défi significatif dans un environnement multi-locataire, surtout les modèles (SaaS). Pour que, les schémas de chiffrement symétriques sont d'habitude combinés à des schémas à clés publics pour obtenir une sécurité accrue.

b) Cryptographie Asymétrique

Le chiffrement asymétrique appelé aussi à clé publique ou Public Key Cryptography (PKC) garantit plusieurs propriétés de sécurité, à savoir la confidentialité de données, la non répudiation et l'authentification, en échangeant des renseignements sur un canal non sécurisé. Contrairement à la cryptographie symétrique où deux entités qui communiquent doivent se partager la clé secrète, la cryptographie à clé publique utilise deux clés : une clé publique et d'une clé privée, où seulement la clé publique est partagée avec les pairs de la communication. Cependant, la clé privée est gardée secrète, comme représenté dans la figure suivante.

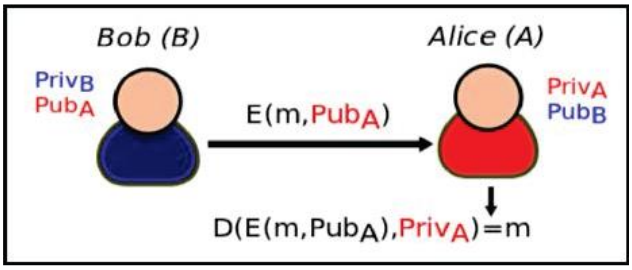

Figure 3-4 : Cryptographie à clé publique

Les algorithmes de chiffrement asymétrique sont basés sur des fonctions facilement calculables mais que leurs fonctions inverses sont extrêmement difficiles à trouver. Nous supposons qu'une fonction "difficile" est "impossible" si nous arrivons à prouver mathématiquement que le temps de calcul de son inverse est presque infini même en utilisant les machines les plus puissantes au monde (plusieurs voir centaines d'années). Les fonctions de chiffrement sont alors choisies afin qu'elles soient difficiles à inverser sauf si nous avons la clé privé tenue secrète. De ce fait, un schéma de chiffrement asymétrique utilise une paire de clés : clé publique et une clé privée. La clé publique est distribuée à travers des annuaires communs pour qu'elle soit connue de tous les participants à la communication. Alors que la clé privée n'est connue que de la personne qui à générer la paire de clés, et qui seule peut déchiffrer les messages. Ce schéma a été proposé par Deffie-Hellman (Diffie & Hellman, 1976). Amélioré par la suite par les travaux de R.L. Rivest, A. Shamir, L. Adleman sur l'algorithme RSA (Rivest, et al., 1978).

Le chiffrement asymétrique est plus fiable par rapport au chiffrement symétrique. Du fait qu'une communauté qui utilise le chiffrement asymétrique de n intervenants n'aura besoin que de de n clés publiques et n clés secrètes pour assurer la confidentialité. Contrairement au cas d'une communauté qui utilise le chiffrement symétrique qui aura besoin de $n*(n-1)$ clés.

Cependant, les cryptosystèmes asymétrique sont lents (RSA est 1000 fois moins rapide que DES). D'une autre part, les preuves mathématiques qui accompagne les implémentations du modèle asymétrique reposent sur la limitation des ordinateurs actuels à réaliser des calculs arithmétiques particuliers et donc résoudre certains problèmes comme le logarithme discret et la factorisation des grands nombres dans un petit laps de temps. Actuellement, nous

sommes toujours à l'abri de ce danger puisque aucune amélioration exponentielle dans les processeurs arithmétiques des machines de calcul actuelles n'a surgi, et un système de chiffrement à clé publique ne peut pas devenir obsolète.

3.6.5. Limites des Systèmes Cryptographiques Traditionnels dans le Cloud Computing

En dépit des garanties que fournissent les systèmes cryptographiques traditionnels symétriques ou asymétriques envers la confidentialité, ils peuvent être insuffisants pour les données stockés dans le Cloud. En fait, plusieurs restrictions réduisent l'apport de ces schémas traditionnels, surtout en raison de l'énorme quantité de données chargée. Dans les environnements Cloud de stockage, la bande passante, la mémoire et les consommations d'énergie sont très importants, puisqu'ils ont un impact direct sur la disponibilité et les performances de services fournis. Par conséquent, la sélection d'outils cryptographiques adéquats pour le soutien de sécurité doit être justifiée.

Premièrement, si le modèle de sécurité du fournisseur de services Cloud est non fiable, le client voudra généralement chiffrer ces données avant de les charger aux serveurs distants. Ainsi, l'usage d'algorithmes asymétriques traditionnels au côté de client est trop lent, surtout pour de grandes quantités de données car les capacités de calcul de la machine client sont significativement réduites.

De même, les algorithmes asymétriques classiques exigent aussi le déploiement d'une infrastructure pour clé public PKI et des fonctions de gestion de certificat responsable de la génération et la livraison de certificats aux entités authentifiées. En plus, le téléchargement périodique de listes de révocation par les clients depuis l'Autorité de Certification (CA) est nécessaire pour vérifier la validité de certificats. Ainsi, la bande passante et la disponibilité seront détériorées.

Deuxièmement, si des schémas cryptographiques symétriques sont adoptés pour chiffrer des données du côté client, cette dernière réserve les clés de déchiffrement hors de portée du fournisseur de services. Cependant, la gestion de confidentialité devient plus complexe avec les données un besoin de partage de donnée entre groupe d'utilisateurs.

Finalement, plusieurs applications qui permettent l'indexation des données ont émergé pour effectuer des recherches rapides dans des données stockées dans une infrastructure Cloud, à savoir, l'Apple Spotlight et le Bureau Google. Mais puisque les outils et les algorithmes traditionnels de chiffrement sont déterministes, ils ne sont pas malléables et ne permettent pas des opérations sur les données chiffrées, telles que la recherche dans des textes chiffrés très commode pour récupérer des informations depuis des données hébergées sur les serveurs cloud ou autres.

3.7 Conclusion

La technologie du Cloud Computing a révolutionné le monde de la Technologie de l'Information. Les logiciels, applications et données d'une entreprise et même d'un particulier les suivent partout sans qu'ils investissent dans l'achat de serveur, licences de logiciel ou recruter des agents de maintenance. Le fournisseur du Cloud offre des panoplies de services à la demande, mais néglige toutefois les procédures et les mesures de sécurité à mettre en place. Ce qui exige une conscience juridique et des outils physique et logique de sauvegarde, récupération ainsi que des mécanismes et des algorithmes qui assurent la confidentialité et la sécurité des milliers d'entrepôts électroniques qui hébergent et conservent des milliards de données pour qu'on soit les seuls à avoir accès à nos données.

Chapitre IV.

Retrait d'Information Privé : État de l'Art.

Chapitre IV. Retrait d'Information Privé : Etat de l'Art

4.1 Introduction

Les services de Cloud Computing et de stockage comme paradigme émergeant, implique le déplacement des données privée vers des plateformes externes. Surtout en raison de cette perte de contrôle direct sur leurs données, les utilisateurs hésitent avant d'adopter les services de Cloud. Ces inquiétudes sur la confidentialité et la sécurité de données sont tout à fait légitimes, étant donné les dernières révélations des media tel l'évènement du novembre de 2013, ou le Washington Post a prouver plus d'activités de capture de flux de données par l'Agence de sécurité nationale (NSA[12] et PIRSM[13]) des états unis qui est en effet une interception des communications de données privés qui circulées entre les utilisateurs et les serveurs de données Yahoo et Google répartis dans le monde entier, après bien sûr décrypter le contenu qui devrait être protégée lors du transit (LOSEY, 2015). Donc, plusieurs mesures de sécurité doivent être mise en place pour répondre aux inquiétudes émergées, à savoir le chiffrement des données et l'intégrité ainsi toute en respectant la disponibilité qui dépond de la bande passante.

Dans ce chapitre, nous fournissons un état de l'art résumé à quelques techniques cryptographiques pour garantir la sécurité et la confidentialité dans les Cloud. Ensuite, nous donnons une vue d'ensemble des travaux et des systèmes cryptographiques dans les environnements Cloud de stockage en relation avec notre contribution, spécialement les algorithmes homomorphes.

4.2 Protocoles de Retrait d'Information Privé PIR : Etat de l'Art

La cryptographie moderne fournit des mécanismes de déchiffrement beaucoup plus flexibles et permet explicitement une meilleure malléabilité sur les textes chiffrés, à savoir la recherche de contenue chiffré (confidentialité), les Proof of Data Possession PDP et les Proof of Data Retrievability (intégrité). Ces avancées sont très intéressantes dans un contexte Cloud multi-locataire.

Dans la section suivante, nous introduisons un état de l'art des schémas asymétriques émergeant dans les environnements Cloud de stockage. Premièrement, nous introduisons l'approche Identité Based Cryptographie (IBC), où la clé publique d'une entité est directement tirée de son identité, sans besoin de certificats. Puis, nous présentons les approches Attribut Based Cryptographie (ABC). Finalement, nous étalons la description des systèmes de cryptographie homomorphe, qui permettent plusieurs opérations mathématiques sur les données chiffrées.

[12] "top-secret accounting dated Jan. 9, 2013"

[13] "PRISM: Here's how the NSA wiretapped the Internet. June 8, 2013"

4.2.1. Identity Based Cryptography (IBC)

En 1984, le schéma Identity Based Cryptography IBC ou Cryptographie basée sur les Identifiants a été introduite par Shamir (Shamir, 1985) qui implémente l'idée originale de fournir des paires de clés publiques/privées sans avoir besoin de certificats ni de déploiement de CA. Shamir suppose que chaque entité utilise un de ses identifiants comme sa clé publique. À condition que ces identifiants doivent être uniques. En plus, il assigne la fonction de génération clé privée à une entité spéciale appelée le Générateur clé privé ou Public Key Generator (PKG). C'est-à-dire, avant d'accéder au réseau, chaque entité doit contacter le PKG pour recevoir sa clé privée. Cette clé privée est calculée depuis la clé publique de l'entité. (Boneh & Franklin, 2001), proposent le premier schéma de chiffrement basé sur l'Identifiant des entités, et utilisant de fonctions d'appariement bilinéaires pour lier des points de courbe elliptiques à un certain nombre un groupe multiplicatif. Nous notons que les certificats peuvent être considérés comme une propriété de l'identifiant, puisqu'ils associent la clé publique de l'utilisateur à son identifiant. Durant toute une décennie, plusieurs améliorations de l'approche IBC a utilisé le chiffrement à base des courbes elliptique (ECC) (Kalyani & Sridevi, 2016).

Dans les sections suivantes, nous présentons d'abord les fonctions d'appariements qui ont été largement utilisées dans les systèmes cryptographiques modernes, puis nous introduisons le processus de génération de clé pour les schémas basés sur l'ID, qui sont aussi basés sur l'appariement des fonctions.

a) Préalables lors de l'Appariement des Fonctions

La fonction d'appariement $\hat{e}$ est une relation bilinéaire qui doit vérifier les propriétés suivantes :

- La Bilinéarité : la fonction d'appariement $\hat{e} : G_1 \times G_2 \rightarrow G_T$ est linéaire par rapport à chacune de ses entrées comme suit :

$$\forall P \in G_1, \forall Q \in G_2 \; et \; \{a, b\} \in Z^2, \quad \hat{e}(a.P + b.P, Q) = \hat{e}(P, Q)^a \hat{e}(P, Q)^b \tag{18}$$

$$\forall P \in G_1, \forall Q \in G_2 \; et \; \{a, b\} \in Z^2, \quad \hat{e}(P, a.Q + b.Q) = \hat{e}(P, Q)^a \hat{e}(P, Q)^b \tag{19}$$

- Le non dégénérescence - cette propriété définit deux relations comme suit :

$$\forall P \in G_1, \hat{e}(P, Q_\infty) = 1_{G_T} \tag{20}$$

$$\forall P \in G_1, \hat{e}(P_\infty, Q) = 1_{G_T} \tag{21}$$

  Considérons un générateur P de G_1 et d'un générateur Q de G_2, pour que la valeur $\hat{e}\ (P,\ Q)$ soit égale au générateur de G_T.

- L'efficacité : la propriété d'Efficacité signifie qu'il existe un algorithme qui calcule la fonction d'appariement.

Les fonctions d'appariement peuvent être divisées en fonctions symétriques et fonctions asymétriques. L'appariement symétrique exige le même groupe d'entrée G1= G2, alors que

l'appariement asymétrique vérifient G1 ≠ G2. En pratique, les fonctions d'appariement bilinéaires sont généralement dérivées des fonctions de Weil ou Tate (Blake, et al., 2005).

b) Génération de Clé basée sur l'ID

Pour être capable de générer la clé privée d'un client, le PKG définit d'abord un ensemble d'éléments publics ID Based Public elements (IBC-PE). Le PKG construit les groupes G_1, G_2 et G_T et la fonction d'appariement $\hat{e}$ depuis $G_1 \times G_2$ dans G_T. G_1 et G_2 sont des sous-groupes additifs du groupe de points d'une Courbe Elliptique (EC). Cependant, G_T est un sous-groupe fini multiplicatif. G_1, G_2 et G_T ont le même ordre q. En plus, G_1, G_2 et G_T sont générés par P, Q et le générateur $g = \hat{e}(P, Q)$, respectivement. La fonction bilinéaire ê est généralement dérivée de modèle Weil ou Tate (Blake, et al., 2005).

Après la spécification des groupes, le PKG définit un ensemble de fonctions de hachage conformes au schéma de chiffrement et de signature basé sur ID. Par exemple, le PKG définit une fonction de hachage $Hash_{pub}()$ pour transformer l'identifiant du client (ID) en clé publique comme suit :

$$PubID = Hash_{pub}(ID) \tag{22}$$

Généralement, la clé publique d'un client est le résultat de hachage de l'un de ses identifiants et c'est soit un point d'une courbe elliptique (Boneh & Franklin, 2001) ou un nombre entier positif (Sakai & Kasahara, 2003).

Le PKG génère la clé privée d'une entité en utilisant un attribut secret local $s_{PKG} \in Z^*$ et une fonction de génération clé privée $PrivGen()$. Notez que la clé privée est calculée comme :

$$PrivID = PrivGen(s_{PKG}, PubID) \tag{23}$$

Par exemple, (Boneh & Franklin, 2001) calculent la clé privée comme suit : $PrivID = s_{PKG} * PubID$, où $PubID$ est un point $\in$ G1. Cependant, (Sakai & Kasahara, 2003) produisent la clé privée comme $PrivID = (1 / (PubID + s_{PKG})) * P$, où $PubID$ est un nombre entier. Différemment, Boneh et Boyen définissent une autre fonction de dérivation clé (Boneh & Boyen, 2004). C'est-à-dire, ils calculent d'abord trois points public P_1, P_2 et P_3 tel que :"$P_1 = \alpha.P, P_2 = \beta.P$, et $P_1 = \gamma.P$", où…α, β et γ sont des secrets choisis par le PKG. Boyen et de Boneh calcule la clé privée de l'utilisateur comme un couple de point d'une courbe elliptique $Priv_{ID} = (Priv_1, Priv_2) = (Pub_{ID}.r.P_1 + \alpha.P_2 + r.P_3, r.P)$, où $r \in Z^*$.

Les groupes G_1 et G_2, l'appariement $\hat{e}$, les points P, Q et $Q_{pub} = s_{PKG}.Q$ et les fonctions de hachage $H_1() \ldots, H_k()$ forment les éléments publics du schéma de chiffrement à base d'ID, comme suit :

$$IBC - PE = \{G_1, G_2, G_T, q, \hat{e}, g, P, Q, Q_{pub}, Hash_{pub}(\ ), H_1(), \ldots, H_k()\} \tag{24}$$

Après avoir produit une clé privée, le PKG doit protéger sa transmission à son propriétaire soit à l'aide de chiffrement ou directement à la personne physique. Dans les schémas précédents de génération de clés, le PKG calcule les clés privées pour les entités membre de

la communication. Ainsi, le *PKG* est capable d'imiter n'importe quel utilisateur en déchiffrant illégalement les données chiffrées. Cette attaque est connue sous le nom *Key Escrow Attack (*KEA). Ce qui implique l'hypothèse que le *PKG* est une entité fortement fiable (Chen, et al., 2006).

4.3 Protocoles IBC dans le Cloud : état de l'art

L'application des schémas Cryptographiques à Base d'ID (IBC) dans un environnement distribué est un domaine partiellement exploré dans la littérature. IBC a été d'abord adapté aux grilles de calcule. Ian Foster (Foster, 2002) définit la grille comme des ressources informatiques avec une administration décentralisée, des méthodes standardisées, et des ressources utilisées pour accomplir un but commun. Récemment, Ian Foster et autres (Foster, et al., 2009) montrent que les grilles ressemblent aux Cloud dans la technologie et l'architecture, mais ils Diffèrent dans d'autres aspects tels que les modèles de sécurité. L'idée d'appliquer le modèle IBC pour assurer la sécurité des grilles a été explorée par Lim et Robshaw en 2004 (W. & Robshaw, 2004). Dans leur papier, chaque organisation virtuelle a son propre *PKG* et tous ses utilisateurs partagent même *IBC-PE* certifié par une autorité de certification de grille. Leur schéma offre à l'entité qui chiffre les données plus de flexibilité pendant le processus de génération de clé et autorise à ajouter de la granularité à la clé publique. En fait, Lim et Robshaw proposent d'inclure la politique de sécurité dans l'identifiant utilisé comme input pour l'algorithme de génération de la clé public. Cependant, leur proposition a deux inconvenants. Premièrement, l'utilisateur a besoin de maintenir un canal sûr indépendant avec le *PKG* pour la récupération de sa clé privée. Deuxièmement, le *PKG* est capable d'exécuter une attaque de type Key Escrow, il connaît tous les clés privées des clients.

Ensuite, Lim et Robshaw (Lim & Robshaw, 2005) ont introduit le concept d'infrastructure de clé dynamique dans la grille, pour simplifier Les principaux problèmes de gestion énumérés dans (Lim & Robshow, 2004). Ils ont proposé une approche hybride qui combine le mécanisme IBC implémenté au niveau du client et l'approche traditionnel PKI pour soutenir gestion de clé au-dessus du niveau du client. Dans (Lim & Robshaw, 2005), chaque utilisateur distribue un jeu de paramètre fixe à travers le standard X.509. Ce paramètre permet de négliger le besoin d'un proxy de certification, et d'éviter l'attaque de type Key Escrow et ou même le besoin d'un canal sûr pour la distribution clé privée dans un système IBC. Malheureusement, les utilisateurs sont obligés de vérifier l'ensemble des paramètres de tous les autres entités, ce qui est fastidieux. En plus, cette contribution ne gère pas les attaques de type Man in the Middle (Schridde, et al., 2010).

En 2011, Lim et Paterson ont à leurs tours proposé d'utiliser IBC pour protéger les grilles (Lim & PAterson, 2011). Ils décrivent plusieurs scénarios dans lequel IBC simplifies la sécurité des solutions grille actuelles, par l'élimination de l'utilisation de certificats, génération de proxy simple, révocation facile de proxy de certification et l'économie en bande passante en utilisant l'approche d'appariement proposée par (Boneh & Franklin, 2001).

De la même façon, Li et autres proposent d'utiliser IBC comme une alternative au protocole authentique SSL dans un environnement Cloud (Li, et al., 2009). Ils introduisent le modèle IBC hiérarchique de trois couches. La couche supérieure est la racine PKG et correspond à l'administrateur du cloud. La deuxième couche présente un sub-PKG qui correspond à un data center du cloud, tandis que la troisième couche est tout client de cloud. En tant que tel, chaque clé publique de client est dérivée de la concaténation d'un ensemble d'identités dans le modèle hiérarchique. Évidemment, une hiérarchie de confiance est indispensable pour garantir la fiabilité.

Récemment, Schridde et autres ont présenté une infrastructure de sécurité originale, en utilisant IBC, dédié aux architectures SaaS afin de surmonter les problèmes relatifs aux solutions à base de certificat (Schridde, et al., 2010) . Dans leur proposition, chaque client doit être enregistré à un serveur d'autorité. L'enregistrement inclut la spécification de la méthode de payement pour octroyer le service désiré et ainsi récupérer du crédit d'ouverture de session associé au compte correspondant. Pendant l'enregistrement, chaque client obtient une clé d'identité privée unique pour le compte chois. Bien que l'auteur de ce papier résous le problème de maintien d'un environnement fiable, l'usage d'une identité unique implique un problème de partage de données parmi un groupe dynamique d'utilisateurs.

4.3.1. Attribut Base Cryptography ABC

En 2005, Sahai et Waters ont introduit le concept de chiffrement basé sur les attributs ou Attribut Base Cryptography ABC (Sahai & Waters, 2005), comme un nouveau moyen contrôle d'accès chiffré. Dans ABC, le texte chiffré n'est pas chiffré pour un utilisateur particulier comme dans la cryptographie clé publique traditionnelle. Ainsi, les clés privées et texte chiffré est associés à un ensemble d'attributs ou à une politique sur les attributs. L'utilisateur est capable de déchiffrer un message s'il y a un match entre sa clé privée et le message chiffré.

Un état de l'art des approches ABC (Lee, et al., 2013) donne une vue d'ensemble des schémas de chiffrement basés sur les attributs et illustre les applications et leurs propriétés intéressantes dans les environnements cloud de stockage. Parmi ces applications, nous distinguant le travail de Sahai et Waters (Sahai & Waters, 2005), qui ont présenté un schéma Attribute Based Encryption (ABE) muni d'un seuil. C'est-à-dire, les messages chiffrés sont étiquetés par un ensemble d'attributs S et la clé privée de l'utilisateur est associée à un paramètre de seuil t et à un autre ensemble des attributs S'. Pour déchiffrer des données, au moins t attributs doivent correspondre entre le texte chiffré et la clé privée. Une des motivations de ce travail est la conception d'un schéma de chiffrement tolérant à l'erreur (fuzzy) à base de données biométriques.

(Goyal, et al., 2006) Et (Bethencourt, et al., 2007) ont proposé le protocole Key-Policy Attribute Based Encryption (KP-ABE) qui imbrique les paramètres de la politique d'accès dans clé privée de l'utilisateur. Les données chiffrées sont étiquetées par un ensemble d'attributs et les clés privées sont associées aux structures d'accès qui contrôlent quelle donnée chiffrée un utilisateur est autorisé à déchiffrée. Les schémas KP-ABE garantissent la

flexibilité et le contrôle d'accès accru. Cela élimine le besoin de compter sur le serveur de stockage pour prévenir les accès non autorisés. Cependant, l'inconvénient de KP-ABE est que la politique d'accès se trouve sur la clé privée ce qui veut dire que le propriétaire de données ne peut pas sélectionner qui peut déchiffrer les données, mais peut munir la clé privée d'un ensemble d'attributs qui peuvent décrire les données. KP-ABE est proche conceptuellement aux méthodes de contrôle d'accès traditionnelles telles que le Rôle Based Access Control (RBAC).

En 2010, Yu et al., ont proposé une politique basée clé du protocole ABE pour protéger des données externalisées dans le cloud (Yu, et al., 2010), où un simple propriétaire de données peut crypter ses données et les partager avec plusieurs utilisateurs autorisés, en leur distribuant des clés. Les clés distribuées contiennent des droits d'accès sous forme d'attributs. Comme les opérations de mise à jour des attributs des clés peuvent être regroupées au fil du temps, le schéma proposé présente de faibles seuils de surcharge de communication.

4.3.2. Cryptographie Homomorphe

L'idée de base des algorithmes de chiffrement homomorphes est de chiffrer les données avant de les envoyer aux fournisseurs du Cloud, mais ce fournisseur devra les déchiffrer (donc doit avoir l'accès à la clé de déchiffrement) à chaque fois qu'il a besoin d'effectuer des traitements dessus (car il ne pourra pas réaliser un traitement sur des données chiffrées sans les déchiffrer), ce qui pourra nuire à la confidentialité des données stockées dans le Cloud (Fahsi & Benslimane, 2014a). Il y a peu de temps, il était impossible de chiffrer des données et de les confier à un tiers pour les garder en sécurité et pouvoir exécuter des calculs distants sur ces mêmes données sans les déchiffrer. Mais quelle est la méthode qui permet de réaliser des opérations sur des données chiffrées sans les déchiffrer ? C'est bien le chiffrement Homomorphe.

Avant de définir le chiffrement Homomorphe, nous rappelons d'abord la cryptographie asymétrique ou à clé publique : Considérons deux ensembles arbitraires X, Y, avec f :$X \rightarrow Y$. Soit f(X) l'ensemble image de X par f. La fonction f est dite à sens unique si pour tout x de X, il est facile de calculer f(x), mais il est difficile de retrouver, pour tout y $\in$ f(X) un x $\in$ X tel que f(x)=y.

Les fonctions à sens unique ne peuvent pas servir telles quelles comme système de chiffrement, en utilisant par exemple f(M) pour chiffrer un message M puisque même le destinataire légal ne serait pas en mesure de déchiffrer le cryptogramme. La notion de fonction à sens unique à trappe est d'une utilité plus immédiate pour la cryptographie à clé publique.

Définition :

Une fonction $f: X \rightarrow Y$ est dite à trappe si elle peut être calculée efficacement dans le sens direct. Le calcul dans le sens inverse est aussi efficace pourvu qu'on dispose d'une information secrète, la trappe, qui permet de construire une fonction g telle que g o f = Id. Ainsi, il est

facile de calculer l'image de f de n'importe quelle entrée mais calculatoirement impossible d'inverser f sans connaître g. La publication de f ne doit rien révéler sur g (Xun, et al., 2014).

La sécurité est d'autant plus une préoccupation primordiale surtout dans les domaines de très haute importance (telle la sûreté militaire ou politique d'un pays, données bancaires, données médicales et vie privée) que la cybercriminalité ne cesse de progresser et d'évoluer. Les techniques de chiffrement n'ont cessé d'évoluer depuis leur apparition pour permettre d'atteindre le niveau de sécurité souhaité. La cryptographie asymétrique ou (à clé publique) apparue dans les années 70 est un procédé asymétrique utilisant une paire de clés pour le chiffrement, soit une clé publique qui chiffre des données, et une clé privée pour le déchiffrement. À noter également qu'il est impossible de deviner la clé privée à partir de la clé publique. Cette propriété est utilisée pour assurer la confidentialité et la signature électronique (intégrité). Le principe est donc de distribuer la clé publique tout en conservant la clé privée secrète. Tout utilisateur possédant une copie de la clé publique pourra ensuite chiffrer des informations que seul le propriétaire de la clé privée pourra déchiffrer.

Les avantages de la cryptographie asymétrique :

- Impossibilité de substitution du destinataire : clé privée connue de lui seul
- Aucun transfert de clé privée : confidentialité assurée
- Un seul couple de clés pour plusieurs expéditeurs

Les inconvénients de la cryptographie asymétrique :

- Complexe
- Authentification incertaine de l'expéditeur
- Requiert beaucoup d'opérations, donc peu recommandé pour transférer de grandes quantités de données

Le chiffrement homomorphe a connu ces dernières années des avancées théoriques importantes. Malheureusement, très peu d'implémentations de ces systèmes ont été présentées et discutées, laissant la question de l'application de ce mode de chiffrement sur les données sensibles hébergées dans le Cloud ouverte.

Les systèmes de chiffrement Homomorphe permettent d'effectuer des opérations sur des données chiffrées sans connaître la clé secrète (donc sans déchiffrer), au bénéfice de l'utilisateur, seul possesseur de la clé secrète. Lorsqu'on déchiffre le résultat de l'opération, il est bien le même comme si nous avions mené les calculs sur les données brutes.

Dans ce qui suit la fonction Enc est une fonction de chiffrement d'un message m avec la clé publique P_k ; Dec est la fonction de déchiffrement correspondante en utilisant la clé secrète S_k .

Définition : Un chiffrement est homomorphe si à partir de *Enc(a)* et *Enc(b)*, il est possible de calculer *Enc(f (a, b))* où f peut être, par exemple : $+, \times, \oplus$ et sans que la clé privée ne soit utilisée.

Parmi les chiffrements Homomorphes, nous distinguons, selon l'opération qu'ils permettent d'évaluer sur des clairs, les chiffrements homomorphes additifs (se basent sur des opérations d'addition uniquement), c'est le cas des cryptosystèmes de Pailler et de Goldwasser-Micalli et les chiffrements homomorphe multiplicatifs (se basent sur des opérations de multiplication uniquement) tels que les cryptosystèmes de RSA et d'El Gamal (Xun, et al., 2014).

4.3.2.1. Historique du chiffrement Homomorphe :

Ronald Rivest et autres ont évoqué pour la première fois le concept de chiffrement homomorphe (Rivest, et al., 1978). Depuis, peu d'avancées ont été faites pendant 30 ans. Le système de chiffrement Shafi Goldwasser et Silvio Micali (Goldwasser & Micali, 1984) avais proposé un schéma de chiffrement à sécurité prouvée. Il est homomorphe pour l'addition, mais ne peut chiffrer qu'un seul bit. Suivant le même concept, Pascal Paillier (Paillier, 1999) propose un système de chiffrement efficace à sécurité prouvée qui est homomorphe pour l'addition. Peu d'années après, (Boneh, et al., 2005) inventent un système de chiffrement homomorphe à sécurité prouvée, avec lequel nous effectuerons un nombre illimité d'additions mais une seule multiplication. En 2009, Craig Gentry a proposé le premier système de chiffrement "complètement homomorphe" qui permet d'évaluer un nombre arbitraire d'additions et de multiplications et donc calculer tout type de fonctions sur des données chiffrées (Gentry, 2009). Il est toujours en phase d'expérimentation car sa durée de chiffrement et de déchiffrement est loin d'être acceptable. En 2010, Van Dijk et autres ont proposé un système de chiffrement "complètement homomorphe" avec une limitation à un seul bit (Van Dijk, et al., 2010).

Dans la suite de ce chapitre, nous allons analyser les différents algorithmes de chiffrement Homomorphes, citer leurs avantages et inconvénients et nous allons programmer et simuler les algorithmes Homomorphes RSA et Pallier pour comparer leurs durées de traitement, de chiffrement et de déchiffrement à celles de Golwasser-Micalli et cela en fonction de la taille de la clé et du texte à chiffrer.

4.3.2.2. Chiffrement Homomorphe additif

Dans un contexte de chiffrement additif, un serveur distant pourra retourner le résultat d'une opération d'addition sur les messages en clair en faisant le calcul sur des messages chiffrés, sans disposer de la clé secrète.

Définition 3.2 : Un chiffrement homomorphe est additif si : $\text{Enc}(x \otimes y) = \text{Enc}(x) \oplus Enc(y)$ et $\prod_{i=1}^{n} Enc(m_i) = Enc(\sum_{i=1}^{n} m_i)$

En d'autres termes, soient :

- Enc_p une fonction de chiffrement à clé publique p.
- Dec_s une fonction de déchiffrement à clé secrète s.

Alors :

$$Dec_s\left(Enc_p(m) \times Enc_p(n)\right) = m + n \quad (25)$$

Les chiffrements qui réalisent cette propriété de chiffrement Homomorphe additif sont : Paillier et Goldwasser-Micalli.

a) Le chiffrement Homomorphe de Paillier

Le crypto système de Paillier est un cryptosystème asymétrique, conçu par Pascal Paillier en 1999. Ce cryptosystème est celui qui a la plus grande bande passante, appelée aussi taux d'expansion : rapport entre la longueur du clair et la longueur du chiffré (Paillier, 1999). Ce cryptosystème est basé sur les propriétés de la fonction lambda de Carmichael dans Z.

L'algorithme de Paillier est détaillé ci-dessous :

Génération des clés :
- Choisir p et q premiers
- Calculer $n = pq$
- Choisir $g \in Z^*_{n^2}$ tel que :

PPCM (L ($g^{\lambda} \bmod n^2$), n) = 1 avec L(u) = $\frac{u-1}{n}$

Clé publique : pk = (n, g)

Clé privée : sk = (p, q)

Chiffrement : Enc (m, pk, r)
- Choisir $r \in Z^*_n$
- Calculer $c = g^m . r^n \bmod n^2$

Déchiffrement : Dec (c, sk)
- Calculer $m = \frac{L(c^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)} \ mod\ n$ L

Comme explication des étapes, il y'a la génération des clés :

- On choisit deux grands nombres premiers p et q ;
- On Calcule n = pq ;

- On choisit un entier g $\in Z_{n^2}^*$ tel que n et L (g^{λ} mod n^2) sont premiers entre eux, où L désigne la fonction :

$$L : Z_{n^2}^* \to Z_n \quad u \to \frac{u-1}{n} \tag{26}$$

λ Désigne la fonction de Carmichael : $\lambda(p.q) = ppcm(p-1\ q-1)$.

La clé publique est donc formée de (n, g) et la clé privée des deux facteurs premiers *(p, q)*.

Supposons que nous avons c_1 et c_2 deux textes chiffrées avec l'algorithme de Paillier et m_1 et m_2 les textes clairs correspondants tel que :

$$c_1 = g^{m_1}.r_1^n \ mod\ n^2$$
$$c_{é} = g^{m_2}.r_2^n \ mod\ n^2$$

Alors :

$$c_1.c_2 = g^{m_1}.r_1^n.g^{m_2}.r_2^n \ \ mod\ n^2$$
$$= g^{m_2+m_2}.(r_1r_2)^n \ mod\ n^2$$

Donc, le chiffrement de Pailler réalise la propriété du chiffrement Homomorphe additif.

b) Le chiffrement Homomorphe de Goldwasser-Micalli

Le cryptosystème de Goldwasser-Micalli (GM) est un algorithme asymétrique de cryptographie à clé publique, développé par Shafi Goldwasser et Silvio Micali. Goldwasser et Micali ont introduit la notion de chiffrement probabiliste, tout système de chiffrement doit intégrer de l'aléa dans le processus de chiffrement pour être considéré comme sûr. Le schéma de GM (Goldwasser & Micali, 1984) qui repose sur la difficulté du problème du résidu quadratique n'est pas efficace : les textes chiffrés peuvent être des centaines de fois plus longues que les textes d'origine.

Génération des clés :

Le problème de résidu quadratique : étant donné un entier composite impair n, et un a$\in Z_{n^2}^*$, tel que $(\frac{a}{n})$=1, décider si (a) est ou non un résidu quadratique modulo n.

Supposons n = pq (produit de deux nombres premiers), $(\frac{a}{n})$ = 1 implique soit $(\frac{a}{p})$= $(\frac{a}{q})$= 1 (a est résidu quadratique) ou $(\frac{a}{p})$= $(\frac{a}{q})$= -1 (a est non-résidu quadratique).

- On Choisit p et q premiers ;
- On Calcule n=pq ; *z*
- On Choisit $z \in Z_n$ tel que z soit un résidu non quadratique modulo n et $(\frac{z}{n})$= 1

La clé publique est donc formée de (n, z) et la clé privée des deux facteurs premiers (p, q).

L'algorithme de Goldwasser-Micalli se présente comme suit :

Génération des clés :
- Choisir p et q premiers
- Calculer $n = pq$
- Choisir z ∈ Z_n tel que : $(\frac{z}{n})$= 1 et $(\frac{z}{p})$ =−1

Clé publique : pk = (n, z)
Clé privée : sk = (p, q)

Chiffrement : Enc (m_i, pk, r_i)
- soit M = {0,1} l'espace des messages en clair ;

Pour tout m∈M, m est composé de t bits : $m_1m_2...m_t$
- choisir aléatoirement pour $\forall i \in [1,t]$ un r_i
- Calculer $c_i = z^{mi}.r_i^2 \bmod n$

Déchiffrement : Dec (c, sk)
- Calculer $\left(\frac{c_i}{p}\right) = e_i$ pour $\forall i \in [1,t]$ avec $c=c_1c_2...c_t$

Si e_i=1 alors m_i=0 sinon m_i=1

Le chiffrement de Goldwasser-Micalli est un chiffrement XOR-Homomorphe.

4.3.2.3. Chiffrement Homomorphe multiplicatif

Par analogie avec ce qui précède, un système basé sur le chiffrement homomorphe multiplicatif permet de n'effectuer que des produits sur les clairs, sans disposer de la clé secrète.

Définition :

Un chiffrement homomorphe est multiplicatif si :

$$\mathrm{Enc}(x \otimes y) = \mathrm{Enc}(x) \otimes Enc(y)$$

$$\prod_{i=1}^{n} Enc(m_i) = Enc(\prod_{i=1}^{n} m_i)$$

En d'autres termes, soient :

Enc_p une fonction de chiffrement à clé publique p.

Dec_s une fonction de déchiffrement à clé secrète s.

Alors : $Dec_s\left(Enc_p(m) \times Enc_p(n)\right) = m \times n$

Parmi les algorithmes Homomorphes permettant ce type d'opération, nous citons RSA et El Gamal.

a) Le chiffrement Homomorphe de RSA

Le premier système à clé publique à être proposé fut celui de Ronald Rivest, Adi Shamir et Leonard Adleman connu sous le nom RSA. Cet algorithme a été décrit en 1978 (Rivest, et al., 1978). Parmi tous les systèmes cryptographiques asymétriques à l'heure actuelle, RSA

est considéré comme un des plus solides, si ce n'est le plus solide. Il a résisté à des années de cryptanalyse intensive et il est encore jugé assez robuste pour protéger les échanges bancaires et autres données critiques. Ce niveau de sécurité réside dans la difficulté de factoriser des grands nombres. Retrouver le texte en clair à partir d'une clé et du texte chiffré est supposé équivalent à la factorisation du produit des deux nombres premiers.

Les étapes de la génération des clés de chiffrement et déchiffrement de l'algorithme de RSA sont les suivantes :

Génération des clés :

- On Choisit deux nombres premiers p et q et nous calculerons le produit n=pq ;
- On Choisit ensuite une clé de chiffrement aléatoire e, tel que e et $(p-1)(q-1)$ soient premiers entre eux ;
- Finalement, nous calculerons la clé de déchiffrement de telle manière que :

$$d = e^{-1} mod\ ((p-1)(q-1))$$

- La clé publique est donc formée des deux nombres e et n, la clé privée est le nombre d.

Ci-dessous l'algorithme de RSA en détail :

Malgré sa robustesse, ce système s'avère vulnérable à l'attaque de l'homme du milieu,

Génération des clés :
- Choisir p et q premiers
- Calculer $n = pq$; $\emptyset(n) = (p-1)(q-1)$
- déterminer d tel que : $e.d \equiv 1 \bmod \emptyset(n)$

Clé publique : pk = (e, n)
Clé privée : sk = d

Chiffrement : Enc (m, pk)
- Calculer $c = m^e \bmod n$

Déchiffrement : Dec (c, sk)
- Calculer $m = c^d\ mod\ n$

c'est à dire par interception et remplacement de la clé publique, l'attaquant récupère la clé publique d'un interlocuteur et fournit au second sa propre clé publique à la place. Supposons que nous avons c_1 et c_2 deux textes chiffrées avec l'algorithme de RSA et m_1 et m_2 les textes clairs correspondants tel que :

$$c_1 = m_1^e \mod n$$
$$c_2 = m_2^e \mod n$$
$$c_1.c_2 = m_1^e m_2^e \mod n = (m_1 m_2)^e \mod n$$

En déchiffrant le produit des chiffrés, nous obtiendrons le produit des claires :

$$Dec_s\ (c_1c_2) = ((m_1m_2)^e)^d \bmod n \qquad = m_1m_2$$

Donc, le chiffrement de RSA réalise la propriété du chiffrement Homomorphe multiplicatif.

b) Cryptosystème TSZ "To, Safavi-Naini and Zhang's"

Proposé par Mutsunari, sakai & Kasahara, il a été créé pour le traçage des schémas en utilisant les carte Bilinéaires (Sakai, et al., 2001).

TSZ utilise un group bilinéaire tel que : $g_1 \times g_1 \rightarrow g_2$ où g_1 *et* g_2 sont des groups de premier ordre q.

Initialisation : deux générateurs arbitraires aléatoires P *et* $Q \in g_1$ et un unitaire polynomiale à coefficients dans $\mathbb{z}_q$ de degré 2K1:

$$f(x) = a_0 + a_1 x + \cdots + a_{2k-2} x^{2k-2} + x^{2k-1}$$

Clé privée : le générateur P, et le polynôme f.

La clé de chiffrement : le tuple $(g, Q_0, Q_1, \ldots, Q_{2k-2})$

Clé de l'utilisateur :

$$\boldsymbol{K_u = f(u)^{-1} P}$$

Algorithme de chiffrement : nous génèrerons un nombre aléatoire$r \in \mathbb{z}_q$, alors la clé de session $s \in g_2$ est le chiffrement en :

$$C = (sg^r, rQ, rQ_0, \ldots, rQ_{2k-2}).$$

Algorithme de déchiffrement : l'utilisateur u calcule premièrement g^r, K_u , et récupère s. en effet

$$g^r = \acute{e}(K_u, rQ_0) \times \ldots \times \acute{e}\left(u^{2k-2} K_u, rQ_{2k-2}\right) \times \acute{e}\left(u^{2k-2} K_u, rQ\right).$$

c) Le chiffrement Homomorphe d'El Gamal

Le cryptosystème d'El Gamal est une méthode de cryptographie à clé publique inventée par Taher El Gamal en 1985. Sa sécurité repose sur la difficulté de calculer le logarithme discret (EL Gamal, 1985). La génération des clés de chiffrement et de déchiffrement pour le cryptosystème d'El Gamal se fait comme suit :

Génération des clés :

- On choisit un nombre premier p et deux nombres aléatoires g et x, tel que g et x soient inférieurs à p ;
- On calcule $y = g^x \bmod p$

La clé publique est donc formée des variables y, g et p, la clé privée est x. L'algorithme d'El Gamal en détail :

Génération des clés :
- Choisir p premier, g et x aléatoires tel que (g, x<p)
- Calculer $y = g^x \bmod p$ Clé publique : pk = (g, p) Clé privée : sk = x

Chiffrement : Enc (m, pk, K)
- Choisir un entier r
- Calculer $K = g^r \bmod p$
- Calculer ensuite $c = my^r$
- le message chiffré est c'= (K, c)

Déchiffrement : Dec (c, sk, K)
- Calculer $m = cK^{-x}$

Notons que ce calcul dépend du choix de $K = g^r \bmod p$ (Voir l'algorithme ci-dessus), et donc pour un message clair donné, il y a plusieurs messages chiffrés correspondants, aussi pour chaque message chiffré il faut envoyer un second élément nécessaire au déchiffrement (K).

Supposons que nous avons c_1 et c_2 deux textes chiffrées avec l'algorithme d'El Gamal et m_1 et m_2 les textes clairs correspondants tel que :

$$c_1 = (m_1 y^r,\ g^r)\ c_2 = (m_2 y^r,\ g^r)$$
$$c_1 c_2 = (m_1 m_2 . y^{2r},\ g^{2r})$$

Le déchiffrement du produit se fait de la sorte :

$$c_1 = (m_1 y^r, g^r)$$

$$c_2 = (m_2 y^r, g^r)$$

$$c_1 c_2 = (m_1 m_2 . y^{2r}, g^{2r})$$

Nous constatons que le chiffrement d'El Gamal réalise aussi la propriété du chiffrement Homomorphe multiplicatif.

4.3.2.4. Chiffrement complètement Homomorphe

Contrairement au chiffrement partiellement homomorphe, avec le chiffrement complètement homomorphe nous pouvons réaliser tout type de calcul sur les données chiffrées stockées dans le Cloud sans les déchiffrer. L'application de ce chiffrement complètement Homomorphe constitue une brique importante dans la sécurité du Cloud. Plus généralement, nous pourrions sous-traiter des calculs sur des données confidentielles à des

serveurs situés dans Cloud tout en gardant la clé secrète qui permet de déchiffrer le résultat du calcul.

En 2014, le chiffrement homomorphe devient très prometteur : la commission européenne appelle dans son dernier appel à projet ICT à utiliser le chiffrement homomorphe dans des applications à l'horizon 2020. Le projet HEAT a réuni avec succès les chercheurs de pointe sur ce sujet en Europe (des universités de Leuven, Bristol et du Luxembourg) ainsi que des partenaires industriels spécialisés en cryptographie avancée (CryptoExperts, NXP et Thales) intéressés par le chiffrement homomorphe (Tancrède, 2014).

Définition 3.4 :

Un système de chiffrement complètement homomorphe est un cryptosystème permettant de faire des calculs sur les données chiffrées sans les déchiffrer. Formellement, si c_1 (respectivement c_2) est un chiffré de m_1 (respectivement m_2) il existe deux opérations $\square$ et $\circ$ telles que :

$$Dec\,(c_1 \square c_2) = Dec\,(c_1) \circ Dec\,(c_2) = m_1 \circ m_2$$

Le chiffrement complètement Homomorphe a été initié par Craig Gentry, ensuite DGHV une nouvelle version de son algorithme appliquée sur les entiers a vu le jour en 2010.

a) Chiffrement de Craig Gentry

La première construction d'un système complètement homomorphe a été décrite par Gentry en 2009 (Gentry, 2009), où il utilise des idéaux d'anneaux de polynômes. La sécurité de ce schéma repose sur les réseaux idéaux.

Pour chiffrer un message, l'idée est d'y ajouter du bruit, c'est-à-dire des petites erreurs. La clé secrète permet de supprimer ce bruit, à condition qu'il ne soit pas trop gros. Les opérations homomorphes qui sont effectuées impactent également ce bruit, les bruits vont grossir. Nous ne pourrons déchiffrer le message que si les bruits initiaux sont choisis très petits. Pour dépasser cette limitation sur le nombre d'opérations, et lorsque le bruit devient trop important, Gentry applique la méthode de "bootstrapping" ou d'amorçage.

Si le déchiffrement était suffisamment efficace, nous pouvions alors changer de clé publique pour réduire le bruit. Nous commencerons par utiliser une première clé, puis, quand le bruit devient trop important, nous utiliserons une seconde clé pour re-chiffrer le même message.

Le bruit ou (l'erreur) e dans un idéal I d'un anneau R est définit par : $e = kI \in I \subset R$. Le message est alors chiffré en ajoutant ce bruit au message,

$$Enc(m) = c = m + kI$$

La procédure de déchiffrement consiste à retirer l'erreur. Les propriétés homomorphes du système sont réalisées, pour :

$$c_1 = m_1 + k_1 I$$

et

$$c_2 = m_2 + k_2 I$$

On a:

$$c_1 + c_2 = m_1 + m_2 + (k_1 + k_2)I$$

Et

$$c_1 \times c_2 = m_1 \times m_2 + (m_1k_2 + m_2k_1 + k_1k_2)I$$

On peut déjà remarquer que le bruit est beaucoup plus affecté par une multiplication que par une addition. Approximativement, une addition double le bruit alors qu'une multiplication l'élève au carré. Si un trop grand nombre d'opérations est effectué, le bruit devient trop grand et la procédure de déchiffrement retourne un message erroné. Cependant, en évaluant régulièrement la procédure de déchiffrement de manière homomorphe, nous peuvons éviter que cela arrive, et c'est exactement ce que fait le bootstrapping :

Étant donné un chiffré c de m, cette procédure retourne un chiffré c' de m où le bruit k' contenu dans c' est plus petit que le bruit k contenu dans c : $\|K'\| < \|K\|$.

Cependant, pour pouvoir évaluer la fonction de déchiffrement de façon homomorphe, il est nécessaire que celle-ci soit suffisamment simple, ce qui n'est pas le cas initialement. Pour faire face à ce problème, Gentry réduit la complexité du circuit de déchiffrement en publiant un ensemble de vecteurs dont la somme d'une partie d'entre eux est égale à la clé secrète. Ce problème est connu sous le nom de "Sparse" Subset Sum Problem, et est prouvé NP-complet.

L'idée de Gentry est de partir d'un schéma dit "somewhat homomorphic encryption scheme" qui peut évaluer des additions et des multiplications tant que le bruit n'est pas trop grand, et de lui appliquer la procédure de bootstrap. Le schéma initial est basé sur le "Ideal Coset Problem". Cependant, pour appliquer la procédure de bootstrap en réduisant la complexité du circuit de déchiffrement il se base sur le "Sparse Subset Sum Problem".

b) Algorithme de DGHV

L'algorithme de (Van Dijk, et al., 2010) présente un schéma complètement homomorphe (DGHV) qui est une application du chiffrement de Gentry sur les entiers et dont la sécurité repose sur le problème du diviseur commun approché. Les étapes de l'algorithme DGHV sont :

Génération des clés : r, p et q, $r \sim 2^n$, $p \sim 2^{n^2}$, $q \sim 2^{n^5}$, p premier
La clé privée : p
$Enc_{sk}(m) = pq + 2r + m = c$
$Dec_{sk}(c) = (pq + 2r + m \bmod p) \bmod 2$

Exactitude : $pq >>> 2r + m$
Ainsi : $c \bmod p = 2r + m$
Donc : $(c \bmod p) \bmod 2 = (2r + m) \bmod 2 = m$

Pour deux messages m_1 et m_2, soient c_1 et c_2 leurs chiffrés respectivement :

$$c_1 + c_2 = (q_1 + q_2)\ p + 2(r_1 + r_2) + m_1 + m_2$$

Donc si :

$$2(r_1 + r_2) + m_1 + m_2 \text{ n } p$$

Alors :

$$((c_1 + c_2) \bmod p) \bmod 2 = [2(r_1 + r_2) + m_1 + m_2] \bmod 2$$
$$= m_1 + m_2$$

Ainsi, DGHV réalise la propriété du chiffrement homomorphe additif.

$$c_1 \text{ x } c_2 = [q_1 q_2 p + (2r_1 + m_1) + (2r_2 + m_2)]\ p + 2(2r_1 r_2 + r_1 m_1 + r_2 m_1) + m_1 m_2$$

Donc si :

$$2(2r_1 r_2 + r_1 m_1 + r_2 m_1) + m_1 m_2 \text{ n } p$$

Alors :

$$((c_1 \text{ x } c_2) \bmod p) \bmod 2 = [2(2r_1 r_2 + r_1 m_1 + r_2 m_1) + m_1 m_2] \bmod 2$$
$$= m_1 m_2$$

De ce fait, DGHV réalise aussi la propriété du chiffrement homomorphe Multiplicatif.

Un chiffrement qui n'est pas complètement Homomorphe, est forcément partiellement Homomorphe, dans la mesure où il peut permettre la réalisation que d'une seule opération à la fois sur des chiffrés tel que (l'addition ou la multiplication), ou bien il accepte plus d'une opération mais avec un nombre limité d'itérations (pas plus d'une seule addition ou multiplication ou pas plus d'un bit).

4.3.2.5. Chiffrement partiellement Homomorphe

Après avoir détaillé dans les sections précédentes les algorithmes homomorphes : RSA, El Gamal, Paillier, Goldwasser-Micalli, Gentry et DGHV, nous allons détailler dans ce qui suit un autre algorithme qui fait partie des algorithmes partiellement homomorphes : Okamot-Uchiyama (Okamoto & Uchiyama, 1998) homomorphe pour l'addition. Cet algorithme se déroule comme suit :

Génération des clés :

- Choisir p et q deux nombres premiers de k bits ;
- Calculer $N = p^2 q$;
- Choisir $g \in Z_N^*$ Au hasard tel que :

  $g^p \mod p^2 \neq 1$

  La clé publique : (N,g,h,k)

  La clé privée : (p,q)

  Chiffrement : pour $m \in \{1, \ldots, 2^{k \neq 1} \neq 1\}$

- Choisir : $r \in Z_N^*$ au hasard ;
- Calculer : $c = g^m h^r \mod N$

Déchiffrement :

$$m = \frac{c^{p-1} mod\ p^2}{g^{p-1} mod\ p^2} mod\ p$$

4.3.3. Récapitulé sur Algorithmes Homomorphes

Les différents algorithmes de chiffrement partiellement ou complètement homomorphe sont toujours loin d'être applicables, et cela du fait de leurs facteurs d'expansion et de la taille du chiffré généré après chaque multiplication ou addition.

Dans ce qui suit, nous allons présenter un tableau récapitulatif des algorithmes de chiffrement, afin de choisir les algorithmes les plus adaptés. La contribution (Fahsi, et al., 2015) nous donnera une idée sur la taille de la clé générée et le temps d'exécution.

Dans le tableau suivant, NA veut dire not announced.

Système Homomorphe	Opération(s) sur les clairs	Opération(s) Sur* les chiffrés	Complexité			Avantages	Inconvénients
			PKG	SKG	Chiff.		
RSA, 1978	Multiplicative	Multiplicative	N^6	N^3	$O(N^3)$	premier cryptosystème homomorphe ; factorisation	Chiffrement déterministe
El Gamal, 1983	Multiplicative	Multiplicative	N^7	N^3	$N^{1.5}$	Expansion du message (x2)	Chaque message à chiffrer est composé d'un couple (c1, c2)
Goldwasser-Micalli, 1983	Multiplicative	Somme direct $\oplus$	NA	NA	NA	résidualité quadratique (plus fort que la factorisation)	Expansion du message (xN)
Benaloh, 1994	Multiplicative	Additive	NA	NA	NA	résidualité quadratique d'ordre supérieure	Temps du déchiffrement long quand Nr est grand, expansion du message (x r) surtout si r est petit
Okamot-Uchiyama, 1998	Multiplicative	Additive	NA	NA	NA	factorisation	Le chiffré est élevé à p ≠ 1 le paramètre k doit être choisit selon la taille du message à chiffrer
Paillier, 1999	Multiplicative	Additive	NA	NA	NA	résidualité composée (plus forte que la factorisation)	Expansion du message est élevée à la puissance m
Sander-Young-Yung, 1999	Multiplication direct $\otimes$	Multiplicative	NA	NA	NA	résidualité quadratique (plus fort que la factorisation)	Expansion (x_l.N)
Gentry, 2009	Additive, Multiplicative	Additive, Multiplicative	N^7	N^3	$N^{1.5}$	premier cryptosystème complètement homomorphe	pas applicable vu la taille des paramètres, pour l'addition le bruit est (x2), pour la multiplication il est élevé au carré
DGHV, 2010	Additive, Multiplicative	Additive, Multiplicative	$\boldsymbol{O(\lambda 10)}$	$\boldsymbol{O(\lambda 2)}$	$\boldsymbol{O(\lambda 5)}$	version améliorée, appliquée aux entiers	temps de chiffrement déchiffrement reste très grand

Tableau 4-1 : Étude comparative des cryptosystèmes Homomorphes.

4.3.4. Données médicales et Cloud computing

Les données sur la santé sont des données très sensibles qui ne devraient pas être mis à la disposition de personnes non autorisées afin de protéger la sécurité des informations des patients. Cependant, les nouvelles technologies comme le cloud computing présente des problèmes de sécurité et de confidentialité qui accompagne un besoin croissant de stockage et de transfert d'informations de santé afin d'améliorer la qualité des soins et d'accroître l'efficacité et l'organisation des services de santé (Zhang & Liu, 2010), (Mehraeen, et al., 2016). Ces informations doivent être accessibles aux fournisseurs de données médicales autorisés, y compris les chercheurs qui tentent de trouver les causes, les traitements et les patients (Alnuem, et al., 2011). De plus, les organisations de santé qui disposent d'un volume important de données nécessitent des outils de calcul puissants pour les traitées. Les physiciens et chimistes doivent aussi avoir accès à ces informations médicales pour fournir un traitement complet et précis aux patients (Lupse, et al., 2012). Ce qui rend la médecine un domaine gourment en termes de support de stockage, traitement ou de personnels (Rostrom & Teng, 2011).

Les développements actuels dans le système répartis de gestion de santé ont été influencés par le développement de l'industrie des technologies de l'information. Le but étant d'offrir une plate-forme de partage des systèmes d'information médicale, des infrastructures de stockage et des applications de traitement. Une prise en charge de la sécurité des communications et de la confidentialité des informations de santé accroîtront la confiance des utilisateurs dans de tels systèmes de télésanté (Khana, et al., 2014). En plus, l'utilisation de données cliniques avec des algorithmes d'exploration de données et/ou de texte (data/text Mining) nécessite des ressources dynamiques et évolutives.

Le cloud computing est la solution idéale pour répondre à ces exigences (Griebel, et al., 2015)du fait que c'est un ensemble de services informatiques loués, fournis à un client et accessible par le biais d'un réseau avec la possibilité d'étendre ou de réduire leurs besoins (Kuyoro, et al., 2011). Il s'agit d'une technologie basée sur le principe «pay on demand» (Bildosola, et al., 2015). Malgré tous les avantages du cloud computing, il existe plusieurs défis qui retardent la migration des applications et des données vers le cloud (Itani, et al., 2009). Le renforcement des mécanismes de contrôle d'accès et la confidentialité des données sont parmi les défis les plus importants qui doivent être pris en considération dans le développement d'une plateforme cloud computing (Neisse, et al., 2015).

Dans le cas d'un cloud de santé, un grand nombre d'ordinateurs et de serveurs est dédié pour répondre aux besoins médicaux. Les services du cloud de santé peuvent être accessibles aux utilisateurs (patients ou médecins) par le biais d'une connexion Internet (Parekh & Saleena, 2015), (Bildosola, et al., 2015). Dans ce cas, la sécurité est un obstacle à l'adoption d'une éventuelle externalisation de données de santé sur un cloud, car cela nécessite un grand niveau d'intégration, d'interopérabilité et de partage de données entre différents médecins et organisations sanitaires. Ceci implique que différents besoins seront exprimés à travers des requêtes de recherche de contenus pertinents à des domaines spécifiques et des groupes

d'intérêts indépendants entre eux. Cette indépendance implique une confidentialité de données mais aussi une confidentialité des requêtes et des résultats de recherche sur la base de données dans le cloud de santé (Alnuem, et al., 2011) (Kuyoro, et al., 2011). Dans la section suivante, nous allons présenter un état de l'art des travaux qui portent sur la confidentialité dans le contexte des données médicales.

4.4 Confidentialité dans le Cloud de Données Médicales : État de l'art

Les données stockées dans un environnement virtualisé peuvent être consultées ou gérées par un grand nombre de personnes (Velumadhava & Selvamani, 2015) tandis que les patients perdent le contrôle physique sur leurs informations personnelles (Li, et al., 2010). Ainsi, l'utilisation du cloud computing dans le domaine de santé a plusieurs problèmes et préoccupations majeurs, y compris la transmission de requêtes et résultats de recherche, la confidentialité ou le contrôle d'accès (Balasubramaniam & Kavitha, 2015). D'autre part, l'intégrité des données constitue une tâche difficile (Azhar & Laxman, 2014). Cependant, en stockant les informations sur la santé dans le Cloud.

Dans ce qui suit, un recueil des approches relatives aux problèmes de sécurité liés au cloud de données médicales est présenté en mettant l'accent sur l'aspect de sécurité accomplis de chaque travail. Nous classifions ainsi les articles par rapport aux questions de recherche suivantes :

(A) Quel aspect de sécurité du cloud de données médicale est concerné par le papier ? confidentialité de stockage et de transmission, control d'accès, disponibilité ou intégrité.

(B) Comment le cloud de données médicale implémente le mécanisme de sécurité ?

Dans (A), l'externalisation et la transmission des données d'une organisation à un autre est un acte inquiétant qui exige une compréhension des risques associés (Kuyoro, et al., 2011). D'après (Rahman & al., 2015), il n'existe actuellement aucune solution complète, anonyme et sécurisée d'échange de données dans un environnement cloud de données médicales

Une autre préoccupation des ressources informatiques partagées dans les infrastructures cloud est l'identité et le contrôle d'accès accompli par l'authentification. L'authentification par des codes PIN remplace actuellement l'authentification à base des techniques biométriques ou d'identifiants des utilisateurs (Saevanee, et al., 2015). Dans le processus d'authentification classique, il peut y avoir une utilisation illégale de données si le mot de passe est divulgué à une personne non autorisée par conséquent, les méthodes d'identification et d'authentification actuelles dans les organismes de santé peuvent ne pas être applicables dans le Cloud Médical, car elles présenteront une faille de sécurité (Gunamalai & Sivasubramanian, 2015).

L'accès distant via Internet est un autre défi important dans le cloud Médical. Les nuages sont sur Internet. Par conséquent, tous les problèmes de sécurité liés à Internet, y compris les fraudes et les attaques de pirates, peuvent se produire (Cheng & Lai, 2012). Lorsque

plusieurs organisations partagent des ressources sur un environnement Internet, il existe une menace d'utilisation abusive des données ce qui affecte la disponibilité (Velumadhava & Selvamani, 2015).

Chen (Chen, 2012) propose un schéma de contrôle d'accès fiable à base du polynôme d'interpolation de Lagrange pour établir un système sécurisé et efficace d'accès multi-utilisateurs à l'information dans un cloud médical. D'autres approches utilisent l'identité de l'utilisateur, le chiffrement multi-biométrique et l'audit de données pour vérifier l'intégrité des données médicales (Aslam, et al., 2014), (Vidya, et al., 2012). L'identité de l'utilisateur doit être vérifiée avant toute autorisation d'accès utilisant le nom d'utilisateur et le mot de passe attribués par les fournisseurs de services cloud. Les patients disposent de droits sur leurs données, cela leurs permettent d'accepter ou de refuser le partage d'information avec d'autres médecins ou organismes médicales. Ainsi, les organismes de santé devraient appliquer un ensemble de contraintes de sécurité et de contrôle d'accès qui garantissent l'intégrité, la confidentialité et la confidentialité des données médicales dans l'environnement cloud. Les administrateurs de l'infrastructure du cloud médical, les médecins et les patients sont donc associés à différents niveaux de privilèges et de permissions pour sécuriser l'accès et permettre la récupération des informations (Youssef, 2014).
La confidentialité est l'un des aspects les plus importants dans le domaine de cloud médicale. La protection de la vie privée et de la confidentialité de l'information médicale est un processus continu assuré par les organisations médicale qui assume la responsabilité juridique et morale (Haufe, et al., 2014). Dans un Cloud de stockage de données médical, la mise en œuvre d'un protocole PIR à base chiffrement homomorphe permet au cloud d'exécuter des calculs sur les données chiffrées au nom du patient ou du staff médical. Ainsi, le fournisseur de cloud ne peut pas connaitre le contenu des mises à jour patientes, des alertes ou des recommandations envoyées qu'après les traitements basés sur les données reçues. Les fonctions à calculer dans ce scénario peuvent inclure des moyennes, des écart-types ou d'autres fonctions statistiques telles que la régression logistique qui peut aider à prédire de certaines situations sanitaires dangereuses. Dans ce contexte, nous pouvons citer les travaux récents de (Yongge, 2016) et de (Yagisawa, 2016), qui optimisent la complexité des algorithmes de chiffrement et n'utilise pas le bruit qui affecte la taille des bases de données chiffrées, tous comme (Li & Wang, 2015). Contrairement aux approches homomorphe classiques, où le cryptosystème soufre de problème de temps d'exécution, de la consommation de la bande passante comme l'indique les travaux sur la complexité des algorithmes homomorphique, tel que (Revathy & Gopu, 2016), (EL-YAHYAOUI & ELKETTANI, 2016), et (Gjosteen & Strand, 2016).

4.4.1. Récapitulé sur Algorithmes Homomorphes dans les Clouds Médicaux

Le tableau 4-2 Récapitule les algorithmes Homomorphes dans les Clouds Médicaux:

No	Chercheur(s)	Année	Protocole Utilisé	Propriétés
1	Yongge Wang	2016	Fully homomorphic, Noise-free(no Bootstrapping)	Symétrique Complexité réduite par l'algorithme utilisé
2	M. Yagisawa	2016	Fully homomorphic, Noise-free(no Bootstrapping)	
3	Fahsi and al.	2015	Homomorphic With noise	Symétrique Complexité réduite
4	Li Wang	2015	Fully homomorphic, Noise-free(no Bootstrapping)	Symétrique
5	Gunamalai, and Sivasubramanian	2015	Column based encryption+ ACL	Asymétrique
6	Rahman et al.	2015	Pairing-based cryptography	Symétrique
7	Haufe et al.	2014	ISO 270xx,	Pas de chiffrement
8	Aslam Khan et al.	2014	AES	Symétrique
9	Azhar and Laxman	2014	Homomorphic IBE	Asymétrique
10	Li, M., Yu, S., Zheng, Y., & Member	2013	Homomorphic ABE ACL	Asymétrique
11	Jaswanthi and NaliniSri	2013	Hybrid Execution model	Asymétrique
12	Chen et al.	2012	Lagrange interpolation+ Access Control	Asymétrique
13	Vidya et al.	2012	Homomorphe General Model	Asymétrique
14	Cheng and Lai	2012	Healthcare cloud legal protection	Asymétrique
15	Li. et al.	2010	Homomorphic ABE	Asymétrique

Tableau 4-2 : état de l'art du chiffrement homomorphique dans les Cloud de santé

4.4.2. Position de Notre Contribution

Dans notre approche, nous vison la réduction du coût des traitements et l'optimisation de la bande passante des transferts, en particulier le coût de communication, lors d'une recherche de contenu dans un Cloud. Cette exigence stricte s'oppose à la nature des protocoles de chiffrement homomorphes en raison de la lenteur du chiffrement et de communication (Lauter, et al., 2011). D'autre part, l'adoption des techniques de chiffrement symétriques classiques n'est pas non plus un choix viable, car elle empêche essentiellement les services du Cloud d'effectuer un traitement sur les données chiffrées, sans parler de la reconstruction des données. Dans ce travail de thèse, nous proposons une amélioration des schémas de chiffrement homomorphes existants par augmentation des performances des algorithmes existants, ce qui encourage leur adoption dans le contexte du Cloud, en particulier médical.

4.5 Conclusion

Les systèmes de chiffrement homomorphes sont d'une grande importance. Ils offrent la possibilité de traiter des données en tout anonymat en respectant ainsi la vie privée à l'égard des utilisateurs propriétaires de ces données.

De nombreux cryptosystèmes homomorphes ont des applications uniquement en théorie mais en pratique dès qu'on veut effectuer un calcul surtout dans le cas de la multiplication, la taille du produit des chiffrés explosent, surtout dans le cas du chiffrement complètement homomorphe qui demande encore des optimisations au niveau des paramètres, chose qui rend son application impossible jusqu'au jour d'aujourd'hui.

Dans ce chapitre nous avons détaillé les algorithmes de chiffrement simplement et complètement homomorphes. Nous avons étudié leurs avantages et inconvénients aussi la durée de chiffrement. Nous avons aussi cité les travaux récents qui couvrent le problème de confidentialité des Clouds Médicaux avec un positionnement de notre approche par rapport aux travaux connexes.

Chapitre V.

Optimisation de la recherche d'information : Protocole de Retrait Privé PIR Homomorphe pour le Cloud Médical

Chapitre V. Optimisation de la recherche d'information : Protocole de Retrait Privé Homomorphe pour le Cloud Médical

5.1 Introduction

À l'égard de l'expansion phénoménale des besoins des entreprises en capacité de stockage et de traitement ainsi que l'importance de la disponibilité freinée par les pannes matériels/logiciels récurrentes et le manque de staff technique compétent, les institutions médicales publiques comme privées se voient obligées d'opté vers une solution cloud. Cette dernière propose plusieurs produits dans le plus approprié est la solution Software as a Service SAAS qui permet à son acquéreur d'implémenter ces propres méthodes afin de gérer ces données. Mais la plateforme SAAS présente un inconvenant majeur : le manque de sécurité et de confidentialité des données et de communications (Bildosola, et al., 2015).

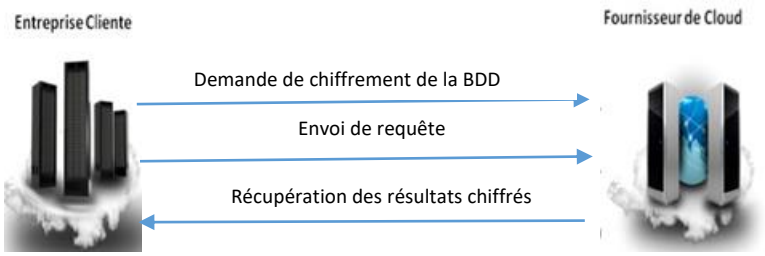

Figure 5-1 : Chiffrement Homomorphe.

Nous proposons dans ce travail une approche qui assure la confidentialité des communications à l'aide des algorithmes de chiffrement homomorphe, pour protéger les requêtes exécutées sur des données entreposées dans les serveurs cloud loués comme SaaS par des entités hospitalières. Parmi les entreprises clientes qui peuvent aussi adopter ce type d'utilisation de Cloud, nous trouverons tout établissement capable de déléguer le traitement et stockage à des serveurs distants (Cloud Computing). La Figure 5.1 ci-dessus éclaircit l'échange homomorphe des données entre le client et le Cloud.

Après avoir étudié la sécurité de recherche d'informations en particulier PIR (Private Information Retrieval), nous allons consacrer ce chapitre à la description de l'architecture de notre Framework, son principe de fonctionnement et les outils exploités pour son développement.

5.2 Problématique liée à la sensibilité des Données médicales et Cloud computing

Comme discuté auparavant, l'adoption de services de cloud par les consommateurs et les entreprises est limitée par les préoccupations concernant la perte de la confidentialité de leurs données ce qui affect leur vie privée ou la valeur commerciale de leurs données privées. Dans cette section, nous présentons quelques applications concrètes du chiffrement homomorphique dans le contexte de retrait d'information privé PIR, dans les secteurs médicaux et financiers. Ces applications peuvent renforcer la préservation des données des clients durant le processus d'externalisation ou de traitement Cloud.

Les protocoles homomorphiques préservent la vie privée d'applications dans le contexte du Cloud. Ils permettent l'implémentation de protocoles de retrait privé donc des recherches sur de données externalisée telle que le client de cloud envoi une requête chiffrée et le serveur Cloud répond par des résultats chiffrés sans jamais accéder à la requête en clair.

Premièrement, dans l'industrie et le finance, il y a un scénario d'application potentiel dans lequel les données ainsi que la fonction à calculer sur les données sont privées. Exemple de fonctions sur des données des sociétés ; calcule des prix des articles en stock ou leur inventaire qui sont souvent pertinents dans la prise des décisions d'investissement. Avec les fonctions homomorphiques du protocole PIR, quelques fonctions peuvent être évaluées en privé comme suit. L'utilisateur charge une version chiffrée de la fonction dans le cloud. Ainsi, le flux de données est chiffré avec la clé publique de l'utilisateur puis chargé dans le Cloud. Le cloud ensuite évalue la fonction privée en appliquant une description chiffrée du programme aux requêtes chiffrées qu'il reçoit. Après le traitement, le cloud répond avec un résultat chiffré au client (Fahsi, et al., 2015).

Deuxièmement, dans un Cloud de stockage de données médicales, la mise en œuvre d'un protocole PIR à base chiffrement homomorphe permet au cloud d'exécuter des calculs sur les données chiffrées au nom du patient ou du staff médical. Ainsi, le fournisseur de cloud ne peut pas connaitre le contenu des mises à jour patientes, des alertes ou des recommandations envoyées après les traitements basés sur les données reçues. Les fonctions à calculer dans ce scénario peuvent inclure des moyennes, des écart-types ou d'autres fonctions statistiques telles que la régression logistique qui peut aider à prédire de certaines situations sanitaires dangereuses.

5.3 Contribution

L'application du chiffrement homomorphe constitue une brique importante dans la sécurité du Cloud Computing. Grâce à ce type de chiffrement, nous pourrions sous-traiter des calculs sur des données confidentielles à des serveurs situés dans le Cloud en gardant la clé secrète qui permet de déchiffrer le résultat du calcul. Néanmoins, l'idée de chiffrer une base de données pour chaque requête reste loin d'être pratique. Ainsi, notre première contribution consiste à mettre en place un Framework qui régit le protocole de chiffrement homomorphique basé non pas sur la requête tous court pour le chiffrement, mais sur les identifiants de l'octroi de service Cloud associé à chaque utilisateur. Cela nous permettra de chiffrer la base de données médicale à chaque fois les attributs d'authentification modifiés (mots de passe, email, Téléphone, etc.). Ce qui réduit énormément la charge des serveurs du fournisseur CSP qui vont associer le chiffrement de la base entière à la politique de sécurité de l'authentification. Cette dernière recommande un changement périodique

Ainsi, nous avons constaté un besoin d'étude des performances des trois classes des cryptosystèmes Homomorphes par rapport à :

- La taille de la clé de chiffrement et son impact sur la durée du chiffrement ;
- Le temps de traitement de la requête par le serveur en fonction de la taille des messages chiffrés et la taille de la clé ;
- Le délai de déchiffrement du résultat de la requête en fonction de la taille de la réponse envoyée par le serveur.

D'où l'idée de notre contribution qui vise l'implémentation et l'évaluation de quelques protocoles de chiffrement homomorphe pour évaluer les performances des schémas de chiffrement, toute en optimisant les résultats par la proposition d'un Framework qui fait appel aux politique d'authentification pour éviter le chiffrement des données externalisées à chaque réception de requête.

5.4 Principe de Fonctionnement

Après analyse approfondie des fonctions de chiffrement homomorphe, nous concluons que c'est un échange de données qui assure la confidentialité des informations et des communications. Cependant, notre contribution porte uniquement sur la confidentialité des communications. Cela veut dire que les données de l'entreprise cliente (dans notre cas un hôpital ou groupe d'hôpitaux) sont stockées en claire au niveau du cloud toute en étant sécurisé par les normes de sécurité ainsi que les conventions de confidentialité signées entre l'entreprise cliente et le fournisseur de service cloud.

Ainsi, les données de l'entreprise seront chiffrées lors de l'invocation de la base de données en vue d'exécuter une requête client (patient ou membre de l'entreprise propriétaire du cloud).

Dans ce cas, une application implémentée par l'entreprise (implémentée dans la plateforme SaaS) chiffre le contenue de la base de donnée en utilisant la même clé publique qui a servi à crypter la requête. Un autre module de la plateforme fera l'appariement (matching) base de données-requête afin de retourner les informations relatives aux besoins exprimés par l'utilisateur du système. Reçues au niveau patient ou membre de l'entreprise, les informations cryptées récupérées sur les serveurs cloud seront décryptées et seul le propriétaire peut les visualiser. Dans cette fonction génératrice des clés et responsable du chiffrement et déchiffrement s'exécute sur les ordinateurs de l'entreprise).

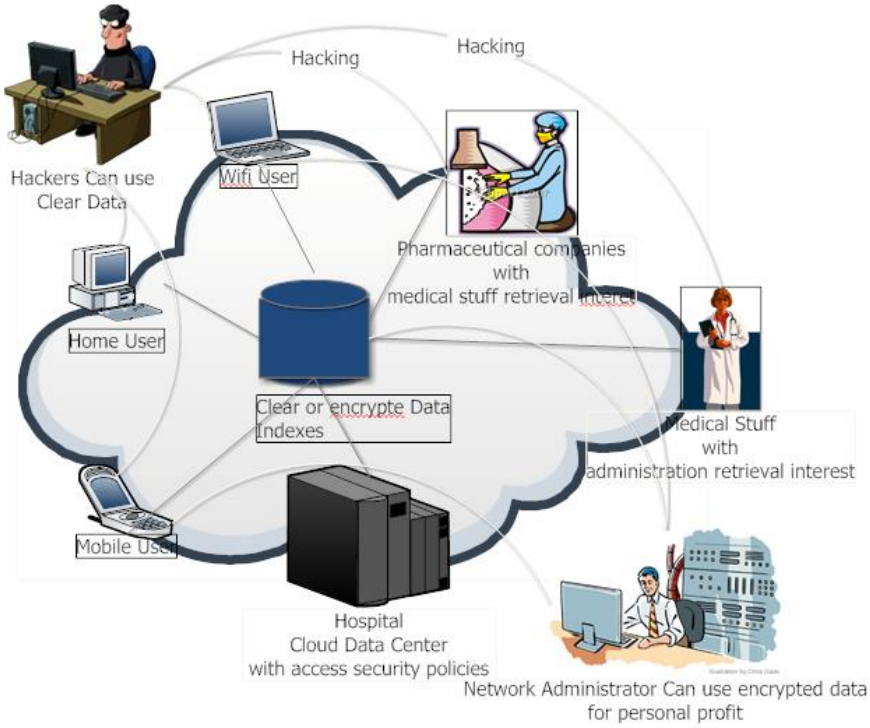

Figure 5-2 : Utilisation du cloud Médical

Ce travail consiste à réaliser une simulation d'un modèle de retrait privé de l'information (PIR) avec plusieurs algorithmes de la littérature. Cette simulation considère un mécanisme de recherche d'information selon le contenu des documents stockés et non pas un appariement SQL exacte. Pour cela, nous utilisons le modèle Sac-à-mots et l'indexation TF-IDF pour mieux représenter l'espace de recherche.

5.5 Description de l'Architecture

Les algorithmes de chiffrement homomorphique ainsi que les méthodes d'indexation de document et recherche sont implémentés en utilisant quatre (03) Modules principaux :

- Module d'authentification : qui permet aux utilisateurs de fournir leurs noms d'utilisateurs ainsi que leurs mots de passe. Ces derniers sont utilisés pour la génération de clés.
- Module de stockage : qui permet l'envoi et le stockage des documents utilisateurs sous forme d'indexes non chiffrés.
- Module de recherche : qui permet de chiffrer la requête utilisateur et la base des indexes en claire, de faire l'appariement, d'envoyer les résultats chiffrés et de faire le décryptage au niveau de l'utilisateur.

Ces modules interagissent suivant les étapes de l'algorithme de fonctionnement général (Algorithme 5-1).

Algorithme 5-1 : Algorithme de fonctionnement général

Coté Client

Input : mot de passe utilisateur (change périodiquement)
Générer la clé privée et la clé publique
Pour chaque utilisateur
Pour chaque requête de recherche
Pour chaque mot clé de la requête
Chiffrer le mot clé utilisateur avec la clé publique,
Envoyer la requête chiffrée et la clé publique au Cloud

Coté Cloud

Input : Documents et rapports médicaux**,** la requête chiffrée et la clé publique
Pour chaque document de la base de données médicale
Indexer le document avec le Model TF/IDF
Chiffrer les mots clés et informations (titre, emplacement… etc.) de chaque document avec la clé publique
Comparer la requête chiffrée et la base de données médicale chiffrée
Envoyer les résultats de recherche au client

Coté Client

Input : Résultat de Recherche, Clé privée
Pour chaque mot clé des résultats de recherche,
Déchiffrer les mots clés et information du document (titre, emplacement… etc.)

Par la suite, un recueil des données d'évaluation sera effectué afin de mesurer l'efficacité de la recherche en utilisant la Précision et le Rappel ainsi que le temps de chiffrement et de déchiffrement des résultats.

Ces modules ainsi que les données d'évaluations seront mieux détaillés dans les sections suivantes.

5.5.1. Module d'authentification

L'authentification est une étape cruciale à vérifier avant de parler de confidentialité des données.

Les figures 5.3 et 5.4 illustrent les détails du Framework proposé. Dans notre modèle, il s'agit d'un mécanisme classique d'authentification et d'inscription avec un chiffrement unidirectionnel des mots de passe en utilisant le hachage MD5.

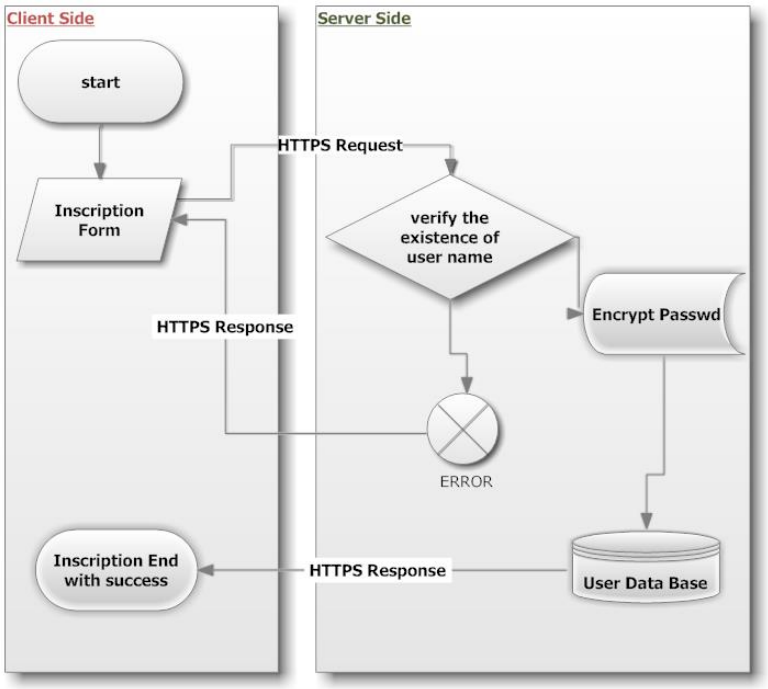

Figure 5-3 : étape d'inscription pour les utilisateurs non enregistrés

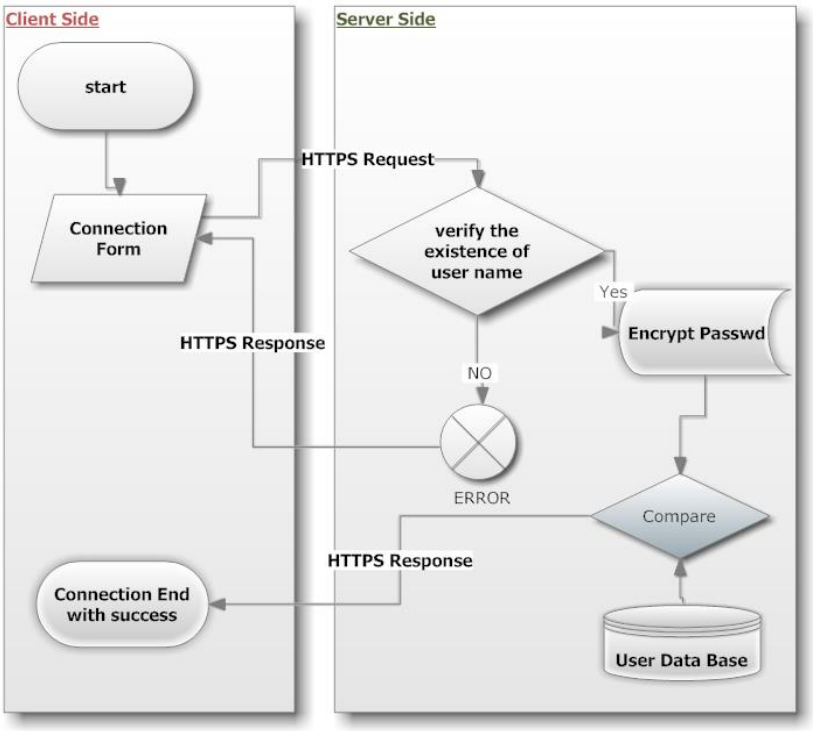

Figure 5-4 : étape d'authentification pour les utilisateurs enregistrés

5.5.2. Module de stockage

Ce module permet à l'utilisateur authentifié du système (patient ou staff médical) d'ouvrir une session. Après s'être authentifié, un utilisateur peut sélectionner les documents qu'il veut ajouter dans l'espace de stockage dédié. Ces documents doivent être de contenus divers (images, rapports médicales, ordonnances, etc.). Cependant, le prétraitement touchera seulement les documents textuels écrits en anglais de différents types (Word, PDF, texte brute, etc. transformés en fichiers text). L'opération d'ajout de documents dans l'espace Cloud de l'établissement hospitalier est appelée chargement est effectuée dans la partie client, et elle est suivie directement par une opération d'indexation de contenu effectuée dans ce cas par les méthodes implémentées au niveau du cloud.

La figure 5.6 explique les mécanismes de sécurité comme le protocole HTTPS ou SSH lors de la phase chargement des données médicales depuis le poste client jusqu'au cloud.

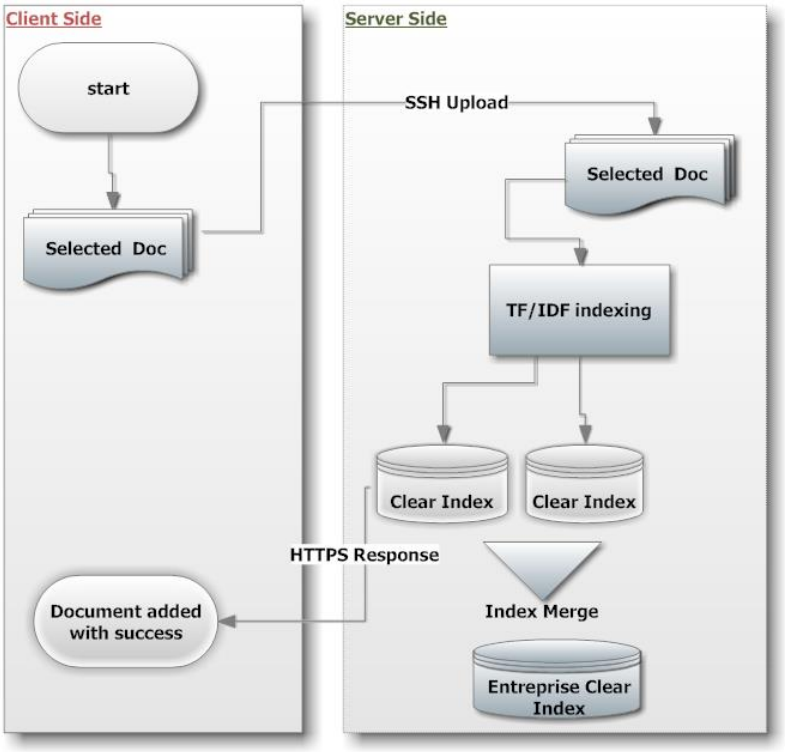

Figure 5-5 : chargement et indexation des données médicales.

Dans la figure 5-5, l'indexation adoptée consiste à faire les étapes suivantes :

- Tokenisation : ou décomposition de texte en tokens ou mots
- Suppression des mots vides : comparer les tokens de l'étape précédente a une liste de mots clé de la langue anglaise.
- Stemming : suppression des préfixes et des suffixes en utilisant l'algorithme porter stemmer.
- Calcul de TF-IDF : cette pondération est basée sur les mesures TF ou term frequency qui est une pondération locale qui mesure le poids d'un mot dans un document, et IDF ou Inverted Document Frequency qui est le poids d'un document par rapport à la collection.

5.5.3. Module de recherche

Le module de recherche de la figure 5-6 se charge de gérer les méthodes relatives à la recherche d'un besoin exprimé par un utilisateur du système sous forme de requête. Nous pouvons dire que c'est le noyau du système car il représente la partie qui assure le but de sécurité à achever, qui est la confidentialité de la communication ainsi que la confidentialité du besoin. Pour atteindre ce but, nous avons utilisé les algorithmes de cryptographie homomorphiques présentés dans les chapitres précédents.

Les protocoles de chiffrement homomorphiques sont des protocoles asymétriques, de ce fait le client du service cloud et le cloud lui-même ne vont pas utiliser les mêmes clés de chiffrement. En conséquence, nous devons implémenter les méthodes du protocole de chiffrement homomorphique comme suit :

- **Méthode de génération de clés :** implémentée au niveau client, elle permet de générer deux clés :
    - **Une clé publique :** utilisée localement pour chiffrer la requête, et utilisée par la méthode chiffrement du cloud pour chiffrer la base des indexes.
    - **Une clé privée :** utilisée par le client pour déchiffrer les résultats retournés par le cloud après l'appariement requête chiffrée et base d'indexes chiffrée.

  La fonction de génération des clés dépond de l'algorithme de chiffrement homomorphique choisi. Pour éviter toutes attaques cryptanalytique, les deux clés publique et privée sont régénérées à chaque fois que l'utilisateur change son mot de passe pour prévenir un décryptage des résultats et aussi pour éviter la surcharge des serveurs du cloud lors du cryptage de la base de données.
- **Méthode de chiffrement :** cette méthode sera implémentée au niveau du client et au niveau du cloud SaaS. Elle prend en entrée le nom de l'algorithme a utilisée et la clé publique respectif. Comme sortie, nous obtiendrons une requête chiffrée au niveau client, et une base d'indexes chiffrée au niveau du cloud.
- **Méthode d'appariement :** inspirée du model vectoriel, elle se charge de calculer la distance entre le vecteur requête chiffrée et chaque tuple de la base de données chiffrée. La distance utilisée est le cosinus des angles entre les vecteurs respectifs. Ensuite les documents seront classés par ordre décroissant et envoyés au client pour être déchiffrés.
- **Méthode de déchiffrement :** implémentée au niveau du client, elle permet de déchiffrer les résultats de la méthode de l'appariement. Cette méthode utilise la clé privée pour effectuer cette tâche.
- **Méthode de récupération :** permet à l'utilisateur de récupérer les documents qu'il souhaite visualiser.

L'ensemble des méthodes précédentes s'exécutent après la sélection d'un algorithme de chiffrement homomorphique parmi les algorithmes détaillés dans la section suivante.

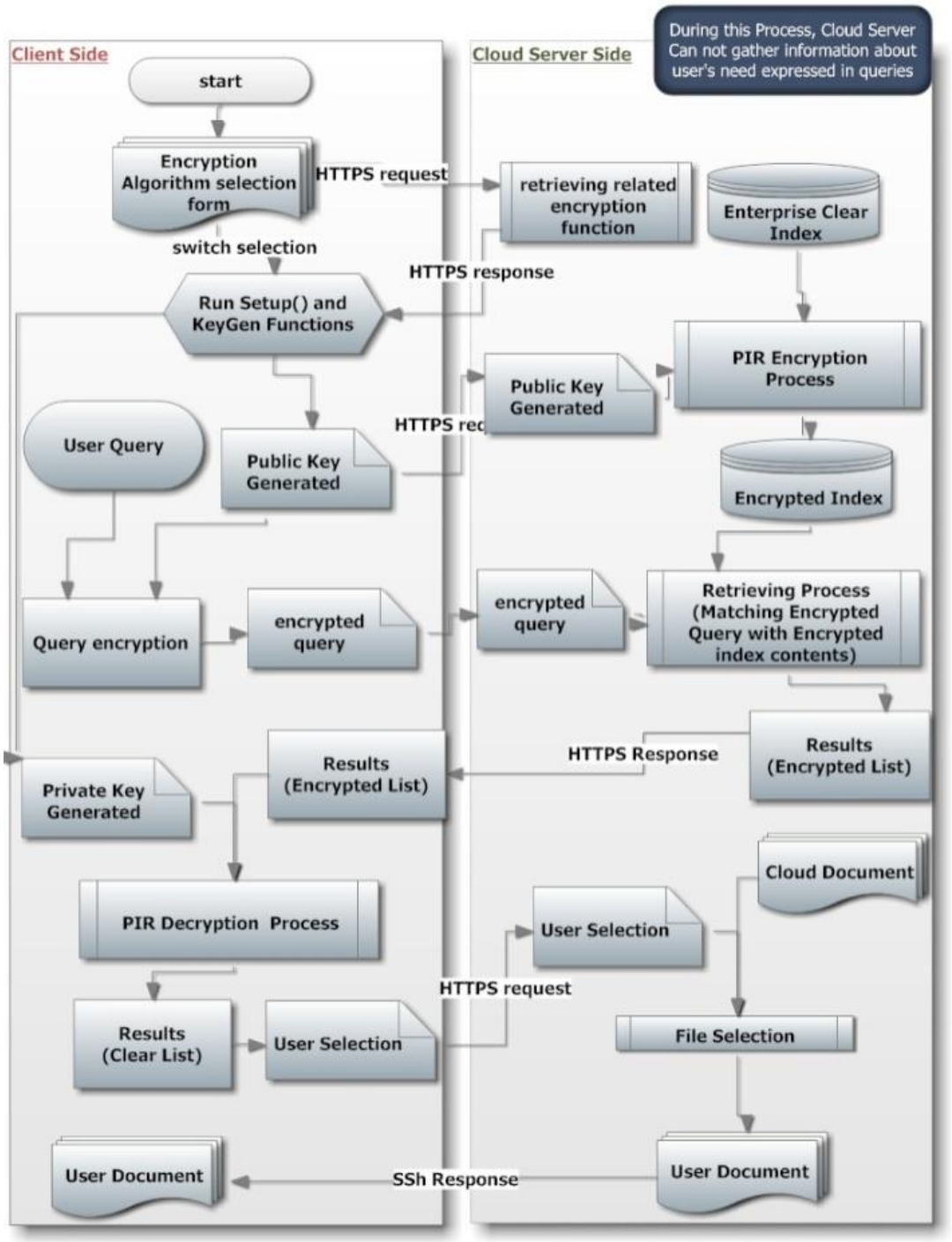

Figure 5-6 : module de recherche homomorphique.

5.6 Expérimentation

Les expérimentations ont porté sur deux facteurs principaux : le temps d'exécution et l'efficacité de la recherche Comme l'indique la figure 5.7.

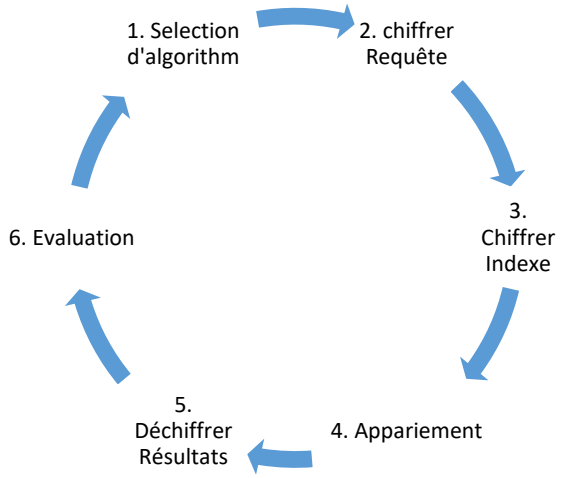

Figure 5-7 : le déroulement des expériences

Le premier facteur a comme but de mesurer l'utilité de l'utilisation des algorithmes de chiffrement homomorphique tandis que le deuxième facteur mesure l'impact de l'utilisation des protocoles PIR sur les résultats de recherche sur de données chiffrées.

5.6.1. Algorithmes implémentés

La faite qu'il existe une panoplie d'algorithmes utilisés pour implémenter les protocoles de retrait privé d'information (PIR), Nous a conduit à implémenter nos expérimentations en utilisant le sous ensemble des algorithmes de chiffrement homomorphiques suivant :

1. TSZ,
2. Waters,
3. Okamot-Uchiyama,
4. Paillier et
5. Goldwasser-Micalli

Les algorithmes de fonctionnement étant décrits au préalable dans la section précédente, le but est de trouver le meilleur algorithme de chiffrement homomorphique.

5.6.2. Plateforme de test et Simulations

Nous avons implémenté les modules décrits dans la section précédente sur une machine potable de marque Sony équipée d'un processeur I7-2620M de quatre cœurs munis d'une mémoire vive de 8Go et disque dur SSD 128Go. La machine en question est accompagnée du système Linux Ubuntu 14.4 64bit.

5.6.3. Corpus de test

Nous avons conduit nos tests sur deux corpus pour vérifier l'apport de type de collection sur l'algorithme de recherche d'information utilisé ainsi que sur chacun des algorithmes de chiffrement homomorphiques. De ce fait, les corpus de test sont les suivants :

1. Le corpus MEDLINE développé par l'organisation NLM (U.S. National Library of Médicine). Ce corpus est composé de 1,033 citation d'articles médicaux sous forme textuelle. Le corpus MEDLINE est aussi un corpus de référence dans le domaine de recherche d'information.
2. Un sous ensemble de la collection Reuter-collection composé de plus que (27000) documents textuels (nous avons utilisé juste 100 de taille entre un (01) Ko et trois (03) Ko dont chaque document contient une petite partie sous forme d'une citation des articles).

5.7 Résultats et Discussion

5.7.1. Choix de format de base de données

En premier lieu, nous avons réalisé des tests sur la collection Reuter. Les tests ont été faits sur deux requêtes : La requête « Sugar » et « Coffee » en utilisant les deux algorithmes Pallier et TSZ. La taille des clés de chiffrement et de déchiffrement a été variée entre 256 et 1024 bits

Après avoir constaté que les plateformes de recherche d'information utilisent des indexes sous forme textuelle, nous nous sommes fixé le but de choisir quel support de stockage utiliser : base de données relationnelle ou fichier texte par comparaison de temps de chiffrement/déchiffrement par rapport à la taille des clés.

Clé	Fichier texte Paillier	BDD Pallier	Fichier texte TSZ	BDD TSZ
256	66	1270	1119	1953
512	136	2783	1863	7448
1024	192	4830	3814	11749

Tableau 5-1 : Résultats de la requête « Sugar » en utilisant les deux algorithmes Pallier et TSZ.

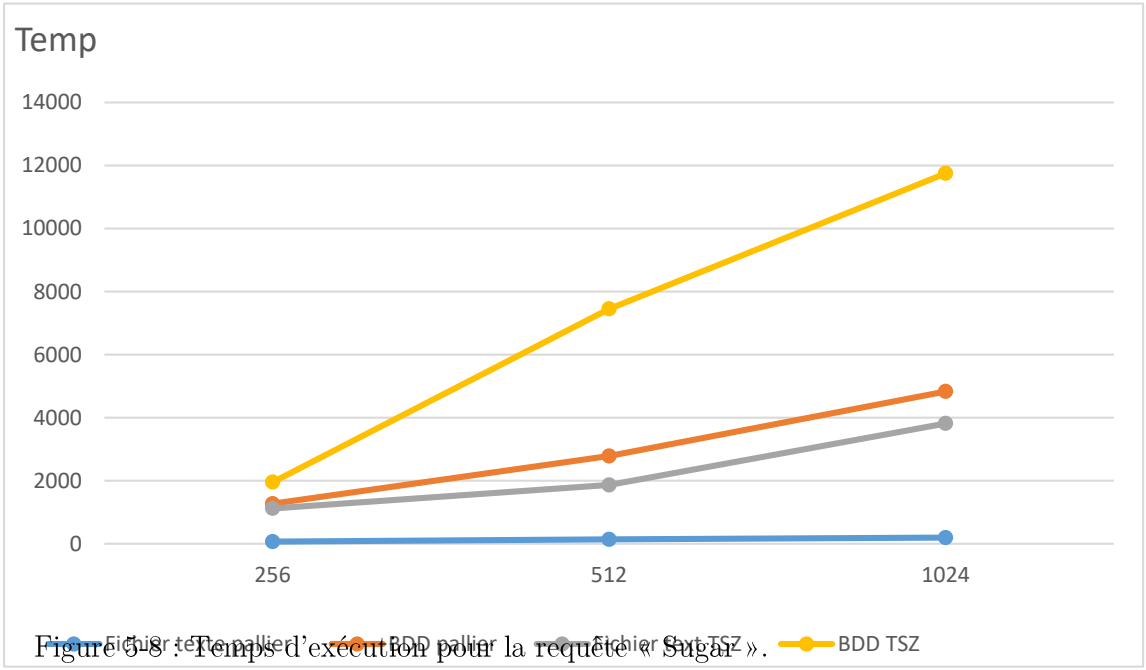

Figure 5-8 : Temps d'exécution pour la requête « Sugar ».

Clé	Fichier texte Pallier	BDD Pallier	Fichier texte TSZ	BDD TSZ
256	66	1143	853	5460
512	185	2179	1707	7716
1024	487	4025	4349	13165

Tableau 5-2 : Résultats de la requête « coffee » en utilisant les deux algorithmes pallier et TSZ.

Les deux figures 5.9 et 5.10 ont démontrées que pour l'algorithme Pallier est plus rapide que l'algorithme TSZ en termes de temps de chiffrement de base de données en clair ou fichier en clair. Même chose pour le chiffrement de la requête, ainsi que le temps de recherche en générale. Sachant que les graphes et les tableaux expriment le temps de recherche qui est égale à la somme des temps de chiffrement, de déchiffrement et de d'appariement des requêtes chiffrées et indexes chiffrés en secondes.

$$Temps_f = Temps\ de\ cryptage + Temps\ de\ décryptage + Temps\ de\ matching$$

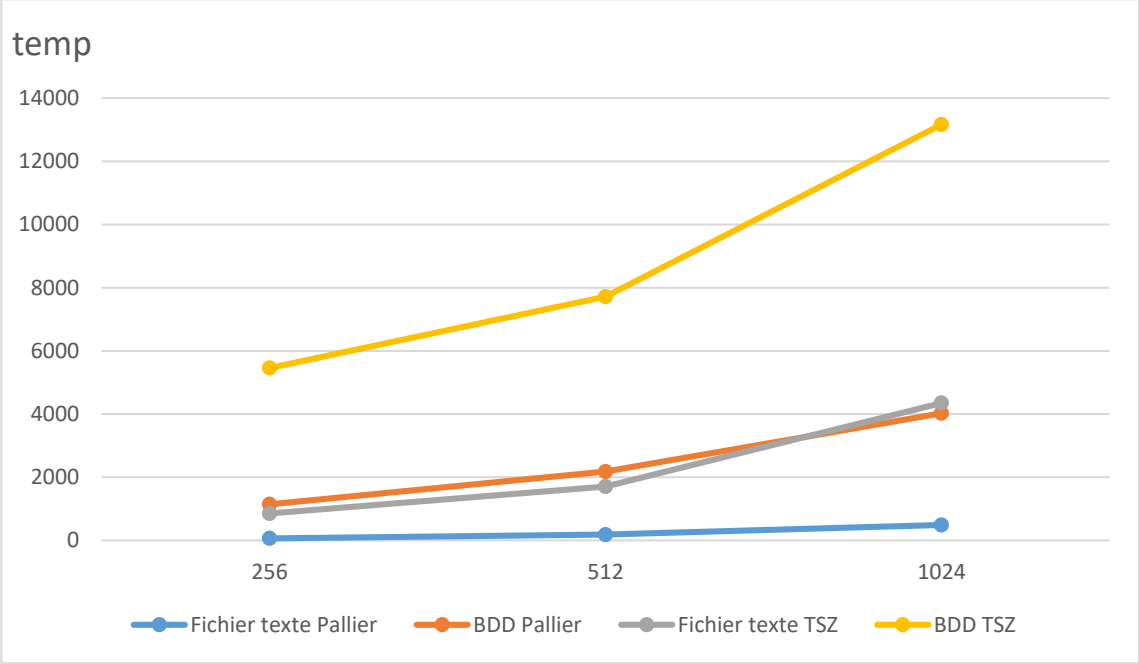

Figure 5-9 : comparaison entre le temps d'exécution pour la requête « coffee ».

Dans cette première partie, nous n'avons pas incorporé le temps d'indexation, car nous le considérons fixe pour les deux approches base de données et fichier texte (différence négligeable).

En comparant le temps d'indexation et le temps de recherche de ces deux supports de stockage, nous constatons que l'approche qui utilise le fichier texte est nettement plus rapide que la base de données. De même concernant le temps de chiffrement et du matching, nous constatons que le fichier texte et toujours mieux que la base de données.

En conclusion de cette partie de tests, nous avons décidé d'utiliser le format texte brute pour stocker les indexes lors de l'utilisation du Framework PIR dans le cloud.

5.7.2. Expérimentations

Après avoir choisi le format textuel, nous avons implémenté le Framework de retrait d'informations privé PIR pour les cinq algorithmes avec des clés de 2048 bits et une requête de taille 368 bit en claire (qui devient 1040 bits en format chiffré). Dans cette étape, nous allons comparer les résultats en terme temps de chiffrement par rapport aux temps de recherche et nous établirons une comparaison de la précision et rappel de chaque méthode.

Dans cette partie, les résultats seront issus d'interprétation d'une requête spécifique au corpus MEDLINE de rapports médicaux plus adapté à notre approche. Il s'agit de la requête "The crystalline lens in vertebrates includes humans". Cette requête a 37 documents pertinents et nous essayons de voir l'efficacité de notre approche par rapport à ce qui est annoté.

Du fait que notre intérêt porte sur l'adaptation des protocoles PIR à un enivrement clouds SaaS, nos résultats expérimentaux seront présentés sous forme de trois (03) tables.

Dans le tableau 5.3 qui concerne la qualité de recherche, le protocole Okamoto-Uchiyama a donné les meilleurs résultats. Cependant, tous les algorithmes ont une bonne précision par rapport à un faible rappel. Cela est dû au modèle sac à mots utilisé.

Type de recherche	Taille requête (bits)	rappel	Précision
Claire	368	0.27	0.70 (07/10)
TSZ	1040	0.35	0.61 (08/13)
Waters	1040	0.27	0.80 (08/10)
Okamoto-Uchiyama	1040	0.37	0.71 (10/14)
Paillier	1040	0.35	0.61 (08/13)
Goldwassers-Micali	1040	0.35	0.53 (07/13)

Tableau 5-3 : Rappel et Précision.

La figure 5.11 nous montre une comparaison graphique entre les protocoles en termes de précision et rappel.

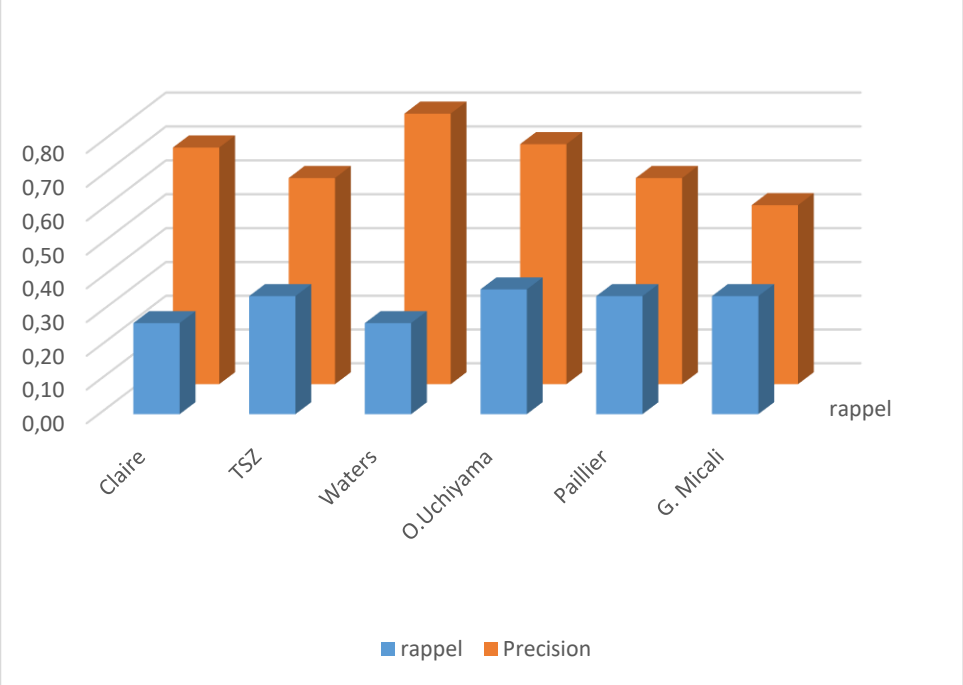

Figure 5-10 : comparaison graphique rappel/précision.

Le temps d'exécution d'un service dans le cloud est un facteur très important dans l'environnement Cloud SaaS car les utilisateurs ne doivent pas attendre longtemps pour recevoir une réponse du serveur.

Nous avons aussi comparé les protocoles en termes de temps de chiffrement et temps de recherche. La table 5.4 liste les temps d'exécution de la phase de recherche pour donner une idée sur le protocole PIR le plus performant.

Type de recherche	chiffrement (ms)	recherche	Total
claire	00	63	63
TSZ	3894	91	3985
Waters	365	346	711
Okamoto-Uchiyama	3746	35	3781
Paillier	1324	130	1454
Goldwassers-Micali	4885	40	4925

Tableau 5-4 : Temps de recherche par protocole.

Il faut notez que le temps d'exécution, surtouts lors du chiffrement de la base de données entière pour chaque requête, est généralement le facteur qui pousse à abandonner l'idée d'utiliser les protocoles PIR.

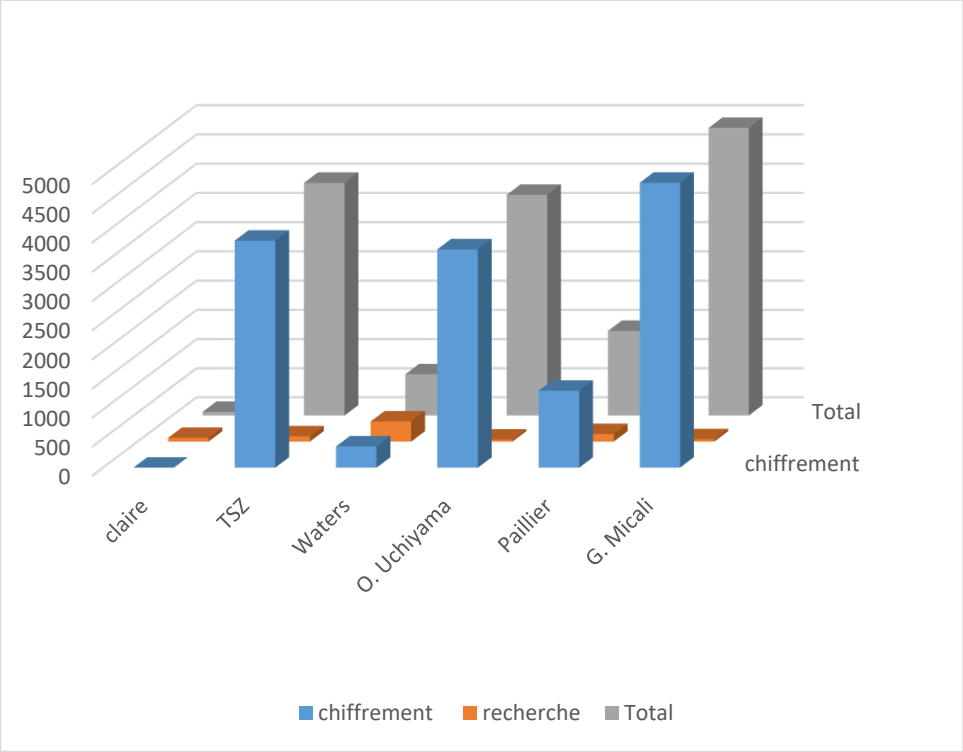

Figure 5-11 : Comparaison graphique entre le temps d'exécution.

Aussi comme le chiffrement, ce que nous avons remarqué c'est que le matériel utilisé, joue un rôle très important dans ce qui concerne la vitesse et le temps d'exécution.

5.8 Conclusion

Notre approche concrétise l'importance des systèmes de chiffrement homomorphes et l'implémentation offre la possibilité de traiter des données en tout anonymat en respectant ainsi la vie privée à l'égard des utilisateurs propriétaires de ces données.

De nombreux cryptosystèmes homomorphes ont été implémentés en pratique pour étudier le temps de calcul surtout dans le cas de la multiplication, la taille des clés et des textes chiffrés par rapport à la nature de l'algorithme ; addition ou produit qui affectent directement l'un l'autre, surtout dans le cas du chiffrement complètement homomorphe.

Dans ce chapitre nous avons aussi spécifié les protocoles PIR à implémenter. Nous avons étudié leurs performances ainsi que la durée de chiffrement, déchiffrement et la durée de traitement des opérations pour les cryptosystèmes. Les détails de la contribution et la présentation de la conception ainsi que l'architecture du Framework ont bien été explicités dans ce chapitre.

Conclusion Générale

Conclusion Générale et perspectives

Dans cette thèse, nous avons réalisé une étude des protocoles de chiffrement homomorphes qui appartiennent aux protocoles PIR (Private information Retrieval) afin d'obtenir une recherche d'information ou un retrait d'information privé et sécurisé. Le but principal est donc de montrer que les protocoles existants peuvent donner des résultats qui justifient leur adoption dans la recherche de contenu dans le Cloud médical et autre.

Bien qu'il soit impossible d'obtenir une protection absolue de la confidentialité des accès dans certains domaines problématiques en raison de l'échelle des bases de données, nous sommes convaincus que les systèmes PIR actuels sont capables de fournir une confidentialité absolue dans de nombreux domaines problématiques. Notre approche de préservation de confidentialité d'accès aux cloud médicaux encouragera les concepteurs de cloud à adopter le PIR, même s'il ne peut fournir que la confidentialité d'accès à un sous-ensemble de données important. Au fur et à mesure que la recherche et le développement continuent à s'améliorer, le protocole PIR deviendra de plus en plus pratique et pertinent pour les transactions des gens sur Internet.

Nous avons tout d'abord présenté les notions de base, et établi clairement quelles sont les primitives dont a priori nous disposons. Le but est l'indexation des documents ou nous avons présenté les concepts fondateurs de la RI. En particulier les techniques d'indexation automatique, les principaux modèles de recherche et les mécanismes de reformulation des requêtes.

Le chapitre 3 introduit les notions de base du Cloud Computing et traite les différents aspects de la sécurité dans cet environnement, en particulier ceux relatifs au chiffrement, le codage, la cryptographie symétrique et asymétrique, les fonctions de hachage et les certificats numériques.

Ensuite, nous avons dédié le chapitre 4 à la présentation de la récupération d'informations privée (ou PIR). Premièrement, nous avons introduit les notions de base du PIR, puis nous avons cité les critères d'évaluation standard afin de rendre la comparaison entre les différents protocoles PIR plus simple et faciliter le choix du meilleur protocole pour une application donnée. Ensuite, nous avons recensé des différents protocoles et indiqué les différentes avancées apportées par chaque protocole, en commençant par l'utilisation des protocoles homomorphe à base d'attributs, d'identité et enfin les protocoles complètement homomorphes. Puis nous avons exposé l'utilisation de systèmes de chiffrement homomorphe dans le contexte du Cloud Médical. Finalement, nous avons comparé les différents protocoles et montré qu'il n'est pas facile de choisir quel protocole utiliser en fonction de leurs propriétés. Un état de l'art consistant a été dressé pour chaque classe.

Les protocoles existants offrent habituellement un surcoût d'accès qui amorti leur efficacité et ralentisse leur adoption dans plusieurs domaines y compris les cloud de données médicales. Dans le chapitre 5, nous avons décrit un Framework qui rend l'accès et la recherche d'information aussi efficace que privé que possible. Il s'agit de l'adoption de la police de changement de mot de passe comme intriguer de changement de clé publique qui réduit la

complexité des protocoles en réduisant le nombre de fois que la base médicale en ligne est chiffrée. Cette dernière qui subit un chiffrement homomorphe pour chaque requête de chaque client, subie dans notre approche, un chiffrement à chaque fois les mots de passe d'authentification sont changés. Rappelons que les mots de passes sont généralement changés chaque mois, voire chaque trimestre.

Enfin, nous avons étudié l'efficacité de ces protocoles en termes de temps d'exécution et en termes d'efficacité de la recherche. Dans ce dernier point, les résultats de recherche par contenu présentent une légère perte due aux bruits qui accompagnent les algorithmes homomorphique, tandis que le temps d'exécution est assez important dans le cas des protocoles homomorphique classique, et moins important pour notre approche.

Pour conclure, malgré les bons résultats fournis par notre approche, plusieurs autres problèmes restent ouverts pour l'adoption des protocoles PIR homomorphiques dans le cloud en général :

- Pour plus de crédibilité, nous comptons tester notre approche sur des bases de test et des collections de recherche d'information plus adéquate telle que la collection TREC.
- Une collection de test plus importante implique un besoin d'utilisation de plateforme adéquate et performante. Dans ce contexte, une future utilisation de la plateforme Hadoop qui support le parallélisme est une idée très intéressante.
- La mise à jour de l'état de l'art, nous a montré de nouvelles approches homomorphiques moins consistantes comme celle de (Yagisawa, 2016) et (Yongge, 2016). Ces approches donnent de meilleurs résultats et nous comptons les améliorer
- Afin d'améliorer les performances du système PIR, nous prévoyons l'utilisation des GPUs plutôt que des CPU pour effectuer la grande partie du calcul.
- Enfin, une implémentation de ces protocoles PIR dans un contexte réel sur des données d'un cloud réel est tés importante pour monter l'importance, mesurer l'efficacité et juger l'adoption de ces derniers.

Bibliographie

Bibliographie

- Alnuem, M., Masri, S. E., Youssef, A. & Emam, A., 2011. Towards Integrating National Electronic Care Records in Saudi Arabia. *International Conference on Bioinformatics and Computational Biology.*
- Armbrust, M., Fox, A. & Griffith, R., 2010. A view of cloud computing. *Communications of the ACM,* 53(4), p. 50–58.
- Aslam, K. F., Ali, A., Abbas, H. & Haldar, N. H., 2014. A cloud-based healthcare framework for security and patients' data privacy using wireless body area networks. *Procedia Computer Science,* Volume 34, pp. 511-517.
- Azhar, M. & Laxman, M., 2014. Secured Health Monitoring System in Mobile Cloud Computing. *International Journal of Computer Trends and Technology,* 13(3), pp. 138-142.
- Baeza-Yates, B.-N. R., 2011. *Modern Information Retrieval : The Concepts and Technolog behind Search (2nd Edition).* Addison-Wesley Professional éd. s.l.:s.n.
- Balasubramaniam, S. & Kavitha, V., 2015. Hybrid Security Architecture for Personal Health Record Transactions in Cloud Computing. *Advances in Information Sciences and Service Sciences,* 7(1), pp. 121-130.
- Bethencourt, J., Sahai, A. & B., W., 2007. Ciphertext-policy attribute-based encryption. *IEEE Computer Proceedings of the 2007 IEEE Symposium on Security and Privacy,* p. 321–334.
- Bildosola, I., Río-Belver, R., Cilleruelo, E. & Garechana, G., 2015. Design and Implementation of a Cloud Computing Adoption Decision Tool: Generating a Cloud Road. *PLOS ONE,* 10(7).
- Bildosola, I., Río-Belver, R., Cilleruelo, E. & Garechana, G., 2015. Design and Implementation of a Cloud Computing Adoption Decision Tool: Generating a Cloud Road. *PLOS ONE,* 10(7).
- Blair, D. C. & Maron, M., 1990. Full-text information retrieval: Further analysis and clarification. *Information Processing and Management,* 26(3), pp. 437-447.
- Blake, I., Seroussi, G., Smart, N. & S., C. J. W., 2005. *Advances in Elliptic Curve Cryptography.* s.l.:Lecture Note Series, Cambridge University Press.
- Boneh, D. & Boyen, X., 2004. *Efficient selective-id secure identity-based encryption without random oracles,* s.l.: s.n.
- Boneh, D. & Franklin, M., 2001. Identity based encryption from the Weil pairing. *21st Annual International Cryptology Conference, Proceedings of CRYPTO.*

- Boneh, D., Goh, E. J. & Nissim, K., 2005. Evaluating 2-DNF formulas on ciphertexts.. *Evaluating 2-DNF formulas on ciphertexts..*
- Bookstein, A., 1983. Outline of a general probabilistic retrieval model. *Journal of Documentation,* 39(2), pp. 63-72.
- Cheng, F. C. & Lai, W. H., 2012. The Impact of Cloud Computing Technology on Legal Infrastructure within Internet—Focusing on the Protection of Information Privacy. *Procedia Engineering,* Volume 29, pp. 241-251.
- Chen, L., Cheng, Z., Malone-Lee, J. & Smart, N., 2006. *Efficient id-kem based on the sakai-kasahara key construction,* s.l.: IEE Proceedings of Information Security.
- Chen, T. S., 2012. Secure Dynamic Access Control Scheme of PHR in Cloud Computing. *Journal of medical systemS,* Volume 36, pp. 4005-4020.
- Christopher, D. M., Prabhakar, R. & Hinrich, S., 2008. *Introduction to Information Retrieval.* 1 edition éd. s.l.:Cambridge University Press.
- Diffie, W. & Hellman, M., 1976. New directions in cryptography. *IEEE Transactions on Information Theory.*
- Diffie, W. & Hellman, M. E., 1976. New directions in cryptography. *IEEE Trans. Inf Theory,* Volume 22, p. 644–654.
- Dimitrios, Z. & Dimitrios, L., 2012. Addressing cloud computing security issues. *Future Generation Comp. Syst. 28(3),* pp. 583-592.
- Dumais, S., 1994. Latent Semantic Indexing (LSI).. *Proceeding of TREC-3, .*
- EL Gamal, T. A., 1985. public key cryptosystem and a signature scheme based on discrete logarithms. *Advances in cryptology, Springer Berlin Heidelberg.*
- EL-YAHYAOUI, A. & ELKETTANI, M. D., 2016. Fully homomorphic encryption: state of art and comparison. *International Journal of Computer Science and Information Security,* 14(4), pp. 159-167.
- Fahsi, M. & Benslimane, S. M., 2014. *Homomorphic Private Information Retrieval Protocol for secure Data Warehouse Access.* The First International Symposium on Informatics and its Applications (ISIA2014), February 25-26, M'sila, Algeria, s.n.
- Fahsi, M. & Benslimane, S. M., 2014. *Studying the effects of conflicting tokenisation on LSA dimension reduction..* Morroco, s.n.
- Fahsi, M., Benslimane, S. M. & Rahmani, A., 2015. A Framework for Homomorphic, Private Information Retrieval Protocols in the Cloud. *International Journal of Modern Education and Computer Science (IJMECS). ISSN: 2075-016. 7(5),* pp. pp. 16-23.
- Faloutsos, C., 1985. Access methods for text. *ACM Computing Surveys, ,* 17(1), pp. 49-74.

- Foltz, P. W., 1990. Using Latent Semantic Indexing for information filtering.. *CACM, ,* pp. 40-47.
- Foster, I., 2002. What is the Grid? - a three point checklist. *GRIDtoday, ,* 1(6).
- Foster, I., Zhao, Y., Raicu, I. & Lu, S., 2009. Cloud Computing and Grid Computing 360-Degree Compared.. *CoRR, .*
- Fox, E., Betrabet, S., Koushik, M. & Lee, W., 1992. *Information Retrieval: Algorithms and Data structures; Extended Boolean model.* Prentice-Hall, Inc éd. s.l.:s.n.
- Frakes, W. B. & Baeza-Yates, R., 1992. *Information Retrieval: Data strutures and Algorithm.* Prentice Hall éd. s.l.:s.n.
- Fuhr, N., 1989. Models for retrieval with probabilistic indexing. *Information processing and management,* 25(1), p. 55–72.
- FURNAS, W., GOMEZ, L. M. & DUMAIS, S. T., 1987. The Vocabulary Problem in Human System Communication. *Communication of the ACM,* 30(11), pp. 964-971.
- Gantz & Reinsel, D., 2012. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east.. *l'International Data Corporation (IDC).*
- Gentry, C., 2009. Fully homomorphic encryption using ideal lattices.. *STOC,* Volume 9, p. 169–178.
- Gjosteen, K. & Strand, M., 2016. *Fully homomorphic encryption must be fat or ugly?,* s.l.: Cryptology ePrint Archive, Report 2016/105.
- Goldwasser, S. & Micali, S., 1984. Probabilistic encryption. *Journal of computer and system sciences, 28(2) :270–299, 1984,* 28(2), p. Probabilistic encryption.
- Goyal, V., Pandey, O., Sahai, A. & Waters, B., 2006. Attribute-based encryption for fine grained access control of encypted data. *Proceedings of the 13th ACM conference on Computer and communications security,* pp. 89-98.
- Grange, P., Ferreira, C. & Vandenborght, D., 2010. *SÉCURITÉ DU CLOUD COMPUTING: Analyse des risques, réponses et bonnes pratiques.* LIVRE BLANC éd. s.l.:Syntec numérique.
- Griebel, L., Prokosch, H.-U., Köpcke, F. & Toddenroth, D., 2015. A scoping review of cloud computing in healthcare. *BMC Medical Informatics and Decision Making,* 15(17).
- Guiraud, P., 1967. *les Structures étymologiques du lexique français," Book les Structures étymologiques du lexique français.* Payot éd. s.l.:s.n.
- Gunamalai, C. & Sivasubramanian, S., 2015. A novel method of security and privacy for personal medical record and DICOM images in cloud computing. *Journal of Engineering and Applied Sciences,* 10(10), pp. 4635-4638.

- Haufe, K., Dzombeta, S. & Brandis, K., 2014. Proposal for a Security Management in Cloud Computing for Health Care. *The Scientific World Journal,* Volume 7.
- Itani, W., Kayssi, A. & Chehab, A., 2009. Privacy as a service: privacy-aware data storage and processing in cloud computing architectures. *Proceedings of the 8th IEEE International Conference on Dependable Autonomic and Secure Computing, Chengdu, China, December, ,* pp. 711-716.
- Jinhui , Y., Shiping , C., Surya , N. & David , L., 2010. *Truststore: Making amazon s3 trustworthy with services composition..* s.l., s.n., pp. 600-604.
- Jones, K. S., 1972. A statistical interpretation of term specificity and its application. *Retrieval Journal of Documentation,* vol. 28,(No 01), pp. pp. 11-21..
- Kalyani, D. & Sridevi, R., 2016. Survey on Identity based and Hierarchical Identity based Encryption Schemes. *International Journal of Computer Applications,* 134(14), pp. 32-37.
- Kanoulas, E., 2016. A Short Survey on Online and Offline Methods for Search Quality Evaluation. *Springer Proceeding of the : RuSSIR 2015,* p. 38–87.
- Khana, F. A., Alia, A., Abbas, H. & Haldar, N. H., 2014. A cloud-based healthcare framework for security and patients' data privacy using wireless body area networks. *Procedia Computer Science,* Volume 34, pp. 511-517.
- Koblitz, N., 1987. Elliptic curve cryptosystems. *Math. Comp,* Volume 48, p. 203–209.
- Kokkinos, P., Kalogeras, D., Levin, A. & Varvarigos, E., 2016. Survey: Live Migration and Disaster Recovery over Long-Distance Networks. *ACM Computing Surveys CSUR,* 42(2), p. No 26.
- Kuyoro, S. O., Ibikunle, F. & Awodele, O., 2011. Cloud Computing Security Issues and Challenges. *International Journal of Computer Networks,* 3(5), pp. 247-255.
- Kwok, K., 1989. A neural network for probabilistic information retrieval. *International ACM SIGIR Conference on Research and Developpement in Information Retrieval,* pp. 21-30.
- Lakshmi, B. N., Garth, G. R., Shankar , P. & Jiri, S., 2007. An analysis of latent sector errors in disk drives..
- Lancaster, F., 1979. *Information Retrieval Systems: characteristics, testing, and evaluation.* s.l.:John Wiley, New York.
- Lang, H., Wang, B., Metzler, D. & Li, J.-T., 2010. Improved Latent Concept Expansion Using Hierarchical Markov Random Fields.. *Proceedings of the 19th ACM international conference on Information and knowledge management,* pp. 249-258.
- Lauter, K., Naehrig, M. & Vaikuntanathan, V., 2011. *Can Homomorphic Encryption be Practical?.* s.l., ACM CCSW Proceeding.

- Lee, C., Chung, P. & Hwang, M., 2013. A survey on attribute-based encryption schemes of access control in cloud environments. *I. J. Network Security,,* 15(4), p. 231–240.
- Li, H., Dai, Y., Tian, Y. & Yang, H., 2009. Identity-based authentication for cloud computing. *Springer Verlag Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09,* p. 157–166.
- Li, J. & Wang, L., 2015. *Noise-free Symmetric Fully Homomorphic Encryption based on noncommutative rings,* s.l.: rapport eprint 641.
- Lim, H. & PAterson, J., 2011. Identity-based cryptography for grid security. *Int. J. Inf. Secur.,* 10(1), p. 15–32.
- Lim, H. & Robshaw, M. J. B., 2005. A dynamic key infrastructure for grid. *Springer-Verlag Proceedings of the European conference on Advances in Grid Computing,* p. 255–264.
- Lim, H. & Robshow, M., 2004. On identity-based cryptography and grid computing. *Lecture Notes in Computer Science,* p. 474–477.
- Li, M., Yu, S., Ren, K. & Lou, W., 2010. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings. *Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, ,* pp. 89-106.
- LOSEY, J., 2015. Surveillance of Communications: A Legitimization Crisis and the Need for Transparency. *International Journal of Communication,* Volume 9, p. 3450–3459.
- Lupse, O. S., Vida, M. M. & Tivadar, L. S., 2012. Cloud Computing and Interoperability in Healthcare Information Systems. *The First International Conference on Intelligent Systems and Applications,* pp. 81-85.
- Marinos, A. & Briscoe, G., 2009. Community cloud computing. *Cloud Computing Springer Berlin Heidelberg,* p. 472–484.
- Maron, J. L., 1960. On relevance, probabilistic indexing and information retrieval. *J. ACM,* 7(3), p. 216–244.
- Mehraeen, E., Ayatollahi, H. & Ahmadi, M., 2016. Health Information Security in Hospitals: the Application of Security Safeguards. *ACTA INFORM MED,* 24(1), pp. 47-50.
- Mell, P. & Grance, T., 2009. *The nist definition of cloud computing.* 5 éd. s.l.:National Institute of Standards and Technology.
- Metzler, D. & Croft, W. B., 2007. Latent concept expansion using markov random fields. *Proceedings of the international ACM SIGIR conference on Research and development in information retrieval,* p. 311–318.

- Miller, R., 2010. Amazon addresses ec2 power outages. *In Data Center Knowledge, volume 1,* pp. 2-4.
- Miller, V. S., 1986. Uses of elliptic curves in cryptography,. *Lect. Notes Comp. Sci. Springer-Verlag,* Volume 218, p. 417–426.
- Neisse, R., Steri, G., Fovino, I. N. & Baldini, G., 2015. SecKit: A Model-based Security Toolkit for the Internet of Things. *Computers & Security,* Volume 54, pp. 60-76.
- Okamoto, T. & Uchiyama, S., 1998. A new public-key cryptosystem as secure as factoring. *Springer-VerlagLecture Notes in Computer Science; Advances in Cryptology,* Volume 1403 , p. 308–318.
- Paice, C., 1996. Method for evaluation of stemming algorithms based on error counting. *Journal of the American Society for Information Science,* 47(8), pp. 632-649..
- Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes. *Advances in Cryptology Eurocrypt,* Volume 1592, p. 223–238.
- Papadimitriou, G., S., A., G., S. & al., a., 2008. *Amazon Enters the Cloud Computing Business.* 353-2008-1. éd. s.l.:Stanford CasePublisher .
- Parekh, M. & Saleena, B., 2015. Designing a Cloud based Framework for HealthCare System and applying Clustering techniques for Region Wise Diagnosis. *Procedia Computer Science,* Volume 50, pp. 537-542.
- Rahman, S. M. & al., 2015. Privacy preserving secure data exchange in mobile P2P cloud healthcare environment. *Peer-to-Peer Netw,* pp. 1-16.
- Revathy, S. & Gopu, D., 2016. A Survey On Secrecy Preserving Multi-keyword Matching Technique On Cloud For Encrypted Data. *International Journal of Latest Research in Engineering and Technology,* 2(1), pp. 47-50.
- Rijndael, 2001. Advanced Encryption Standard (AES). *Processing Standards Publication 197.*
- Rivest, R. L., Adleman, L. & Dertouzos, M. L., 1978. On data banks and privacy homomorphisms. *Foundations of secure computation,* 4(1), p. 169–180.
- Rivest, R. L., Shamir, A. & Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM,* 21(2), pp. 120-126 .
- Rivest, R., Shamir, A. & Adleman, L., 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM ,* 21(2), pp. 120-126.
- Robertson, K., 1988. Relevance Weighting of Search Terms. *Document retrieval systems, Taylor Graham Publishing,* p. 143–160.
- Robertson, S. E., 1977. The Probability Ranking Principle in IR.. *Journal of Documentation,* 33(4), pp. 294-304.

- Robertson, S. & Walker, S., 1997. On relevance weights with little relevance information.. *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval,* p. pp. 16–24 .
- Ronan, C. & Colm, O., 2006. Evolving local and global weighting schemes in information retrieval. *information retrieval,* Volume 9, p. 311–330.
- Rostrom, T. & Teng, C. C., 2011. Secure communications for PACS in a cloud environment. *Proceeding of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* pp. 19-22.
- Sachdev, A. & Bhansali, M., 2013. Enhancing Cloud Computing Security using AES Algorithm. *International Journal of Computer Applications,* 67(9), pp. 19-23.
- Saevanee, H., Clarke, N., Furnell, S. & Biscione, V., 2015. Continuous user authentication using multi-modal biometrics. *Computers & Security,* Volume 53, pp. 234-246.
- Sahai, A. & Waters, B., 2005. Fuzzy identity-based encryption. *Springer Verlag Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques,* p. 457–473.
- Sakai, R. & Kasahara, K., 2003. *Id based cryptosystems with pairing on elliptic curve,* s.l.: Cryptology ePrint Archive.
- Sakai, R., O. K. & Kasahara, M., 2001. Cryptosystems based on pairing over elliptic curve (in Japanese). *Symposium on Cryptography and Information.*
- Salton, G., 1969. “A comparison between manual and automatic indexing methods. *American Documentation,* 20(1), p. 61–71..
- Salton, G., 1971. *The SMART Retrieval System—Experiments in Automatic Document Processing.* Prentice-Hall éd. s.l.:s.n.
- Salton, G., 1983. *Introduction to Modern Information Retrieval,.* s.l.:McGrawHill.
- Salton, G., 1986. *Introduction to Modern Information Retrieval. ,.* McGraw-Hill, Inc éd. s.l.:s.n.
- Salton, G., 1987. *Term Weighting Approaches in Automatic Text Retrieval.* Cornell University, Ithaca, NY éd. s.l.:s.n.
- Salton, G. & Yang, C. S., 1975. A vector space model for automatic indexing. *Proc. Commun, ACM, ,* pp. 613-620.
- Sanderson, M., 2010. Test collection based evaluation of information retrieval systems.. *Foundations and Trends in Information Retrieval,* 4(4), p. 247–375.
- SAVOY, J., 2006. *La toile et ses moteurs de recherche. In Indices, index, indexation.* s.l., proceeding of CERSATES/GERICO laboratory conference.

- Schridde, C. et al., 2010. An identity-based security infrastructure for cloud environments. *Proceeding of IEEE International Conference on Wireless Communications, Networking and Information Security.*
- Schridde, C. et al., 2010. An identity-based security infrastructure for cloud environments. *IEEE International Conference on Wireless Communications, Networking and Information Security.*
- Shamir, 1985. Identity-based cryptosystems and signature schemes. *Lecture Notes in Computer Science,* Volume 196.
- Shannon, C. E., 1949. Communication theory of secrecy systems. *The Bell System Technical Journal,* 28(4), pp. 656-715.
- Singhal, A., Salton, G., Mitra, M. & Buckley, C., 1996. Document length normalization. *Information Processing and Management,* 32(5), p. 619–633.
- Smid, M. E. & Branstad, D. K., 1992.. The Data Encryption Standard: Past and Future. *Contemporary Cryptography: The Science of Information Integrity, IEEE Press.*
- Tancrède, L., 2014. Chiffrement (complètement) homomorphe : de la théorie à la pratique. *GREYC.*
- Van Dijk, M., Gentry, C., Halevi, S. & Vaikuntanathan, V., 2010. Fully homomorphic encryption over the integers. *In Advances in cryptology-EUROCRYPT,* pp. 24-43.
- Van Rijsbergen, C. J., 1979. Information retrieval. *London: Butterworth.*
- Velumadhava, R. R. & Selvamani, K., 2015. Data Security Challenges and Its Solutions in Cloud Computing. *Procedia Computer Science,* Volume 48, pp. 204-209.
- Vernam, G., 1926. Cipher Printing Telegraph Systems. *AIEE (IEEE) ,* XI(V), pp. 109-115.
- Vidya, S., Vani, K. & Kavin, P. D., 2012. Secured Personal Health Records Transactions Using Homomorphic Encryption In Cloud Computing. *International Journal of Engineering Research & Technology,* 1(5), pp. 1-10.
- Voorhees, E., 2007. TREC: Continuing information retrieval's tradition of experimentation.. *Communications of the ACM,* Volume 50, p. 51–54.
- Voorhees, E. & Harman, D., 2005. *TREC: Experiment and Evaluation in Information Retrieval..* s.l.:Digital Libraries and Electronic Publishing, MIT Press.
- W., L. H. & Robshaw, M. J., 2004. On identity-based cryptography and grid computing. *Lecture Notes in Computer Science,* p. 474–477.
- Wayne, J. & Timothy, G., 2011. *Guidelines on Security and Privacy in Public Cloud Computing.* Draft Special Publication 800-144 éd. s.l.:NIST .

- Wong, S., Ziarko, W. & Wong, P., 1985. Generalized vector space model in information retrieval.. *Proceedings of the 8th ACM SIGIR Conference on Research and Development in information retrieval New-York, USA,,* pp. 18-25.
- Xun, Y., Russell, P. & Elisa, B., 2014. Homomorphic Encryption. Dans: *SpringerBriefs in Computer Science. Chapter 2.* s.l.:s.n., pp. 27-46.
- Yagisawa, M., 2016. *Improved fully homomorphic encryption with composite number modulus,* s.l.: Cryptology ePrint Archive Report 2016/50,.
- Yongge, W., 2016. Octonion Algebra and Noise-Free Fully Homomorphic Encryption (FHE) Schemes. *proceeding of the European Symposium on Research in Computer Security.*
- Youssef, A., 2014. A Framework for Secure Healthcare Systems Based on Big Data Analytics in Mobile Cloud. *International Journal of Ambient Systems and Applications,* 1(2), pp. 1-11.
- Yu, S., Wang, C., Ren, K. & Lou, W., 2010. Achieving secure, scalable, and fine-grained data access control in cloud computing. *IEEE Press Proceedings of the 29th conference on Information communications,* p. 534–542.
- Zadrozny, S. & Kacprzyk, J., 2005. *An Extended Fuzzy Boolean Model of Information Retrieval Revisited.* s.l., The 14th IEEE International Conference on Fuzzy Systems, FUZZ '05.
- Zhang, R. & Liu, L., 2010. Security Models and Requirements for Healthcare Application Clouds. *IEEE 3rd International Conference on Cloud Computing, Miami,* pp. 268-275.

I. Résumé

L'utilisation professionnelle de stockage de données du domaine sanitaire dans les Cloud implique des extensions de recherche d'information. Cependant, ces extensions doivent offrir une protection contre des menaces existantes, par exemple, des pirates informatiques, des administrateurs de serveur et les prestataires de services qui utilisent des données personnelles des gens pour leurs propres buts. En effet, les serveurs Cloud maintiennent les traces d'activités d'utilisateur et des requêtes, ce qui met en péril la sécurité des données utilisateur contre des pirates informatiques de réseau. Ils peuvent même utiliser ces traces pour adapter ou personnaliser leurs plateformes sans accords des utilisateurs.

Dans cette thèse, nous nous intéressons à l'application du chiffrement homomorphe au Cloud Computing, particulièrement au Cloud des données médicales, afin d'assurer la confidentialité des données sensibles des patients stockées dans les serveurs distants, et gérées par les fournisseurs de Cloud. Nous étudions et comparons les durées de chiffrement, de déchiffrement et de traitement des cryptosystèmes homomorphes existants. Nous proposons un Framework de retrait d'information privé qui implémente des protocoles de chiffrement homomorphes sur le corpus de rapports médicaux destinés à la recherche d'information. Nous étudions l'efficacité de cette solution par une évaluation de temps de recherche d'information. Les résultats expérimentaux montrent que notre approche assure un niveau raisonnable et acceptable de confidentialité pour la récupération de données dans le cloud.

II. Abstract

Professional use of cloud health storage around the world implies Information-Retrieval extensions. These developments should help users find what they need among thousands or billions of enterprise documents and reports. However, extensions must offer protection against existing threats, for instance, hackers, server administrators and service providers who use people's personal data for their own purposes.

Indeed, cloud servers maintain traces of user activities and queries, which compromise user security against network hackers. Even cloud servers can use those traces to adapt or personalize their platforms without users' agreements.

For this purpose, we suggest implementing Private Information Retrieval (PIR) protocols to ease the retrieval task and secure it from both servers and hackers. We study the effectiveness of this solution through an evaluation of information retrieval time, recall and precision. The experimental results show that our framework ensures a reasonable and acceptable level of confidentiality for retrieval of data through cloud services.

III. ملخص

الاستخدام المهني لتخزين المعلومات الصحية في السحابة المعلوماتية العالمية يتطلب استحداث لملحقات استرجاعها. وينبغي لهذه التطورات أن تساعد المستخدمين على العثور على ما يحتاجونه من بين الآلاف أو الملايين من وثائق المشاريع والتقارير. ومع ذلك، يجب توفر ملحقات حماية ضد التهديدات القائمة، على سبيل المثال المتسللين، مسؤولي الخادم ومقدمي الخدمات الذين يستخدمون البيانات الشخصية للناس لأغراضهم الخاصة. في الواقع، خوادم السحابة قوم بالحفاظ على آثار للأنشطة المستخدم والاستفسارات، هذا الذي يهدد أمن المستخدم ويعرض حسابه لقراصنة الشبكة. كما يمكن للخوادم السحابية استخدام تلك آثار لتكيف أو تخصيص برامجها دون اتفاقات مع المستخدمين.

لهذا الغرض، نقترح تنفيذ خاصة استرجاع المعلومات باستخدام بروتوكولات PIR لتسهيل مهمة البحث والاسترجاع للوثائق وضمان الحصول عليها بسرية. ندرس فعالية هذا الحل من خلال تقييم وقت استرجاع المعلومات، الحجم والدقة. أظهرت النتائج التجريبية ان نظامنا يضمن مستوى معقول ومقبول من السرية لاسترجاع البيانات من خلال الخدمات السحابية.