REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED KHIDER - BISKRA

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة محمد خيضر بسكرة

Faculté des sciences exactes et des sciences de la nature et de la vie
Département d'informatique

THÈSE

Pour l'obtention de grade de

DOCTEUR EN SCIENCES

Option : Informatique

Titre

Une approche basée agent pour la sécurité dans le Cloud Computing

Par
YAGOUB Mohammed Amine

Soutenue le :16/06/2019

Devant le jury composé de

Président	:	**Pr. Benmohamed Mohamed** de l'université de Constantine 2
Rapporteur	:	**Pr. Kazar Okba** de l'université de Biskra
Examinateur	:	**Pr. Imad Saleh** de l'université de Paris 8
Examinateur	:	**Pr. Bennoui Hammadi** de l'université de Biskra
Examinateur	:	**Dr. Benharzallah Saber** (MCA) de l'université de Batna 2
Examinateur	:	**Dr. Rezeg Khaled** (MCA) de l'université de Biskra

Remerciements

En premier lieu, je remercie le bon Dieu de m'avoir donné la force et la patience nécessaire pour achever ce travail de thèse.

Je tiens à remercier très sincèrement toutes les personnes qui, par leurs conseils et leurs encouragements ont contribué à l'aboutissement de ma thèse de doctorat :

- En premier lieu, je tiens à exprimer ma profonde reconnaissance à mon directeur de thèse, Monsieur le Professeur **Kazar Okba** pour m'avoir accueilli au sein de son équipe de recherche et du laboratoire LINFI et également pour m'avoir fait bénéficier de vos connaissances.

- J'adresse également mes remerciements les plus chaleureux à Monsieur **Laouid Abdelkader** et à Monsieur **Mounir Beggas**, Docteurs à l'Université d'Eloued ; vous m'avez guidé et encadré avec un enthousiasme permanent ; vous m'avez transmis les notions nécessaires et indispensables à ma recherche.

- Ma gratitude s'adresse également à Monsieur **Abasse Massaoud,** Docteur à l'Université d'Eloued, et à **Madame Gaia Sana,** Enseignante à l'Université d'Eloued; la richesse et la pertinence de vos remarques, tant sur la forme que sur le fond, ont contribué à améliorer de manière significative le document que je vous soumets aujourd'hui.

- Je veux également remercier tous les membres de jury : **Pr. Benmohamed Mohamed**, Professeur à l'Université de Constantine 2, pour m'avoir fait l'honneur de présider mon jury de thèse, ainsi que **Pr. Imad Saleh**, Professeur à l'Université de Paris 8, Pr. **Bennoui Hammadi**, Professeur à l'université de Biskra, **Dr. Benharzallah Saber**, (MCA) de l'université de Batna2, **Dr. Rezeg Khaled** (MCA) de l'université de Biskra; pour m'avoir fait l'honneur d'accepter de juger ce travail et l'intérêt qu'ils ont porté à mon travail.

Yagoub Mouhammed Amine

Dédicaces

À mes chers parents, que Dieu vous garde auprès de moi.

À mes chers frères et sœurs.

À ma petite famille : ma femme, mes fils et ma fille.

À mes fidèles amis.

Yagoub Mohammed Amine

Résumé

Au cours de la dernière décennie, le Cloud Computing est devenu le fondement de l'architecture informatique de la prochaine génération des technologies de l'information (IT). Cela a également joué un rôle important en motivant les propriétaires de données à externaliser leurs systèmes complexes des sites locaux vers de Cloud Computing. Ceci entraîne une grande flexibilité et des réductions des couts dépenses. Bien que cet aspect soit très positif pour la protection, les données devraient être toujours chiffrées avant d'utiliser des ressources extérieures. L'objectif principal de ce travail de thèse est de mettre en œuvre cette approche de manière précise et fiable. La solution doit fournir un schéma de chiffrement permettant aux utilisateurs de conserver certaines fonctions et propriétés importantes, telles que la possibilité de calculs, la recherche, la modification et la préservation de l'ordre entre les données. Ici, nous pouvons parler de la propriété ou du concept de homomorphique sur des données chiffrées dans le Cloud. D'autre part, même si une autre partie non confiant peut accéder aux données, ces données apparaîtront chiffrées et bêtises. Dans ce travail, nous avons proposé une solution robuste et complète combinant la technologie d'obscurcissement (pour sécuriser l'interface utilisateur), des algorithmes de chiffrement hybrides (pour sécuriser le transport et la communication) et des techniques de chiffrement complètement homomorphique (pour assurer les opérations de stockage). Nous avons également proposé une nouvelle architecture de sécurité basée sur les systèmes multi-agents pour effectuer toutes les opérations de communication et de stockage dans un environnement Cloud. Dans les systèmes proposés, nous avons utilisé le plus grand nombre possible de technologies intelligentes de sécurité.

Mots-clés — Sécurité de Cloud Computing, système multi-agents, chiffrement complétement homomorphique, chiffrement préservant l'ordre.

Abstract

In the last decade, Cloud Computing has been envisioned as the next generation architecture of Information Technology (IT). This advent motivates the data owners to outsource their complex systems from local sites to the Cloud for great flexibility and economic savings. However, the protected data should to be encrypted before outsourcing. The main objective of this thesis is to implement properly this approach. The solution should provide an encryption scheme, in such a way the user may maintain some functions such as arithmetic operations, research, update requests and preserving order, i.e., the homomorphic on the encrypted Cloud data. Moreover, even if another tenant can access to the stored data, all the stored data will appear encrypted to the unknown users. We are looking for the proposition of a strong solution that combines obfuscation technique in order to secure user interface, hybrid encryption algorithms to secure transport, communication operations and fully homomorphic encryption approach for securing storage operations. In this work, we have also proposed new security architecture based on a multi-agent system to conduct all communications and storage operations in a Cloud Computing environment, where we have used various intelligent security technologies as much as possible.

Keywords — Cloud Computing security, multi-agent system, fully homomorphic encryption, order preserving encryption.

ملخـــص

منذ العشر السنوات الماضية، اصبحت الحوسبة السحابية هي اساس بنية الجيل التالي لتقنية المعلومات (IT) . هاته الاضافة لعبت دورا مهما في تحفيز أصحاب البيانات على الاستعانة بمصادر خارجية لنقل أنظمتهم المعقدة من المواقع المحلية إلى الحوسبة السحابية للحصول على مرونة كبيرة و تخفيض للمصاريف. بالرغم من هذا الجانب الجد ايجابي، فلحماية البيانات، يجب دائما تشفيرها قبل الاستعانة بمصادر خارجية ونقلها هناك. الهدف الرئيسي من هته الاطروحة اذا، هو تنفيذ وتجسيد هذا النهج بشكل دقيق وموثوق. الحل هنا، يجب أن يوفر نظام تشفير، يضمن للمستخدم الحفاظ على بعض الخصائص المهمة مثل امكانية اجراء العمليات الحسابية، البحث ،عمليات التحديث والحفاظ على خاصية الترتيب بين البيانات ، يكمننا هنا التكلم عن خاصية او مفهوم التشاكل (homomorphic) على بيانات السحابة المشفرة. من ناحية أخرى، حتى إذا كان هناك طرف آخر غير موثوق يمكنه الوصول إلى البيانات المخزنة، فانها ستظهر له مشفرة ومبهمة . في هذا العمل نتطلع إلى اقتراح حل قوي وشامل يجمع بين تقنية التشويش لتأمين واجهة المستخدم، وخوارزميات التشفير الهجينة لتأمين النقل والاتصال، وتقنيات التشفير الكلية التشاكل لضمان عمليات التخزين. في عملنا هذا اقترحنا ايضا بنية أمان جديدة تعتمد على نظام متعدد العملاء لتجسيد واجراء كل عمليات الاتصالات والتخزين في بيئة ذات حوسبة سحابية حيث استغللنا مختلف التقنيات الأمنية الذكية قدر الإمكان.

الكلمات المفتاحية - أمان الحوسبة السحابية ، نظام متعدد العملاء ، تشفير متشاكل تمامًا ، تشفير محافظ على الترتيب.

Liste des figures

Liste des tableaux

Table des matières

Introduction générale

1. Introduction

Le Cloud Computing est devenu un nouveau paradigme surutilisé pour l'hébergement et la fourniture des ressources via Internet. Il représente la cinquième génération de l'informatique après les mainframes, les ordinateurs personnels, le paradigme client/serveur et le web (World Wide Web). Cette approche n'est pas tout à fait nouvelle. Il y a quelques années, IBM a déjà proposé l'informatique « on-demand » sous l'approche systématique ASP (Application Service Provider). Elle représente le fait de proposer une application sous forme de service. Les années 80 furent aussi le début de la technologie de virtualisation. Tous ces concepts et technologies ont amené, petit à petit, à inventer une nouvelle manière de proposer l'informatique « comme un service ». Les services informatiques sont fournis sous la forme d'un utilitaire pour décrire un modèle commercial en terme de fourniture à la demande. Dans ce modèle, l'utilisateur paye les fournisseurs de services en fonction de sa consommation. De manière similaire, lorsque les clients obtiennent des services traditionnels et publics tels que l'eau, l'électricité, le téléphone, etc.

Le Cloud Computing peut être défini comme un modèle informatique distribué et spécialisé, configurable de manière dynamique, mutualisée, évolutive (Puissance de calcul élevée, espace de stockage important, etc.) et fourni à la demande via des réseaux de communication. Ce nouveau paradigme offre plusieurs avantages comme le déploiement rapide, le paiement à l'usage, la réduction de coûts, l'évolutivité facile, la délivrance plus rapide de service, l'accès au réseau omniprésent, etc. (voir Figure 0-1). En raison de ces diverses caractéristiques, le Cloud Computing devient une solution intéressante pour les entreprises et les chercheurs.

Le Cloud Computing est considéré comme un système entièrement virtualisé, permettant à la fois le calcul, le stockage et l'utilisation des ressources logicielles ainsi que les serveurs en tant que plate-forme unique. Les services de gestion de données sont actuellement exploités dans l'environnement local d'utilisateur, mais ils sont servis à distance par des fournisseurs de services Cloud (CSP Cloud Service Provider). Les services sont proposés sous forme abstraite tels que les utilisateurs peuvent ne pas savoir où, quand, comment, pourquoi et par qui leurs données sont consultées ou modifiées dans un environnement Cloud. Cependant, le Cloud Computing souffre de nombreux problèmes de sécurité. De plus, les CSP sont plus vulnérables aux adversaires et aux pirates informatiques susceptibles d'exploiter ces avantages. Le Cloud est vulnérable du point de vue sécurité et confidentialité des données, car les données sensibles des utilisateurs sont stockées dans un CSP tiers.

Tous ces faiblesses et défiances construisent un élément commun qui est la problématique de confiance et de sûreté entre les fournisseurs et les consommateurs, parce que le Cloud Computing nécessite la confiance aux fournisseurs des services cloud. En conséquence, la confiance représente l'un des principaux facteurs pour l'adoption de ce nouveau paradigme.

À travers cette thèse, nous procédons à l'identification des défis liés à l'adoption du Cloud Computing. Nous nous intéressons en particulier aux risques liés au traitement et stockage des données en proposant un ensemble de nouveaux techniques et solutions de sécurité fonctionnant à la fois au niveau fournisseurs et au niveau clients. Le but ultime est d'offrir une expérience plus sécurisée d'un environnement informatique en Cloud.

2. Contexte

Le domaine de Cloud Computing est devenu plus large, vague et en chevauchement. Comme il n'existait aucune définition standard du Cloud Computing, l'Institut national des normes et de la technologie (NIST) a proposé une définition en 2011 qui inclut les modèles de déploiement, les types, les caractéristiques essentielles de Cloud Computing. La Figure 0-1 présente des aspects importants de Cloud Computing, tels que proposés par le NIST.

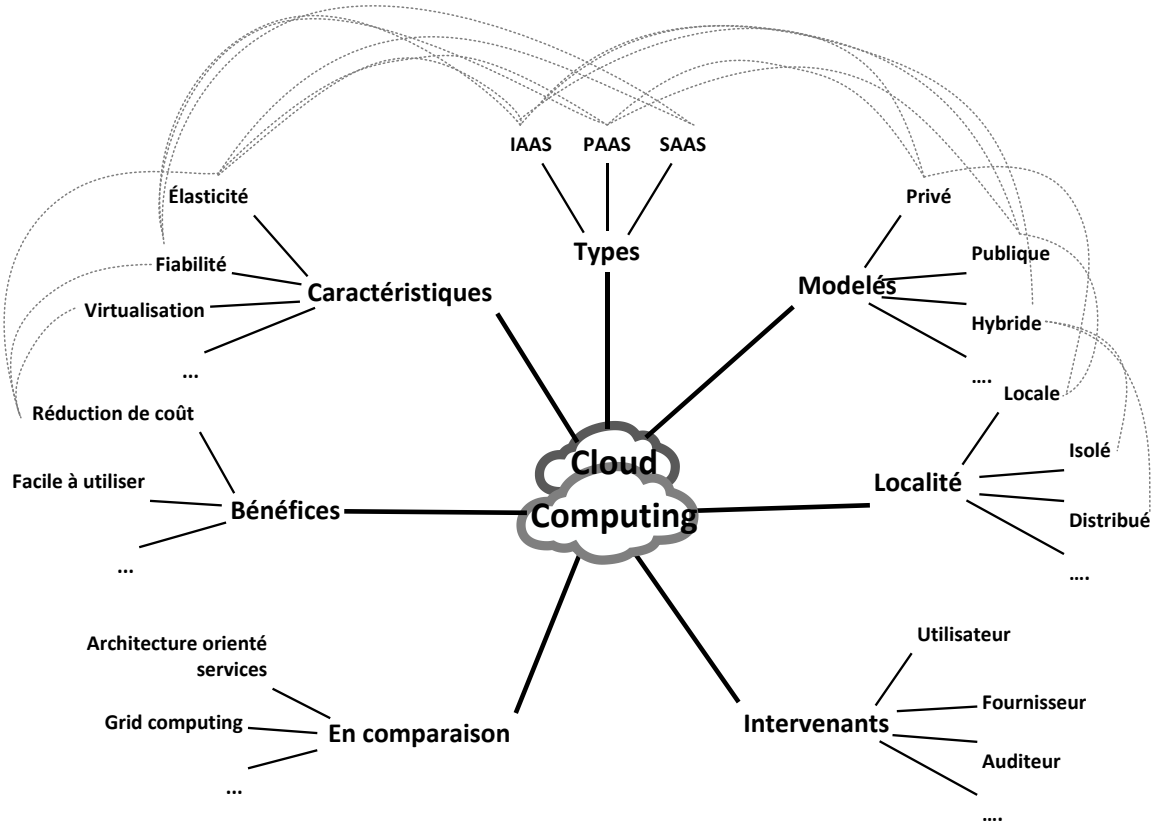


Figure 0-1 :Les aspects importants de Cloud Computing

Le Cloud Computing consiste à ne plus gérer les données sur les dispositifs des utilisateurs (ordinateur, Smartphone, . . .) mais en confie la gestion à une entité tierce accessible par Internet. Les utilisateurs peuvent ainsi accéder à leurs données de n'importe où et à n'importe quel moment, pour les utiliser dans un traitement ou pour les consulter. Cependant, on peut diviser le Cloud Computing en deux axes qui sont le calcul en Cloud, où le traitement des données est confié à l'entité tierce. Le deuxième c'est le stockage en Cloud, où le stockage des données est assuré par cette même entité. Ces deux approches peuvent être compatibles entre elles. Nous décidons cependant de se focaliser sur le stockage en Cloud.

3. Problématique et motivation

La sécurité est le principal facteur du succès de la technologie Cloud. Nombreuses recherches ont mentionné que la sécurité est désormais le principal défi à relever dans l'environnement Cloud. Il y a des nombreux problèmes de sécurité liés au Cloud Computing, qui sont similaires aux architectures traditionnelles. Cela nécessite de réévaluer les risques liés à chacun de leurs domaines critiques dans un nouvel environnement hasardeux, où les ressources sont partagées par plusieurs intervenants (utilisateurs ou des fournisseurs). En fonction de l'architecture Cloud, le client utilise la caractéristique multi-locataire, dans ce cas, les niveaux des problèmes de sécurité sont différents. Mais sans aucun doute, le fonctionnement « en tant que service » du Cloud, présente le plus grand risque. Plusieurs normes et mesures de gestion de la sécurité ont été conçues pour protéger le système Cloud, mais sa sécurité présente néanmoins un risque élevé à cause des techniques de piratage innovantes.

Les problèmes de sécurité relevés par le paradigme du Cloud ne sont pas toujours traités avec le point de vue de l'utilisateur. La sécurité de l'information est un aspect essentiel à prendre en considération dans le Cloud comme dans tous les types de systèmes d'information traditionnels. Les utilisateurs du Cloud travaillent avec des données et des applications qui sont souvent situées hors site. Cependant, de nombreuses organisations sont inconfortables avec la fonctionnalité de partage de ressources du Cloud, dans laquelle les données et les applications partagent des ressources informatiques communes. Par conséquence, elles ne disposent pas du contrôle de sécurité complet. D'autre coté, les services de stockage de données en Cloud consistent à offrir aux utilisateurs la possibilité de stocker des données sur un serveur distant tel que le serveur est connecté à Internet. Un utilisateur pourra créer, récupérer, modifier, traiter ou stocker, en passant par des différents réseaux de connexion. Toutefois, la question de la sécurité des données doit encore être résolue de manière adéquate.

Lorsqu'on parle de la sécurité des données il faut toujours parler sur le triplet confidentialité, intégrité et disponibilité. Dans les conditions d'un environnement Cloud, la confidentialité des données devient plus difficile et conflictuelle. Cela est dû en grande partie du fait que les utilisateurs chargent leurs données sur des serveurs distants, qui sont contrôlés et gérés par des fournisseurs de Cloud non confiants. Les données stockées dans le Cloud sont confidentielles et peuvent être sensibles à l'utilisateur et sont susceptibles d'être exploitées par un tierce non

autorisés. Actuellement, la plupart des utilisateurs du Cloud Storage ne protègent pas leurs données, d'autres utilisent des contrats SLA (Service Level Agreement). Ces Contrats sont basés généralement sur la confiance et la réputation du fournisseur. Le reste des clients utilisateurs des solutions de sécurité, sont orientés vers les techniques de chiffrement. Il est communément admis que le chiffrement des données côté client est une bonne alternative pour réduire ces problèmes de confidentialité des données. Ainsi, l'utilisateur conserve les clés de déchiffrement hors de portée du fournisseur de Cloud. Néanmoins, ces approches soulèvent plusieurs problèmes essentiels de gestion, tels que le stockage et le maintien de la disponibilité des clés chez le client, la possibilité de traitement de données chiffrées et les caractéristiques des résultats de traitement. De plus, la préservation de la confidentialité devient plus compliquée avec le partage de données flexibles entre un groupe d'utilisateurs.

Les solutions de sécurité existantes qui sont fournis par les fournisseurs de Cloud ne répondent pas aux préoccupations raisonnables des utilisateurs en matière de sécurité et spécialement pour les questions suivantes :

- Comment l'utilisateur peut-il garantir que ses données restent confidentielles dans le Cloud?
- Comment amener les utilisateurs à faire confiance aux fournisseurs de services?
- Quelle sont les Clouds qui assurent la confidentialité dans un environnement multi-locataire?
- Existe-t-il un moyen de s'assurer que les données de l'utilisateur ne sont pas corrompues après la protection? Est-ce que le traitement sur ces données reste sûr et sans aucune difficulté?
- Est-il possible pour le client et le fournisseur de se faire confiance mutuellement et de coopérer pour la réussite de l'adoption d'un projet en Cloud?

Le traitement de ces problèmes de sécurité a renforcé la confiance dans le système informatique en Cloud et attiré donc de nouveaux clients. Mais pour répondre aux questions précédentes, quelques autres pointes doivent être posées :

- Quels sont les principaux obstacles de sécurité pour l'adoption du Cloud Computing?
- Comment ces systèmes de protection fonctionnent et comment sont-ils protégés contre les attaques potentielles?
- Comment pouvons-nous faire confiance aux garanties de sécurité réclamées par le système?
- Comment peut-on concevoir une architecture générique permettant de réduire les risques liés à la confidentialité dans le Cloud?
- Quels facteurs et qualités de service influent sur la fiabilité et la sûreté de sécurité?

Toutes ces faiblesses nous ont motivé à réfléchir à des solutions qui permettent aux utilisateurs de sécuriser leurs données pour éviter toutes utilisations malveillantes.

4. Objectifs

4.1. Objectif général

L'objectif général de cette thèse est de proposer une solution complète de sécurité des données dans un environnement Cloud Computing. Il est nécessaire de comprendre puis de développer un système de gestion de sécurité des données dans le Cloud afin d'améliorer la confidentialité des données stockées dans le Cloud. Pour atteindre cet objectif général, un certain nombre de sous-objectifs sont envisagés :

4.2. Objectifs spécifiques

Objectif A – Le premier objectif est l'étude des défis et challenges de la sécurité des données dans les environnements de Cloud Computing puis l'analyse et la comparaison des approches de sécurité courantes.

Objective B – Le deuxième objectif est la conception et l'implémentation d'un système de gestion du stockage des données en préservant la confidentialité dans le cadre de la sécurité du Cloud. Le système proposé doit fournir des services de stockage sécurisés et résoudre le problème lié à la confidentialité de données stockées dans le Cloud. En effet, on va proposer un mécanisme efficace pour le téléchargement, le chargement et le traitement sécurisés dans le Cloud.

Objectif C – Le troisième objectif est de proposer un système intégré et sécurisé de stockage dans le Cloud de la phase de connexion jusqu'à la phase d'achèvement, tout en améliorant le transfert des données entre les intervenants. En effet, les mécanismes de sécurité proposés devraient assurer à la fois la robustesse, la sûreté et l'efficacité, à savoir la prise en charge du contrôle d'accès flexible, le traitement efficace et illimité sur les données des utilisateurs.

Objectif D – Le quatrième objectif est l'implémentation des techniques proposées à l'aide de normes et de schémas largement déployés, et en validant la faisabilité et l'impact sur l'architecture Cloud.

Objectif E – Le cinquième objectif consiste à fournir des preuves mathématiques de la validité et de la justesse des schémas proposés.

5. Contributions

Lors de la définition des solutions pour la sécurité des données stockées dans le Cloud, nous avons considéré les aspects suivants: la facilité de déploiement, la robustesse, la rapidité de traitement, la performance (délai d'exécution) et la flexibilité de fonctionnement. En considérant les objectifs cernés dans la section précédente, nous résumons les contributions de cette thèse comme suit:

Contribution 1 :

- La proposition d'un nouvel algorithme et d'un schéma de chiffrement plus efficaces, flexibles et garantissant la confidentialité et l'actualisation de notre politique de sécurité. Les données doivent être chiffrées avant de les envoyer au Cloud. Nous avons utilisé un algorithme de chiffrement complétement homomorphique garantissant l'ordre afin de bénéficier de ses avantages en termes de robustesse, de rapidité, et de flexibilité de traitement. Le modèle proposé permet de sécuriser les données et de les protéger contre les attaques. Ce travail a donné lieu à une communication dans une conférence internationale [1].
- La proposition d'une nouvelle architecture Cloud sécurisée pour la gestion de système de stockages des données dans le Cloud. En utilisant dans ce cas le schéma proposé dans [1]. Ce travail fait partie de [1].
- Développement d'une proposition améliorant notre première approche [1], la valeur ajoutée de cette évolution consiste à sécuriser davantage l'architecture, renforcer sa fiabilité et le rendre plus confiée. Ce travail a été soumis comme publication dans un journal international [2].

Contribution 2 :

- Afin d'améliorer notre première approche, nous avons développé une deuxième solution fiable, autonome et rapide pour sécuriser les données stockées dans le Cloud. Cette solution est basée sur les systèmes multi agents. Nous avons proposé un système multi agents basé sur le chiffrement homomorphique qui permet d'assurer la confidentialité. Nous présentons ainsi un nouveau système hybride de communication qui utilise les techniques de chiffrement symétriques et asymétriques assurant la confidentialité dans cette phase. Ce travail a fait l'objet d'une publication internationale [3].
- Développement d'une architecture Cloud basée sur les système multi agents améliorant notre approche présentée dans [3]. Dans ce travail nous avons tenté de traiter le problème de sécurité en utilisant le concept de la virtualisation. Dans cette tentative, nous avons parlé sur la distribution des traitements et la concurrences des communications. Le travail a donné aussi naissance à une publication internationale [4].

Pour résumer, les travaux réalisés au cours de cette thèse ont fait l'objet de communications et de publications suivantes :

Les conférences

[1] Mohammed Amine Yagoub, Okba Kazar, Abdelkader Laouid, Ahcène Bounceur, Reinhardt Euler et Muath AlShaikh, « An Adaptive and Efficient Fully Homomorphic Encryption Technique», The 2nd International Conference on Future Networks & Distributed Systems, June 26-27, 2018, Amman, Jordan.

[4] Mohammed Amine Yagoub, Okba Kazar, Abdelkader Laouid et Kerthio Ismail, «Intelligent Cloud protection based on Multi Agent System Approach Using Advanced Cryptographic Algorithm».

Les publications

[3] Mohammed Amine Yagoub, Okba Kazar et Mounir Beggas, « A multi-agent system approach based on cryptographic algorithm for securing communications and protecting stored data in the Cloud-computing environment ». L'article est accepté dans le journal : « International Journal of Information and Computer Security ».

[2] Mohammed Amine Yagoub, Okba Kazar et Abdelkader Laouid, « An Efficient and Adapted Fully Homomorphic and Order Preserving Encryption in Cloud Computing».

6. Organisation de la thèse

Cette thèse comporte quatre chapitres. L'introduction générale présente le contexte et la problématique de la recherche. Elle identifie également les motivations et les objectifs de la thèse et présente brièvement les principales contributions.

Dans le premier chapitre, on présente un état de l'art sur le Cloud Computing. On donne une introduction sur le Cloud Computing, y compris la définition, ses caractéristiques, les types des services Cloud, les modèles de déploiement ainsi que les principaux défis et challenges de recherche dans le Cloud.

Le deuxième chapitre aborde les principes, les exigences, les normes, les stratégies et la gestion de la sécurité du Cloud Computing. Les travaux et la littérature connexes sur les problèmes liés à la sécurité et à la confidentialité et les préoccupations des données dans le Cloud sont discutés. En outre, ce chapitre décrit et traite en détail les problèmes de sécurité, de confidentialité, de contrôle des données et les menaces de la sécurité dans le calcul.

Le troisième chapitre décrit en détail notre première contribution intitulée « Une technique complétement homomorphique et préservant de l'ordre adapté au Cloud Computing ». Au début nous décrivons une taxonomie des travaux relatifs aux techniques de la sécurité dans le Cloud. Ensuite, nous détaillions les solutions proposées. Enfin, nous présentons l'implémentation et l'analyse des résultats vis-à-vis des travaux connexes.

Le quatrième et dernier chapitre décrit en détail la deuxième contribution intitulée « Une approche SMA pour sécuriser les communications et les données dans le Cloud ». Au début nous décrivons une explication sur les systèmes multi agents et les travaux relatifs aux solutions SMA de la sécurité dans le Cloud. Ensuite, nous détaillons les solutions proposées. Enfin, nous présentons l'implémentation et l'analyse des résultats des solutions proposées.

Finalement, la conclusion générale fait une synthèse des contributions globales et met en évidence les perspectives de cette recherche.

Chapitre I

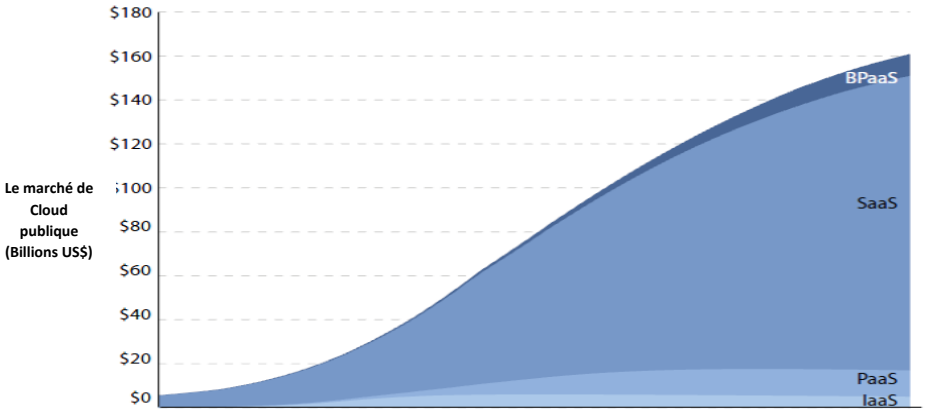
Cloud Computing

I.1 Introduction

Il n'y avait pas une date précise à laquelle nous pourrions dire que le Cloud Computing était né! Cette notion fait référence à un dessin de nuage, tel que l'on a l'habitude de l'utiliser lorsqu'on veut représenter Internet. Cette représentation, très populaire dans les années 60, disparu au milieu des années 70 jusqu'à l'apparition de notion de l'ASP « Application Service Provider » qui a aussi sa part dans l'historique du Cloud Computing. Une ASP désigne une application fournie comme un service, ce que l'on appelle aujourd'hui SaaS « Software as a Service ». C'est une classe très importante dans la terminologie actuelle du Cloud Computing. Ensuite, vers la fin des années 90, ce concept a pris de l'importance avec la parution du Grid Computing [5] et l'évolution de la technologie de la virtualisation, lancé la première fois dans les travaux de IBM [6].

La technologie Cloud Computing trouve ses origines dans l'histoire de l'informatique au cours de l'année 2006 [7] avec la parution d'Amazon EC2 (Elastic Compute Cloud). Puis, en 2009 survint la vraie explosion du Cloud avec l'arrivée des sociétés comme IBM Smart Business Service (de IBM), Google App Engine (de Google), Microsoft Azure (de Microsoft), Sun Cloud (de Sun) et Ubuntu Enterprise Cloud (de Canonical Ltd). Dans [8], Forrester a mené une étude sur le marché du Cloud Computing que s'élevait à environ 5,5 milliards de dollars en 2008, il devrait atteindre plus de 150 milliards de dollars d'ici 2020, comme illustré en Figure I-1.

De nos jours, le mot « Cloud Computing » est devenu de plus en plus populaire en informatique. Il est très courant d'entendre parler sur le Cloud Drive, de la base de données Cloud, du serveur Cloud et de la sécurité de Cloud. Apparemment, la signification de « Cloud Computing » est insuffisante, c'est une nouvelle technologie informatique agrégée qui s'étend rapidement de la recherche dans des petites régions de développement et d'utilisation aux régions de grande échelle. Evidemment, la popularisation du « Cloud Computing » n'est pas une coïncidence, mais est un besoin du marché de l'Internet. En outre, cette nouvelle technologie constituera le fondement d'Internet de la prochaine génération et initiera le nouveau modèle de services Internet.



	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
BPaaS ($)	0.15	0.23	0.35	0.53	0.80	1.26	1.95	2.93	4.28	6.00	7.66	9.08	10.02
SaaS ($)	5.56	8.09	13.40	21.21	33.09	47.22	63.19	78.43	92.75	105.49	116.39	125.52	132.57
PaaS ($)	0.05	0.12	0.31	0.82	2.08	4.38	7.39	9.80	11.26	11.94	12.15	12.10	11.91
IaaS ($)	0.06	0.24	1.02	2.94	4.99	5.75	5.89	5.82	5.65	5.45	5.23	5.01	4.78

Figure I-1 : Prévisions de la taille du marché du Cloud Computing publique [8]

Ainsi, pour comprendre l'évolution du « Cloud Computing », il est nécessaire d'avoir une compréhension des concepts de base de Cloud Computing. Néanmoins, l'idée du Cloud Computing est encore floue pour les non-spécialistes en informatique. Alors, quel est exactement le Cloud Computing? Quels sont les points clés pour construire un Cloud réussi? Quels sont leurs avantages dans notre vie? Y-a-t-il des problèmes de sécurité et comment pouvons-nous les résoudre?

I.2 Définition

L'ambiguïté qui entoure le concept du Cloud Computing rend à trouver plusieurs définitions pour ce concept. Aujourd'hui, le Cloud Computing est le plus important domaine dans l'industrie informatique. Avec ses nouveaux aspects et capacités qui ont été proclamés, le Cloud Computing est un modèle en évolution rapide. Le Cloud, pas à pas, est devenu la prochaine évolution de l'histoire informatique et modifie radicalement la manière dont une entreprise gère ses systèmes informatiques.

Beaucoup de chercheurs dans les domaines industriels et universitaires ont tenté de donner une définition au « Cloud Computing » et de préciser quelles sont les caractéristiques uniques qu'il présente :

Selon le "National Institute of Standards and Technology (NIST) " [9], "*Le Cloud Computing est un modèle qui permet d'accéder rapidement à un pool de ressources informatiques mutualisées, à la demande (serveurs, stockage, applications, bande passante, etc.), sans forte interaction avec le fournisseur de service* ".

Selon IBM [10], "*le Cloud Computing, souvent appelé simplement « nuage », est une fourniture de ressources informatiques à la demande, des applications aux centres de données, en passant par l'Internet et le paiement à l'utilisation* ".

Selon Gartner [11], *"le Cloud Computing est un style d'informatique dans lequel des moyens informatiques disponibles, évolutives et élastiques sont fournies sous forme de service aux utilisateurs externes utilisant les technologies Internet. L'opération de fourniture est basée sur cinq attributs: la multi-location (ressources partagées), une extensibilité massive, l'élasticité, le paiement à l'utilisation et l'auto-provisionnement des ressources "*.

En analysant ces définitions on constate que la définition du Cloud tourne au tour de ses deux principaux groupes des caractéristiques: (1) Du point de vue fournisseur de services Cloud, on trouve l'élasticité, mis à jour automatique, évolutivité massive, l'auto-approvisionnement de ressources, et les ressources partagées. (2) et du point de vue utilisateur final du Cloud, on trouve l'instantanéité, la disponibilité, l'accès à la demande, le payement à l'utilisation.

I.3 Caractéristiques du Cloud Computing

Selon les définitions précédentes du Cloud Computing, il existe cinq caractéristiques essentielles qui sont étendues par tous les chercheurs du Cloud [9, 10, 11] :

I.3.1 Accès libre à la demande

Le Cloud Computing permet aux clients de consommer des ressources informatiques comme des services, telles que des serveurs, capacité de stockage ou de calcul, plateforme de développement, etc. à travers les réseaux, selon ses besoins, de manière simple et flexible, sans nécessiter d'interaction humaine avec le fournisseur de services. Le Cloud donc, est un système autonome qui vise à construire des systèmes informatiques capables d'être auto-administrés en s'adaptant à des changements internes et externes. Le matériel, le logiciel et les données Cloud peuvent être reconfigurés, orchestrés et consolidés de façon automatique en une seule image qui sera fournie à l'utilisateur [12].

I.3.2 Service doit être accessible via un réseau

Les services proposés aux utilisateurs par le Cloud doivent être disponibles sur le réseau et accessibles à travers des mécanismes standards favorisant l'utilisation des plateformes hétérogènes, par exemple ordinateurs portable, poste de travail, tablettes, téléphones mobiles [9].

I.3.3 Ressources partagées

Le Cloud Computing est un nouveau modèle informatique basé sur un modèle commercial dans lequel les services et les ressources sont agrégées et mises à disposition aux consommateurs sur un modèle multi-locataires. Généralement, les consommateurs n'ont aucune connaissance ni aucun contrôle de l'endroit exact où sont stockés les ressources fournies [9]. En d'autres termes, contrairement au modèle traditionnel, les ressources physiques ou virtuelles sont attribuées et réaffectées de manière dynamique en fonction de la demande.

I.3.4 Élasticité rapide

Les ressources informatiques peuvent être rapidement augmentées et diminuées selon les besoins. De plus, ces ressources peuvent être complètement libérées pour d'autres utilisations lorsqu'elles ne sont plus nécessaires. Contrairement aux infrastructures traditionnelles qui ont des capacités fixes et limitées, le Cloud Computing permet de redimensionner massivement les ressources (bande passante, espace de stockage, etc.). Cette élasticité dérive des services Cloud évolutifs pour satisfaire toute les demandes croissantes des utilisateurs. Cette évolutivité doit se faire d'une façon automatique et en cours d'exécution. L'évolutivité linéaire est une technique plus utilisée pour assurer cette caractéristique où le système découpe les principales taches en un ensemble de petits morceaux, et de les distribuer sur l'infrastructure virtuelle du Cloud [13].

I.3.5 Service mesuré

Le Cloud assure la transparence entre le fournisseur et le consommateur du service, en fournissant des outils permettant aux consommateurs du Cloud de contrôler et de superviser l'utilisation de leurs ressources.

En plus de ces cinq caractéristiques, il y a d'autres caractéristiques dont nous citons les plus pertinentes :

I.3.6 Paiement à l'usage

Toute offre de Cloud comprend un paiement qui s'effectue chaque fois qu'on utilise un service. Le fournisseur est capable de compter de façon précise la consommation (en quantité et en durée) des différents services (stockage, CPU, bande passante…). Cela lui permettra de facturer à l'utilisateur selon sa réelle consommation. Ce mode de facturation permet tout d'abord d'économiser les coûts de mise en service, les frais d'investissement et les frais d'exploitation des entreprises [14].

I.3.7 Basé Virtualisation

Un système Cloud Computing est un système complètement virtuel. La virtualisation désigne l'abstraction des détails du matériel physique et fournit des ressources virtuelles pour les applications de haut niveau. Elle permet la mise en commun des ressources dans des centres de données. Il existe différentes formes de virtualisation qui sont: la virtualisation de systèmes d'exploitation, la virtualisation de stockage, la virtualisation de base de données, la virtualisation d'applications et la virtualisation de matériels. La virtualisation constitue le support du Cloud Computing, parce qu'elle offre la possibilité de mettre en commun des ressources informatiques à partir de clusters de serveurs, et ainsi d'attribuer dynamiquement des machines virtuelles aux applications à la demande [15, 16].

I.3.8 Basé SLA (Service Level agreement)

Le SLA définit dans un contrat entre un fournisseur de services et un client des politiques, telles que les paramètres de livraison, la maintenabilité, les niveaux de disponibilité, l'exploitation, la performance ou autres attributs du service, comme la facturation, et même des sanctions en cas de violation du contrat [17]. Par exemple, le contrat SLA entre le fournisseur de services Internet et l'opérateur télécom, généralement, précise le niveau de service fourni aux clients en déterminant le temps moyen entre deux défaillances, le temps moyen de réparation, le temps moyen de récupération, et identifiant quelle partie est responsable de signaler les erreurs ou payer les frais de réparation. Avec les services Cloud, un client peut négocier le niveau de service qu'il lui convient et il doit payer pour cela selon des garanties de QoS [18, 19].

I.3.9 Simplicité, flexibilité, fiabilité et tolérance aux pannes

Les environnements Cloud doivent garantir la qualité de service pour les utilisateurs, tels que la flexibilité, la simplicité, la fiabilité et la tolérance aux pannes. L'allocation et l'utilisation des ressources Cloud doivent être simples. Idéalement, elles doivent se faire à travers des interfaces efficaces et génériques. Ces services, sont aussi destinés à supporter des lourdes de taches comme les petites charges de travail dans des environnements qui tirent parti de la redondance intégrée du grand nombre de serveurs qui les composent en permettant des niveaux élevés de disponibilité et de fiabilité [19, 20, 21].

I.3.10 Sécurité efficace

La sécurité, dans tous les systèmes, joue un rôle très important, surtout dans le cas où les données sensibles résident dans le Cloud. La perte ou l'accès illégal sur les données peuvent donner des effets mal entendus spécialement pour les données. Pour cette raison, les chercheurs et les fournisseurs de Cloud prennent en considération ce point par l'introduction des architectures de sécurité, politiques de chiffrement et d'authentification très efficaces [22].

Le Cloud est associé avec plusieurs technologies telles que le Grid Computing qui est un paradigme de calcul réparti qui coordonne des ressources géographiquement distribuées et autonomes pour réaliser un calcul commun et intensif en utilisant un partage dynamique des ressources entre des participants, des entreprises et des organisations. La similarité de Grid Computing au Cloud Computing est que les deux basant sur le concept d'offrir les ressources sous forme de services. De plus le Cloud exploite la stratégie de l'utility Computing qui est une délocalisation d'un système de calcul ou de stockage en adoptant un système de tarification basée sur l'utilité. A cause de cette stratégie économique, la fourniture des ressources à la demande et le paiement à l'usage, maximise l'utilisation des ressources et minimise leur coûts d'exploitation [23].

L'environnement du Cloud Computing est composé principalement de cinq acteurs majeurs :

1) Le fournisseur du Cloud qui a comme activité l'allocation, la gestion et l'orchestration des ressources, il offre ces services tout en assurant le bon niveau de sécurité.

2) L'utilisateur du Cloud qui consomme le service fourni par le fournisseur Cloud. Il peut être un développeur ou un utilisateur final donc il peut être un groupe de personnes, une personne, des petites et moyennes entreprises, des gouvernements ou des multinationales [24].

3) Le fournisseur de réseau qui est l'intermédiaire entre l'utilisateur et le fournisseur. Il garantit principalement la connectivité entre les ressources Cloud et la liaison entre les parties de l'écosystème Cloud. Cet acteur joue un rôle plus important en offrant des fonctionnalités avancées dans le réseau. Ces fonctionnalités sont basées sur des SLA [24].

4) Le quatrième acteur c'est le courtier Cloud. Il est un médiateur qui négocie la relation entre les fournisseurs Cloud et les consommateurs du Cloud. Il offre des services qui simplifient les tâches de gestion des utilisateurs du Cloud. Ce dernier peut demander les ressources Cloud auprès du Cloud Broker au lieu du Cloud Provider directement.

5) Enfin, L'auditeur Cloud s'occupe de l'audition et la vérification des services Cloud. Il évalue les services offerts par le fournisseur Cloud, le courtier Cloud et le fournisseur de réseau Cloud du point de vue performances et sécuritaires et en vérifiant que les fournisseurs respectent bien les SLA qu'ils proposent [24, 6].

I.4 Types de services du Cloud

Le Cloud Computing permet aux utilisateurs ou entreprises de consommer, à la demande, des services informatiques. Ces services peuvent se présenter, comme illustré dans la Figure I-2, sous plusieurs formes selon le type du service correspond au niveau de responsabilité dans la gestion des couches de l'environnement informatique standard que ce soit par les utilisateurs ou par les fournisseurs [9]. L'environnement informatique standard est composé par des couches partent du bas niveau (le matériel physique) et autres de haut niveau (les applications à utiliser). Ces couches sont: Environnement d'exécution, Données, Applications, Intergiciel, Environnement de développement, Système d'exploitation, Virtualisation, Calcul, Réseau et Stockage. Dans l'environnement Cloud, contrairement au environnement traditionnel, l'utilisateur n'a plus en charge la totalité des couches et en fonction du niveau des sous-ensembles de couche nous distinguons le type de service.

Selon NIST [9] et comme illustré dans la Figure I-2, il y a principalement trois types de services Cloud Computing qui sont l'infrastructure en tant que service (IaaS), la plate-forme en tant que service (PaaS) et le logiciel en tant que service (SaaS) que nous allons détailler par la suite.

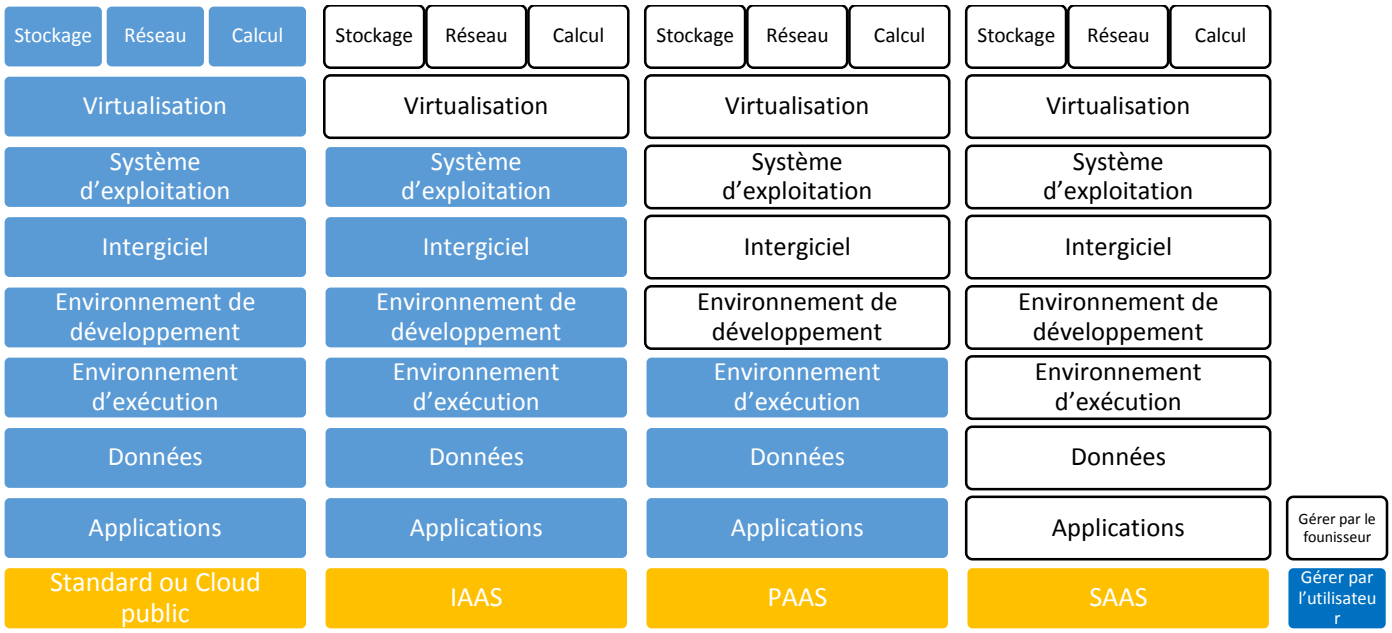


Figure I-2 : Types de service Cloud Computing

I.4.1 IaaS (Infrastructure as a Service)

L'IaaS ou l'infrastructure en tant que service offrent des plateformes de virtualisation configurables qui met à la disposition des utilisateurs des ressources informatiques prêtes à l'emploi telles que des équipements réseau, des serveurs, de l'espace de stockage, etc. [25]. Ce modèle de service est vu comme une abstraction d'un centre de calcul ou de stockage de données sur lequel les utilisateurs déposent leur environnement de production (système d'exploitation, intergiciel, logiciel…). Les serveurs virtuels peuvent êtres démarrer ou arrêter à la demande des utilisateurs, et cela permet de concentrer davantage sur la réalisation et le développement des applications sans avoir à se préoccuper de l'acquisition de serveurs ou de la gestion de l'infrastructure.

A ce niveau d'infrastructure, on peut trouver un ensemble important d'outils utilisés qui ont pour but de fournir une abstraction du stockage ou du calcul pour une approche orientée virtualisation. Parmi ces outils, le logiciel open source ***Ceph*** [26]. Il assure un accès de manière fiable et autonome aux objets de stockage distribué selon l'algorithme CRUSH (Controlled Replication Under Scalable Hashing). Il fournit une gestion dynamique et distribuée des méta-données et une opération de stockage dans un OSD (Object Storage Devices). Chaque OSD utilise un journal d'écriture selon trois stratégies de réplication qui sont: (1) "*primary-copy*" où le premier OSD transfère les écritures dans les autres OSDs et l'opération de lecture ne sera autorisée que si le dernier OSD a envoyé son acquittement, (2) "*chain*" où les écritures des objets sont effectuées séquentiellement et l'opération de lecture sera autorisée une fois que le dernier objet a été écrit dans l'OSD correspondant et (3) "*splay replication*" telque la moitié des objets sont écrits de façon séquentielle, le reste étant effectué en parallèle [27]. Le journal d'écriture est pour accélérer les

opérations de lecture et d'écriture en réunissant plusieurs petites opérations et en les envoyant de façon asynchrone vers le système de fichiers. Le journal peut être un dispositif, une partition ou un fichier. Le point fort de *Ceph* est qu'il est un système de stockage efficace, robuste, évolutif et fortement versatilité. Par contre, la confidentialité et l'intégrité des données n'y sont pas implémentées et restent en question.

HDFS (Hadoop Distributed File System) le système de distribution des fichiers Hadoop est un outil Open Source développé par Apache [28, 29]. Il a été mis en avant par des grands acteurs du web tels que Yahoo! et Facebook. Il est structuré pour comporter de façon fiable de très gros fichiers dans des grands serveurs. Les données sont partagées en blocs, et ensuite répliquées de façon asynchrone et distribuées dans plusieurs DataNodes qui est géré par un service central (généralement appelé aussi maître) appelé Namenode. Les serveurs HDFS détectent des défaillances par l'envoi des messages et un DataNode est déclaré défaillant si ne répond pas durant dix minutes. Ces réponses contiennent également des informations statistiques permettant d'assurer l'équilibrage de charge.

Les utilisateurs accèdent aux données en utilisant leur URLs dans l'espace des noms et contactent le NameNode pour déterminer où sont stockés les blocs de données. Dans l'opération de la lecture dans HDFS, l'utilisateur récupère la liste des DataNodes sur lesquels les blocs sont situés. Il va ensuite chercher les blocs sur les différents DataNodes. Pour l'écriture, l'utilisateur récupère la liste des DataNodes sur lesquels placer les différents répliquas. Ensuite il écrit le bloc sur tous les DataNodes. HDFS est amélioré pour les lectures récurrentes mais il est sensiblement moins performant pour des écritures concurrentes.

S3 (Simple Storage Service) est un service de stockage de données incluses dans AWS (Amazon Web Services). L'objectif de S3 est de fournir une solution de stockage hautement disponible, à faible coût et avec un modèle de facturation de type paiement à l'usage, avec une configuration facile des droits d'accès aux fichier et avec une possibilité de chiffrement du contenu pour des raisons de sécurité. Les données stockées dans ce service sont organisées de manière versionnés sur deux niveaux d'espace de nommage et peuvent être répliqués automatiquement sur plusieurs datacenters AWS. S3 fournit trois services de stockage pour l'accès aux données qui sont: SOAP (Simple Object Access Protocol), REST (Representational State Transfer) et BitTorrent [30]. L'interface S3 est utilisée par divers plateformes de gestion de Cloud Computing tels qu'OpenStack et Cloudstack [31].

I.4.2 PaaS (Platform as a Service)

La PaaS (ou la plate-forme en tant que service) est un modèle de Cloud correspond principalement à des environnements de développement qui met à disposition des plates-formes d'exécution, de déploiement et de développement d'applications. Pour cela, PaaS fournit un niveau d'abstraction supplémentaire par rapport à IaaS sauf que l'utilisateur n'a plus en charge que les couches de données et d'applications. Une Plateforme de développement sur le Cloud permet d'utiliser et réutiliser un large spectre d'outils, de composant logiciels, et de blocs de codes ce qui

permet de réduire considérablement les couts de déploiement des applications avec une grande rapidité de développement et d'hébergement [32]. Les exemples typiques du PaaS sont: Google App Engine, ConPaaS, Windows Azure, Elastic Beanstalk, etc.

Google App Engine* (*GAE) est une plate-forme de développement et d'hébergement d'applications basée sur les serveurs de Google. Elle supporte les applications écrites en Python, Java PHP, Node.js et Go. Elle vise à éliminer les tâches d'administration système et de développement pour faciliter l'écriture d'applications évolutives. Elle fournit également plusieurs types de services parmi lesquels Big Query (Datawarehouse), Bigtable, Cloud Datastore, stockage de données (MySQL, NoSQL et stockage orienté objet), etc. Les applications App Engine sont faciles à construire, faciles à maintenir, et supportent les besoins croissants de stockage de données. L'inconvénient majeur de ces applications est que les développeurs doivent utiliser les API propriétaires d'AppEngine, ce qui restreint la portabilité des applications vers d'autres infrastructures [33].

ConPaaS est un environnement d'exécution open source qui vise à simplifier le déploiement d'applications en Cloud. Dans ConPaaS, une application est définie comme une composition d'un ou de plusieurs services. Chaque service est caractérisé par l' autogéré et la flexibilité que signifie qu'elle peut se déployer, surveiller ses propres performances et augmenter ou réduire sa capacité de traitement sur le Cloud selon l'provisionnement dynamique des ressources [34]. Parmi les services de ConPaaS que dispose il existe : les services d'hébergement des applications PHP et JSP, un service MapReduce, les services de stockages MySQL ou NoSQL basé sur scalarix, , un service XtreemFS offrant un système de fichiers distribué et répliqué, un service HTC (***H****igh-****T****hroughput* ***C****ondor*) etc. ConPaaS se présente donc comme une plate-forme Cloud en tant que service qui nécessite une infrastructure de niveau inférieure pour s'exécuter, mais son avantage major est qu'il supporte les deux IaaS : EC2 (*Amazon Elastic Compute Cloud*) et OpenNebula [35].

Windows Microsoft Azure est un ensemble sans cesse croissant de services Cloud destinés à aider les organisations de construire, de gérer et de déployer des applications sur un énorme réseau mondial en utilisant leurs infrastructures et outils favoris [36]. Les fonctions et services fournis sont exposés à l'aide de protocoles REST ouverts dans des bibliothèques disponibles pour plusieurs langages de programmation qui sont publiées sous licence open source et hébergées sur GitHub. Microsoft Azure est toujours en améliorations, notamment à Visual Studio Online, avec des outils de collaboration pour les développeurs et pour la gestion des cycles de vie des applications. Les plus importantes services disponibles : (Azure Active Directory) pour le contrôle d'accès, (Azure API Management) de publication des interfaces (API), (Azure Backup) pour gérer les sauvegardes dans le Cloud, etc [37].

Amazon Web Services (AWS) propose plus d'une centaine de services, chacun d'entre eux étant spécialisé dans un ensemble de fonctionnalités spécifique. ***Elastic Beanstalk*** est une solution PaaS propriétaire parmi ces services de AWS. Elle permet de déployer et gérer rapidement des

applications dans le Cloud sans préoccuper de l'infrastructure qui les exécute. Elle simplifie le déploiement et le passage à l'échelle des applications et services web développés avec Java, .NET, Python, PHP, Node.js, Go, Ruby et Docker sur des serveurs tels qu'Apache, Passenger, IIS et Nginx [38]. Pour utiliser Elastic Beanstalk, il faut créer une application, charger une version d'application sous la forme d'un bundle source d'application sur le Cloud, puis fournir des informations sur l'application. Si l'application utilise des choix de stockage ou de base de données, il peut utiliser Amazon RDS, Amazon DynamoDB, Amazon SimpleDB, Microsoft SQL Server, Oracle, IBM DB2 ou Informix. Comme dans le cas de Google AppEngine, Elastic Beanstalk est uniquement disponible dans les Clouds d'Amazon [39].

I.4.3 SaaS (Software as a Service)

Les services Cloud Computing de type SaaS (ou le logiciel en tant que service) correspondent tout simplement à un modèle où le matériel, l'hébergement, le framework d'application et le logiciel sont dématérialisés et offertes à la demande sous forme des services [40, 41]. L'utilisateur n'a rien à gérer ou contrôler sauf quelques configurations bien spécifiques, et c'est le fournisseur qui s'occupe de la gestion des ressources nécessaires [42]. Les services sont hébergés, dans son propre centre de données, comme c'est par exemple le cas pour les outils ERP, serveurs web, outils collaboratifs, CRM, serveurs de messagerie (Yahoo Email, Gmail, …), Google Apps, Dropbox, office 365 (outil collaboratif), Salesforce CRM, Google Documents, Facebook, Twitter, MobileMe, Zoho, etc. L'inconvénient majeur de ce modèle est que le fait de confier les données sur des machines virtuelles appartenant à un fournisseur Cloud soulève des problèmes de confidentialité et de sécurité supplémentaires. Aussi, les utilisateurs sont limités aux qu'est proposé par le fournisseur. De plus, ce modèle pose également de vraies questions sur la pérennité du fournisseur.

I.4.4 XaaS (Anything-as-a-Service)

L'acronyme aaS se développe et devenu référence à l'expression XaaS Everything as a Service (Tout en tant que Service). Ceci la lettre X fait référence au mot « everything » (tout) ou «anything» (n'importe quoi). Bien que les trois types (IaaS, PaaS et SaaS) soient la base de distinction du type de service, la notation XaaS a été utilisé ailleurs pour caractériser des services et ressources vues comme un sous-ensemble des trois types de base. Parmi ces abréviations, nous citons les plus communs et réussies: SecaaS Security as a Service, BaaS Backup as a Service, CaaS Communication as a Service, DaaS Data as a Service, DBaaS Database as a Service, MaaS Monitoring as a Service, etc [43, 44].

Les services DBaaS ***Amazon RDS***, ***Microsoft SQL Azure et Google Cloud SQL*** sont connus sur le nom Relationnel Cloud. Elles permettent d'alléger une grande partie de la charge opérationnelle liée au provisionnement, à la configuration, à l'évolution, à l'amélioration des performances, à la sauvegarde, à la confidentialité et au contrôle d'accès des utilisateurs de la base de données exploitée par le service. Malgré ces efforts les problèmes de la multi-location,

l'évolutivité élastique et la confidentialité des bases de données restent en question [45]. Face à ces questions et à une croissance de façon exponentielle des besoins en termes de charge et de volumétrie de données, le NoSQL (Not only SQL) a vu le jour.

Les solutions NoSQL répondent aux propriétés CAP (Consistency, Availability et Partion-Tolerance) énoncé par Eric Brewer [46]. ***Cassandra*** [47, 48, 49] qui fait partie des bases de données NoSQL orientée colonnes, est utilisée par Amazon et dans le modèle BigTable de Google. ***SimpleDB*** est un autre système de gestion de base de données NoSQL qui est écrit en Erlang par Amazon. *SimpleDB* supporte le modèle de cohérence éventuelle basé sur la réplication asynchrone [47]. Autre exemple, c'est ***MongoDB*** qui est une base de données NoSQL orientée documents. Il est écrit en C++ sur un système à données largement réparties qui permet de manipuler des objets structurés au format BSON (JSON binaire), similaire au service Google App Engine de Google. Il est devenu plus utilisés, notamment pour les sites web de Craigslist, eBay, Foursquare, SourceForge.net, Viacom, etc. [50].

Le projet ***CloViS*** est un intergiciel capable de faire à la fois du SaaS, PaaS et IaaS avec un choix complet de modalités d'accès aux données. Il est implémenté d'une couche de virtualisation de stockage en s'appuyant sur l'expertise acquise dans la virtualisation du stockage appliquée aux grilles informatiques. CloViS fournit un service de stockage complet, utilisable dans tous les environnements classiques du Cloud. Au niveau PaaS, il présente des méthodes d'accès standard aux blocs de données sous le protocole iSCSI. En plus, au niveau IaaS, il utilise des éléments particuliers pour guider le stockage. Ce niveau est divisé en deux fonctions bien distinctes : le stockage des machines virtuelles qui est temporaire, et le stockage de type « bloc »qui doit stocker de façon sécurisée et pérenne les données utilisateur des machines virtuelles [51].

I.5 Modèles de déploiement du Cloud Computing

Les modèles de Cloud Computing sont distingués en fonction de l'utilisation des ressources physiques par les intervenants. Les ressources peuvent être localisées chez l'utilisateur ou chez un fournisseur, peuvent être partagées ou non, et peuvent être pour des entreprises ou pour des autre types d'utilisateurs. Pour cela, le Cloud offre quatre modèles ou typologies de déploiement.

I.5.1 Cloud privé

Le modèle privé de déploiement est destiné aux entreprise privées qui mettent l'ensemble des ressources à la disposition exclusive et de les hébergé dans ces entreprises. Ce modèle est un ensemble de réseaux propriétaires, souvent des centres de données résident dans l'entreprise qui ont pris en charge pour le contrôle et la gestion de ces ressources Cloud. Les problèmes d'intégration, les questions de sécurité des données et applications critiques sont les principales raisons qui poussent à choisir ce modèle. Cependant, ce n'est pas toujours vrai qu'un Cloud privé est nécessairement plus sûr. La sécurisation de l'environnement de virtualisation lui-même (c'est-à-dire la sécurité au niveau de l'hyperviseur, le matériel physique, les logiciels, etc.) doit toujours être traitée, alors que dans un Cloud public le fournisseur c'est lui le responsable [52].

Dans le Cloud privé, on peut classer quatre types. Le premier c'est le *Cloud privé typique* où l'organisation héberge le Cloud dans l'un de ces propres centres de données derrière un pare-feu. Le *Cloud privé géré* permet à un fournisseur tiers la gestion et le contrôle des infrastructures qui sont propriétaire de l'entreprise. Dans le *Cloud privé hébergé,* les fournisseurs de service offrent l'infrastructure et la responsabilité de gestion nécessaire sans partager ces ressources avec d'autres organisations. En fin, le quatrième type est le *Cloud privé virtuel.* Dans ce type les fournisseurs proposent les services Cloud dans un environnement multi locataire, et le lieu d'hébergement est connu par l'entreprise cliente et se trouve souvent dans le même pays que cette dernière [53].

I.5.2 Cloud public

Le Cloud publique représente le Cloud traditionnel utilisé par la majorité des clients sur Internet. Dans ce modèle, le consommateur et le fournisseur de service sont des organisations différentes et les ressources sont auto-approvisionnées dynamiquement dans un environnement multi-locataire via des applications ou des services web. Cette catégorie de Clouds offre une facilité et une flexibilité sans investissement initial d'utilisation qui constituent la solution idéale pour les utilisateurs. En effet, le contrôle d'accès et la sécurisation des ressources est assuré entièrement par le fournisseur, ce qui limite la liberté des clients dans l'opération de contrôle et de configuration [54, 55]. Des améliorations considérables ont été réalisées dans ce modèle de déploiement, spécialement pour l'IaaS. Plusieurs compagnies investissent sur cette espace comme Amazon, avec Elastic Compute Cloud (EC2), Cloud Offerings de Rackspace et BlueCloud de IBM. D'autres formes d'offres de Cloud public sous la forme d'application ou Platform-as-a-Service, comme AppEngine de Google et la plateforme de services Azure, SimpleDB, Cloud Front, et S3 Simple Storage.

I.5.3 Cloud communautaire

Le modèle Cloud communautaire ou Cloud collectif est pour partager l'infrastructure par plusieurs organisations indépendantes ayant des intérêts communs. Cette communauté d'organisation peut partager aussi les tâches de gestion de ces infrastructures, comme la sécurisation des données, le déploiement d'applications, l'authentification, etc. [56]. L'avantage des Clouds communautaires est qu'ils permettent à plusieurs entités indépendantes d'obtenir les avantages financiers d'un Cloud non public partagé tout en évitant les problèmes de sécurité et de réglementation qui peuvent être associés à l'utilisation d'un Cloud public générique qui ne répondait pas à ces préoccupations dans son contrat SLA. Pour cela, différents types de Cloud communauté sont envisagés spécialement aux États-Unis et aux l'Union européenne sur les gouvernements aux niveaux national ou local.

On peut distinguer deux types de Cloud communauté. Le premier est le modèle fédéré où toute ressource inutile d'une organisation peut être utilisée par une autre organisation membre à la communauté. Le deuxième c'est le tiers de confiance, où un "broker" est le responsable de l'acquisition des différents services essentiels et les met à la disposition de tous les membres [57].

I.5.4 Cloud hybride

Le Cloud hybride, comme son nom l'indique, se forme lorsqu'une organisation développe un Cloud privé et souhaite exploiter des Clouds publics ou communautaires en conjonction avec son Cloud dans un but particulier. En réalité, un Cloud hybride pourrait être constitué de n'importe quelle combinaison des trois types public, privé et communautaire. À cause de ce modèle hybride, les entreprises peuvent utiliser le Cloud public pour les applications moins sensibles et le Cloud privé pour les applications et les données essentielles et sensibles. Donc, les Clouds hybrides permettent d'assembler les avantages des autres modèles et par conséquence il fournit un modèle qui offre des tolérances aux pannes et haute disponibilité [58].

Actuellement, la majorité des fournisseurs du Cloud comme HP, VMware et Amazon fournissent des services Cloud hybride. Parmi ces fournisseurs qui hébergent leurs services dans deux environnements, et si l'un de ces environnements tombe en panne, le consommateur de service peut toujours accéder à l'autre. Un autre exemple, un Cloud privé peut être utilisé pour exploiter l'infrastructure d'une entreprise, mais cette dernière peut avoir besoin de tester une mise à niveau ou de déployer un nouveau système. Il peut être avantageux de payer un Cloud public pendant quelques mois pour réaliser les tests et, lorsque leur propre Cloud privé est mis à niveau, arrêter l'utilisation de ce dernier.

Toutefois, le Cloud hybride soufre de quelques problèmes, par ce qu'il représente l'environnement le plus complexe, tels que l'exigence d'investissement initial et des coûts de maintenance pour les clients, les lois et les problématiques de sécurité et de protection des données confidentielles, etc. Afin de résoudre ces problèmes, il faut développer un ensemble de lois et opérations pour chaque environnement.

I.6 Principaux challenges de recherche dans le Cloud

Le domaine de Cloud Computing pose de nombreux problèmes, même si certaines des caractéristiques essentielles du Cloud Computing ont été réalisées par des efforts universitaires et commerciaux. Ces défis ont une influence significative sur la sécurité des données et les performances des systèmes en Cloud. De nombreux problèmes existants n'ont pas été pris en compte, et d'autres nouveaux défis continuent d'émerger [59]. D'après une enquête menée par IDC en 2008, les organisations reconnaissent les principaux problèmes qui empêchent l'adoption du Cloud Computing, comme le montre la Figure I-3. Dans cette section nous présentons les principales questions et défis rencontrés :

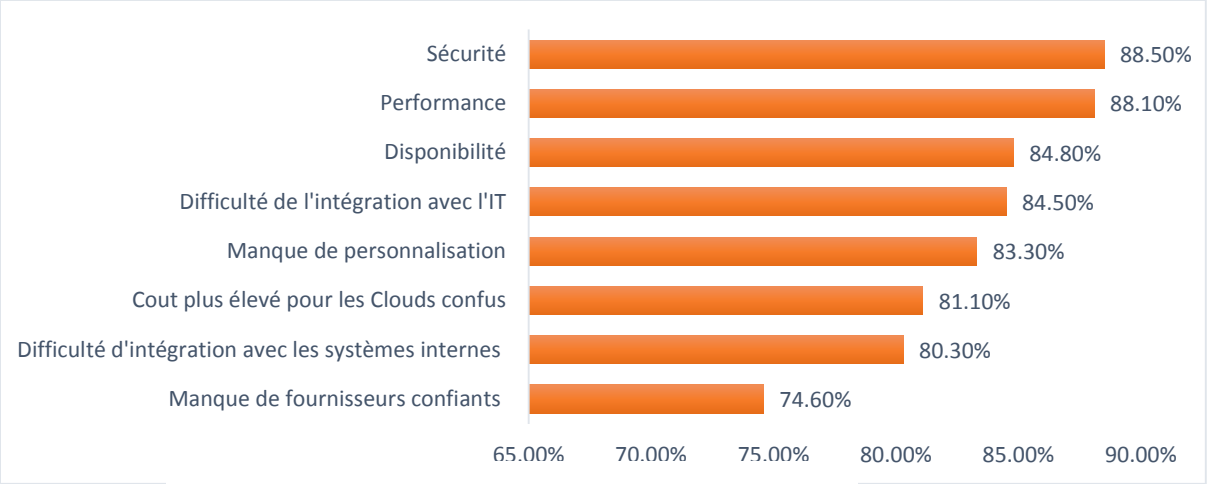


Figure I-3 : Problèmes d'adoption

I.6.1 Qualité de service

La qualité de service est l'une des principales occupations des clients [57], notamment la performance et la disponibilité. Les fournisseurs Cloud doivent assurer un certain niveau de qualité à leurs clients, et à chaque fois que la qualité est détériorée, le module de contrôle doit mettre à jour le système pour améliorer cette qualité de service. Mais cette solution n'est pas possible vu que la migration de données entre machines virtuelles et entre les Datacenter peuvent prendre plusieurs minutes. Un autre exemple de problèmes est la concurrence entre les services qui sont situés sur le même Datacenter avec une même solution Cloud, ce qui influe sur la performance de ce Cloud [58].

I.6.2 Problèmes de Migration

La migration pose deux problèmes majeurs. Le premier c'est lorsque les utilisateurs ou les fournisseurs de services de Cloud font passer leurs données à un autre fournisseur de Cloud et les données échangées n'ont pas une structure standard. Parce que la plupart des fournisseurs de services de Cloud Computing utilisent des applications Cloud propriétaires et qui ne sont pas interopérables. La deuxième, c'est la migration de machines virtuelles qui est pour équilibrer la charge de travail entre les Datacenter. Le principal avantage de la migration de machines virtuelles est d'éviter les points chauds (hot spots); toutefois, cela n'est pas simple à réaliser [60].

I.6.3 Consolidation de serveurs

La consolidation des serveurs reste toujours un sujet d'actualité au sein de l'ensemble de la communauté informatique. Elle englobe toutes les stratégies et technologies capables de réduire le nombre ou la répartition géographique des serveurs. Cette approche est pour maximiser l'utilisation des ressources, tout en minimisant la consommation d'énergie des infrastructures de Cloud Computing. Plusieurs stratégies sont utilisées pour résoudre ce problème telles que la centralisation des serveurs, la consolidation des logiciels, la consolidation physique des serveurs,

la technologie de migration de machines virtuelles, etc. Par exemple, la technologie de migration de machines virtuelles est souvent utilisée pour consolider les machines virtuelles, telles que les machines virtuelles situées dans plusieurs serveurs sous-utilisés sont consolidées sur un seul serveur, de sorte à mettre ces derniers en mode d'économie d'énergie [61].

I.6.4 Ordonnancement

Les systèmes de Cloud Computing permettent de faire l'allocation des ressources informatiques qui sont des fois insuffisantes pour satisfaire les demandes, dans ce cas, des mécanismes d'ordonnancement sont nécessaires. Le problème, donc, consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (contraintes de délai, contraintes d'enchaınement, etc.) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises. Selon la configuration de Cloud ciblée, il existe trois niveaux d'ordonnancement : niveau service, niveau tâche et niveau machine virtuelle. Le problème d'optimisation de tâches et d'allocation de ressources dans un contexte hétérogène est un problème difficile. Ce problème devient encore plus difficile lorsque les critères à prendre en considération pour l'optimisation sont multiples [62, 63].

I.6.5 Interopérabilité

Depuis l'apparition du Cloud, les entreprises consommatrices se posent la question de l'interopérabilité entre les différents services Cloud. Le problème de l'interopérabilité due à l'absence de standard qu'en peut l'utiliser pour développer des interfaces et API standardisée. L'interopérabilité est nécessaire, non seulement entre deux types différents de Cloud, mais encore entre des services de même Cloud ou des services identiques fonctionnant dans des Clouds différents. Par exemple, dans une solution de Cloud hybride, un service composant peut-être déployé dans un Cloud privé, une copie pouvant être exécutée dans un Cloud public et les deux services composants doivent fonctionner ensemble. Et avec l'absence d'outils de standardisation commune entre les fournisseurs de Cloud, ce qu'implique de sérieux problèmes de synchronisation, de portabilité et d'Interopérabilité [64].

I.6.6 Problèmes de sécurité

Le problème de sécurité est essentiellement un problème de confiance où la principale question qui se pose est celle de la transparence de transport, stockage et traitement des données dans ces environnements. Les données transférées entre les dispositifs de l'utilisateur et les Datacenter des fournisseurs de services de Cloud sont cible facile pour les pirates ou les parties non confiées. La sécurité des données et la confidentialité doivent être garanties, que ce soit sur le réseau ou encore dans les Datacenter de Cloud où elles seront stockées. Dans cette thèse, notre intérêt porte sur le volet de la sécurité des données dans toutes ces phases.

I.7 Conclusion

Dans ce chapitre, nous avons donné un aperçu détaillé sur l'approche du Cloud Computing. Nous avons présenté des définitions de cette notion et ses caractéristiques essentielles. Ensuite, nous avons montré comment le Cloud offre un large choix de services informatiques à la demande et avec facturation à l'usage aux utilisateurs selon leurs besoins. Ces services se présentent sous la formes d'un logiciel, plates-forme ou infrastructure et qui sont déployés sous quater modèles possibles qui sont: le Cloud privé, le Cloud public, le Cloud communautaire et le Cloud hybride.

Grâce aux avantages offerts par le Cloud, les utilisateurs finaux trouvent que cette technologie est un bon choix pour l'utilisation des services. Malgré, toutes ces solutions produites par le Cloud, il existe toujours des limites. Nous avons présenté les principaux défis dont la technologie Cloud Computing doit faire face pour améliorer la qualité des services fournis aux utilisateurs.

Après avoir analyser les recherches existantes, plusieurs défis en termes de sécurité sont mis en évidence. Ces insuffisances inspirent l'intérêt de nos futures recherches. Les chapitres suivants traitent le problème de la sécurité. En particulier, des questions telles que le transfert, le traitement et le stockage sécurisé des données dans le Cloud sont discutées.

Chapitre II

Sécurité dans le Cloud Computing

II.1 Introduction

Le Cloud Computing est devenu une pierre angulaire dans l'architecture de la nouvelle génération des systèmes informatique dans l'entreprise. Par contre, les mesures de contrôle et de sécurité dans le Cloud sont restées similaires à ceux utilisés dans les systèmes informatiques traditionnels. Contrairement aux solutions traditionnelles, le Cloud Computing déplace les logiciels et les données dans des grands centres de données externes qui résident dans les locaux des fournisseurs, où la gestion et le contrôle des données et des services ne peuvent pas être totalement fiables et confiants. Cependant, ces caractéristiques soulèvent de nombreux nouveaux problèmes de sécurité.

Une étude réalisée par l'International Data Corporation IDC en 2009 montre que 74% des responsables informatiques et des hommes d'affaires estimaient que les problèmes de sécurité liés au Cloud Computing constituaient le principal défi qu'ils empêchaient à utiliser les services de Cloud Computing [65]. D'autres études montrent que le principal problème des entreprises, lors de l'adoption de services Cloud Computing publics, est la sécurité et la confidentialité [66]. John Chamber président et PDG de CISCO avait dit que « le Cloud Computing est un cauchemar de sécurité et ne peut pas être traité de manière traditionnelle » [67]. Les principaux fournisseurs de Cloud Computing tels que Amazon [68], Google Docs [69], Google Gmail [70] et VMware [71] ont successivement signalé plusieurs accidents. Des incidents de sécurité graves ont même conduit à l'effondrement des fournisseurs de Cloud Computing ; à titre d'exemple LinkUp [72].

Toutefois, les questions de confidentialité, de sécurité, de fiabilité et d'interopérabilité doivent encore être résolues de manière adéquate ; en particulier, les problèmes de sécurité et de confidentialité des données qui sont très importants et hautement prioritaires. Par conséquent, le milieu de la recherche doit prendre en compte ces préoccupations en proposant et en mettant en place des mécanismes de protection solides permettant d'obtenir les avantages du Cloud Computing sans risquer la sécurité et la confidentialité. En effet, certaines premières tentatives, déjà proposées, visent à fournir une couche sécurisée pour traiter les données, mais d'autres solutions possibles doivent être explorées pour renforcer la protection afin de créer un environnement de Cloud Computing solide.

Dans ce chapitre, nous donnons une vue précise du la sécurité du Cloud Computing. Nous commençons par la présentation de la problématique de la sécurité dans le Cloud, les exigences de sécurité pour l'architecture Cloud et les modèles de sécurité. Nous soulignons ensuite certains problèmes potentiels de sécurité et quelques travaux liés à cette technologie.

II.2 Exigences de sécurité du Cloud Computing

Dans la norme ISO 7498-2 [10], produite par l'Organisation internationale de normalisation (ISO), la sécurité de l'information devrait couvrir un certain nombre de thèmes suggérés. La sécurité du Cloud Computing doit également être guidée à cet égard afin de devenir une solution technologique efficace et sécurisée. Les exigences de sécurité de la norme ISO 7498-2 seront mise en évidence ci-dessous dans le contexte du Cloud Computing.

La sécurité est généralement liée aux trois aspects principaux et d'autres secondaires. Les principaux aspectes sont la confidentialité, l'intégrité et la disponibilité (***CID***).

II.2.1 Confidentialité

Dans le Cloud Computing, la confidentialité joue un rôle majeur, notamment dans le contrôle des données des organisations situées dans plusieurs bases de données distribuées. La confidentialité consiste à garantir que les données des utilisateurs qui résident dans le Cloud ne sont pas accessibles à des tiers non autorisés. Elle est nécessaire et spécialement lorsque on utilise un Cloud public en raison de la nature d'accessibilité de ce type [73].

Le Cloud Computing implique le stockage de données des utilisateurs sur des serveurs distants appartenant à des tiers ou exploités par des tiers. L'intégralité du contenu du périphérique de stockage d'un utilisateur peut être stockée chez un seul fournisseur de Cloud ou chez de nombreux fournisseurs. Pour cela, la menace de données est augmentée, en raison du nombre croissant de parties, de périphériques et d'applications utilisés, ce qui conduit à une augmentation du nombre de points d'accès. La gestion du contrôle de données dans cet environnement conduit à une augmentation de risques de données traitées.

La multi-location présente un certain nombre de menaces pour la vie privée et la confidentialité. La réutilisabilité des infrastructures est une caractéristique importante des environnements en Cloud, mais les infrastructures réutilisables doivent être soigneusement contrôlées pour que ne créer pas une vulnérabilité grave. La confidentialité des données pourrait aussi être violée involontairement en raison de la rémanence des données. La rémanence de données est la représentation résiduelle de données qui ont été effacées ou supprimées nominalement. D'un autre côté, il peut également arriver par une malveillance qu'un utilisateur demande une large quantité d'espace disque, puis il récupère des données sensibles [74].

L'assertion de la confidentialité des profils des utilisateurs et la protection de leurs données est effectuée avec des techniques d'accès virtuelle qui permettent d'appliquer les protocoles de sécurité au niveau de différentes couches du Cloud. La confidentialité peut être obtenue grâce à

des techniques de chiffrement appropriées prenant en compte le type de chiffrement. En fait, tout est basé sur la politique de sécurité du fournisseur de Cloud et dépend également au choix des clients qu'ils peuvent chiffrer leurs données avant de les télécharger [7].

La confidentialité des données dans le Cloud est corrélée à l'authentification de l'utilisateur. La protection du compte d'un utilisateur contre le vol est un problème très vaste qui essaye de contrôler l'accès aux objets, à la mémoire, aux périphériques, aux logiciels, etc. L'authentification consiste à établir la confiance dans l'identification des utilisateurs lorsqu'ils sont présentés à un système. Le manque d'authentification solide peut conduire à un accès non autorisé au compte des utilisateurs sur un Cloud, et par conséquence une brèche de la confidentialité [74].

II.2.2 Intégrité

L'exigence d'intégrité réside dans l'application de la diligence requise dans le domaine du Cloud, principalement lors de l'accès aux données. Par conséquent, les propriétés ACID (atomicité, cohérence, isolation et durabilité) des données du Cloud doivent être imposées de manière robuste dans tous les modèles de Cloud Computing.

L'intégrité est un aspect clé de la sécurité de l'information. Elle signifie que les objets ne peuvent être modifiés que par les parties autorisées ou par les moyens autorisés. Elle peut être associée aux données, aux logiciels ou au matériel. L'intégrité des données consiste à protéger les données contre toute suppression, modification ou opération non autorisée. La gestion des droits d'accès des ressources spécifiques garantit que les données et services ne sont ni maltraités ni détournés. De plus, les mécanismes d'assurance de l'intégrité offrent une meilleure visibilité pour déterminer qui peut modifier les données du système (responsabilité) [74].

Dans le Cloud Computing, l'intégrité fait référence à la capacité du fournisseur de Cloud d'assurer un fonctionnement fiable et correct du système de Cloud afin de respecter ses obligations légales, telles que les contrats SLAs, ainsi que les normes techniques auxquelles les propriétés ACID (atomicity, consistency, isolation and durability) des données stockées dans le Cloud devraient, être imposées de manière robuste. Les données peuvent être cryptées pour assurer la confidentialité. Cependant, rien ne garantit que les données n'aient pas été modifiées pendant qu'elles résident dans le Cloud. On fait, seulement la confiance au fournisseur de Cloud Computing pour maintenir l'intégrité et l'exactitude des données. Les fournisseurs ont, donc, une obligation vis-à-vis de leurs clients et les obligations légales en cas de violation ou d'incident. Dans ce cas, le client doit disposer les informations suffisantes et la visibilité sur le système pour pouvoir fournir des rapports au règlement.

Les fournisseurs Cloud doivent garantir la tolérance aux pannes et la réparation des incidents possible. Pour un fournisseur de Cloud, l'une des événements les plus dévastatrices peut être la panne de service due à une défaillance du système de Cloud. Les fournisseurs de Cloud doivent assurer que les zones de stockage des services soient isolées afin de prévenir les pannes et mettre en place des mécanismes de reprise. La reprise après sinistre est un autre problème très complexe.

Même si l'utilisateur ne sait pas où se trouvent ses données, le fournisseur de Cloud devrait lui dire ce qu'il se passe en cas de sinistre. Il est dangereux que toute offre ne répliquant pas les données et les applications sur plusieurs sites, c'est une vulnérable à un échec total. Il existe plusieurs stratégies de réplication des données peuvent être établies pour résoudre ce problème [75, 76].

II.2.3 Disponibilité

La disponibilité est l'une des exigences les plus critiques en matière de sécurité des informations dans le Cloud Computing, car c'est un facteur essentiel dans la décision pour choisir entre les fournisseurs de Cloud privés, publics ou hybrides, ainsi que dans les modèles de déploiement. En termes simples, la disponibilité signifie qu'une organisation dispose de l'ensemble de ses ressources informatiques accessibles et utilisables à tout moment. La disponibilité peut être affectée de manière temporaire ou permanente, et une perte peut être partielle ou complète. Les SLAs sont le document le plus important qui souligne l'inquiétude de la disponibilité des services et des ressources entre le fournisseur et le client du Cloud. L'objectif de la disponibilité pour les systèmes Cloud (y compris les applications et les infrastructures) est de garantir que les utilisateurs peuvent les utiliser à tout moment et en tout endroit. C'est l'un des premières préoccupations critiques des organisations pour le fonctionnement et la sécurité.

La disponibilité ne concerne pas seulement les données et les logiciels, mais également le matériel disponible sur demande pour les utilisateurs autorisés. Elle peut s'étendre également à la nécessité de migrer vers un autre fournisseur. La disponibilité du système inclut la capacité du système à poursuivre ses activités même lorsque certaines autorités se comportent mal. Le système doit pouvoir continuer en opération même en cas de violation de la sécurité. Le fournisseur de services Cloud doit garantir que les informations et le traitement de l'information sont disponibles pour les clients à la demande, ce qui peut sous-estimer la forte dépendance de la disponibilité du réseau omniprésent [73].

Les menaces les plus connus devant la disponibilité sont les pannes d'équipement, les catastrophes naturelles et les attaques par déni de service (DDoS Distributed Denial of Service). Il est très difficile de détecter les menaces qui peuvent être des attaques basées sur le réseau, telles que les attaques par déni de service. La co-localisation des données avec celles d'une organisation et un profil de menace, pourrait également entraîner un déni de service, en tant que victime accidentelle d'une attaque dirigée contre cette organisation. De même, une attaque contre les ressources physiques d'un fournisseur est également possible. La possibilité d'une menace interne est considérablement élargie lors de la sous-traitance de données et de processus est dans les Clouds.

La disponibilité des applications SaaS garantit aux entreprises un service pendant toute la journée. Cela implique des modifications architecturales au niveau de l'application et de l'infrastructure pour ajouter de l'évolutivité et une haute disponibilité. Une architecture de multi-niveaux doit être adoptée, supportée par une exploitation équilibrée des instances d'application qui

s'exécutent sur un nombre variable de serveurs. La résilience face aux pannes matérielles ou logicielles, ainsi qu'aux attaques par déni de service, doit être développée à partir de la base de l'application [77, 78].

Dans la littérature et les recherches dans la sécurité de Cloud, en parallèle avec les exigences de base ***CID***, plusieurs exigences sont détaillées et présentées comme des factures importantes pour assurer le bon fonctionnement du système de sécurité. Dans la suite nous détaillions ceux qui ont plus d'influence sur la sécurité de Cloud.

II.2.4 Identification et authentification

Dans le Cloud Computing, en fonction du type de Cloud ainsi que le modèle fourni, les utilisateurs doivent tout d'abord établir les informations d'authentification en précisant les priorités d'accès. Ce processus vise à vérifier et à valider des utilisateurs du Cloud en utilisant des noms d'utilisateur et des mots de passe pour protéger leurs profils. L'authentification est en générale comporte l'authentification d'origine et l'authentification d'entité ou l'identification. L'authentification d'origine est un service de sécurité pour vérifier l'identité du système prétendu qu'il est la source originale des données reçues. L'authentification d'entité est le processus de vérification d'une déclaration dans laquelle un système entité ou une ressource du système a une certaine valeur d'attribut. Cet attribut est généralement l'identité de l'utilisateur [79].

II.2.5 Autorisation

L'autorisation est une exigence importante en matière de sécurité dans le Cloud ; c'est elle ait garanti le maintien de l'intégrité référentielle. L'autorisation est le fait d'accorder des droits ou des privilèges à une personne, un utilisateur ou un processus. Elle peut prendre plusieurs formes comme les listes de contrôles d'accès (ACL) qui sont de simples listes d'utilisateurs accompagnées de leurs droits comme écriture, lecture, modification, suppression ou exécution sur des ressources ou de classe de ressources spécifiques. L'autorisation est gérée par l'administrateur de système dans un Cloud privé.

II.2.6 Confiance

La confiance est un concept utilisé dans nombreuses disciplines. En informatique, on se concentre sur la conception d'outils pour l'assistance des utilisateurs dans diverses tâches. Souvent, ces outils fonctionnent sur un modèle donné de confiance et fournissent des méthodes pour mesurer la confiance dans un contexte d'application spécifique. La complexité du Cloud Computing rend la problématique de sécurité d'une importance primordiale pour les consommateurs potentiels et les fournisseurs de services. Ces différents points soulèvent la problématique de confiance dans l'utilisation des services de Cloud Computing. Pour cela, la confiance a été utilisée pour convaincre les observateurs qu'un système (modèle, conception ou implémentation) est correct et sécurisé [80].

Dans les environnements Cloud, le client dépend au fournisseur avec divers services. Parmi ces services qui obligent le client de stocker ses données confidentielles dans le côté du fournisseur. Ainsi, un cadre de confiance qui dépend en grande partie du modèle de déploiement sélectionné et la gouvernance des données et des applications devrait être établie pour gérer des exigences évolutives en matière de confiance et d'interaction / partage [81, 82].

Dans les architectures traditionnelles, la confiance était assurée par une politique de sécurité efficace, qui adresse des contraintes sur les fonctions et leur flux, des contraintes sur l'accès des systèmes externes et des adversaires, y compris des programmes ainsi que l'accès aux données des personnes. Dans un déploiement dans le Cloud, cette sensibilité est totalement occultée dans le cas des Clouds publics ou communautaires où le contrôle est délégué par le fournisseur de Cloud.

II.2.7 Audit et conformité

La conformité et l'audit comprend différentes activités qui contiennent la génération, la collecte et l'analyse du réseau, d'un système et d'une application de manière à maintenir une vision réelle sur la sécurité. Cela aidera également les auditeurs à vérifier le respect des différentes politiques de contrôle d'accès, les audits et les rapports périodiques. La surveillance de la sécurité se fonde sur une estimation automatisée de ces données d'audit. Elle se concentre sur les contrôles et les procédures de sécurité, les procédures de sauvegarde, les plans de secours, la sécurité des ressources et d'autres domaines.

Pour les systèmes Cloud, le client doit pouvoir autoriser et éventuellement surveiller l'accès au système. Une telle surveillance doit être aussi simple pour suivre les journaux sur une interface en ligne et sophistiquée pour regarder le journal d'audit en temps réel des actions de l'administrateur sur le système - que ce soit sur une machine virtuelle spécifique ou sur l'hyperviseur du système entier. L'audit consiste aussi à inspecter et examiner les enregistrements d'autorisation et d'authentification afin de vérifier si la conformité aux normes et aux stratégies de sécurité prédéfinies est assurée. Des règles sont définies dans l'inspection des logs permettant l'extraction efficace d'événements liés à la sécurité. Le logiciel d'inspection de logs sur les ressources Cloud permet de détecter tout comportement suspect.

II.2.8 Non-répudiation

La non-répudiation est le fait de s'assurer qu'un contrat, notamment un contrat signé via internet, ne peut être remis en cause par l'une des parties. La non-répudiation dans le Cloud Computing est une propriété du stockage de données, exigeant que lorsqu'un propriétaire de données (consommateur) envoie une demande à un fournisseur de Cloud pour le téléchargement de données, la transaction de téléchargement de données doit être effectuée de manière que ni le propriétaire des données ni le fournisseur de stockage ne peuvent nier de cette transaction. Autrement dit, la non-répudiation de l'origine prouve que les données ont été envoyées, et la non-répudiation de l'arrivée prouve qu'elles ont été reçues [83].

Généralement, il est admis par la communauté que la non-répudiation peut être obtenue en appliquant les protocoles de sécurité traditionnels du commerce électronique telles que les signatures numériques. En effet cette technologie permet de prouver l'identité d'une personne par la possession de sa propre clé privée. La protection de cette clé devient alors une préoccupation pour l'utilisateur. Celui-ci peut utiliser des authentificateurs tel que les cartes à puce.

Pour satisfaire aux exigences de sécurité et résoudre les problèmes de sécurité analysés ci-dessus, nous pouvons résumer les issues de sécurité par l'architecture illustré dans la Figure II-1.

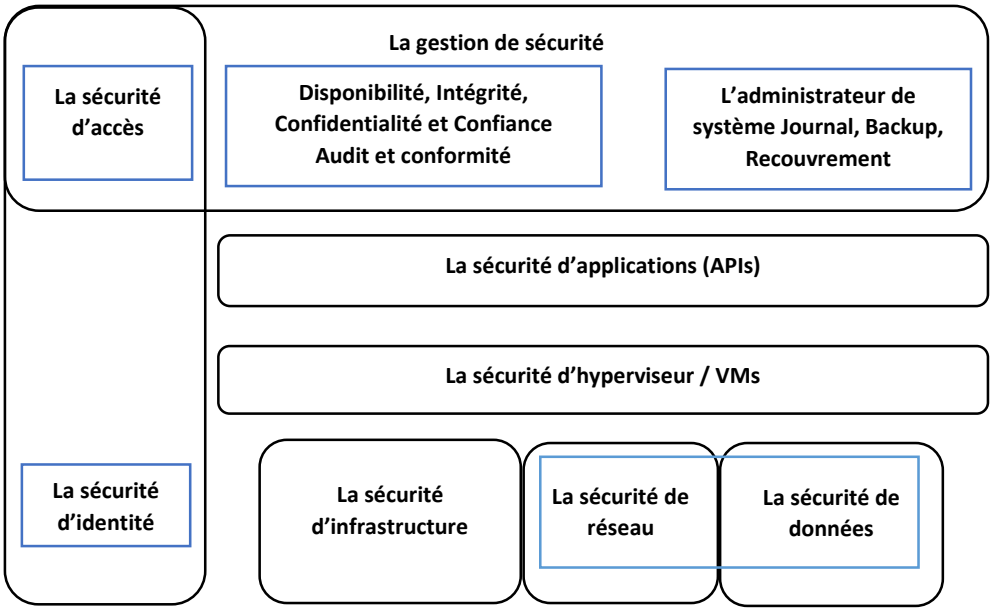


Figure II-1 : Exigences de sécurité

De nombreux préoccupations de sécurité sont associés au Cloud Computing. Cloud Security Alliance (CSA) a identifié quatorze domaines de préoccupation dans sa troisième édition [84]. Selon Gartner [85], les utilisateurs devraient demander aux fournisseurs sept issues de sécurité spécifiques: accès privilégié des utilisateurs, conformité réglementaire, localisation des données, séparation des données, possibilité de récupération, support de l'investigation et viabilité à long terme. Les chercheurs dans [86, 71] ont examiné les issues de sécurité liés au Cloud Computing sous différents angles, notamment les issues de sécurité liés à l'architecture, aux modèles de prestation de services, aux caractéristiques du Cloud et aux utilisateurs du Cloud.

II.3 Gestion de la sécurité du Cloud

Lors de la migration vers le Cloud, les entreprises doivent bien évaluer et comprendre les risques de sécurité potentiels associés au Cloud et définir des attentes réalistes avec les fournisseurs. Les questions critiques de sécurité pour la mise en œuvre des technologies de Cloud Computing devront ensuite être étudiées. La question suivante décrit certains problèmes de gestion et de contrôle du Cloud qui doivent être traités [87]:

- Comment, pour une entreprise, gérer les risques actuels et aussi les nouveaux risques de conformité du Cloud? Cela sera traité en utilisant l'impact potentiel de Cloud Computing qu'il peut avoir sur l'entreprise en matière de gouvernance et législation.

- Comment le Cloud Computing peut affecter la vie stratégique de l'entreprise en influençant potentiellement sa différenciation du marché?

Lors de la mise en place d'une cadre de Cloud qui adresse spécifiquement la sécurité de l'information des entreprises, les cadres supérieurs et la direction, il faut chercher à adapter et à intégrer les politiques actuelles de protection des données, de confiance et de confidentialité dans la formulation de l'ensemble compréhensive de directives du Cloud Computing. Ces directives peuvent inclure:

- La mise en place d'une politique globale des affaires en Cloud qui souligne la position des entreprises en matière de protection des informations.

- Gouverner l'installation et la communication de l'informatique en Cloud lorsque les décisions concernant la technologie de l'information (IT) sont réalisées.

- L'exploitation des processus actuels d'audit de la technologie de l'information et aussi bien les processus TAX avec l'intégration des pratiques de divulgation de la sécurité du Cloud et d'audit Cloud.

Les directives de Cloud Computing sont considérées comme la pierre angulaire de la stratégie Cloud. Ils comportent la gouvernance du Cloud et leurs parties responsables. Dans la suite, nous analysons ces implications dans la gestion de la sécurité Cloud.

II.3.1 Gouvernance dans le Cloud

La gouvernance signifie l'emplacement des stratégies relatives à l'utilisation des ressources Cloud, telles que le calcul (computing) et le stockage, ainsi que les services Cloud. Les développeurs doivent comprendre les mécanismes qui font un bon programme de gouvernance du Cloud et ils prennent des mesures pour trouver la bonne solution pour leur projet et l'intégrer dans leurs processus de développement ainsi que dans leurs opérations. Au cours de la phase de développement, il est important de mettre des politiques autour de la surveillance et la gestion de la qualité de Service (SLA). En plus, la définition des politiques d'accès dans l'entreprise est un élément clé de la définition des politiques de gouvernance.

La plupart des entreprises ont des politiques et procédures de sécurité pour protéger leur propriété intellectuelle et leurs activités. En plus, les entreprises devraient établir un cadre de gouvernance de manière formel qui définit les chaînes d'autorité, de responsabilité et de communication. Ceci décrit les responsabilités des personnes impliquées, comment elles interagissent et communiquent, ainsi que les politiques et règles générales [71].

II.3.2 Transparence de la sécurité

Une sécurité transparente impliquerait que les fournisseurs de services Cloud divulguent des informations adéquates sur leurs politiques, leurs conceptions et leurs pratiques en matière de sécurité, y compris la déclaration des mesures de sécurité pertinentes dans les opérations quotidiennes [85]. Le SLA est l'un des protocoles les plus importants pour assurer la transparence dans le Cloud Computing qui est utilisé d'après les fournisseurs de services Cloud [88].

Les Clouds publics sont plus susceptibles d'être perçus comme le Cloud qui a un degré de transparence plus élevé par rapport aux modèles de Cloud hybride ou privé. Cela est dû au fait que les fournisseurs de Cloud public ont une offre de Cloud « standardisée », ciblant ainsi une base de clients plus large. Par contre, les Clouds privés sont généralement conçus pour des entreprises spécifiques qui ont plus d'attention à la personnalisation aux fonctionnalités.

II.3.3 Impact de la sécurité sur le Cloud Computing

Le principal impacte de la sécurité est l'impact économiques. Le partage et la mutualisation du matériel ainsi que la responsabilité de sécurité affectée aux fournisseurs permet d'optimiser les coûts, et de réaliser des applications partagées sans avoir besoin de louer chez un fournisseur ses propres serveurs et sans s'inquiéter de la situation de sécurité. Cela évite aussi toute maintenance des ressources. Ces avantages poussent à l'utilisation du Cloud, mais le système de paiement aux ressources consommées peut produire une surfacturation de la part du fournisseur pour des ressources allouées mais non consommées. Cette surfacturation est une perte pour le client [89]. En adition, si un fournisseur tombe en faillite, peut entraîner la faillite du client lorsque les données qui étaient hébergées chez le fournisseur soient sensibles et indispensables.

Comme les fabricants d'ordinateurs, les employeurs et les universités déploient des Clouds basés sur les outils de bureau, de nombreux utilisateurs risquent de ne pas se rendre compte à les utiliser. Ce risque de confusion augmentera lorsque les applications en Cloud sont exploitées par des navigateurs non connus qui ont parfois continu à fonctionner même lorsque l'utilisateur n'est pas en connexion. Il est possible pour les sites Web malveillants d'exploiter les vulnérabilités des navigateurs afin de dérober des informations associées à d'autres sessions de navigation, telles qu'un compte de messagerie ou une session de banque en ligne. L'utilisation des protocole HTTPS cryptée, SW Sécurisée et les normes de sécurité XML [90] pour réduire ces risques, nécessite beaucoup plus de puissance de traitement et de mémoire pour un serveur Web que pour une connexion Web normale [91].

Les attaques ont un impact économique pour le client. Ces attaques peuvent être contrées grâce à la scalabilité du Cloud et donc engendrer des coûts supplémentaires pour le client. Pour cela, les entreprises doivent toujours décider si les mesures de sécurité appropriées sont mises en place ou si elles partagent une responsabilité commune avec les fournisseurs lorsqu'elles s'engagent dans l'environnement Cloud [92]. Dans le cas où l'entreprise n'a pas mis en place ces types de solutions, il risque que son infrastructure soit hors service et donc des pertes de bénéfices pour celui-ci.

II.3.4 Implications de sécurité

Les entreprises doivent bien comprendre les risques de sécurité du Cloud et analyser les implications en matière de sécurité de l'information et de confidentialité du Cloud Computing pour garantir une bonne gestion de sécurité. Les principales implications de la gestion de la sécurité sont comment gérer les personnes, les rôles et les identités et d'assurer la confidentialité ? Les entreprises doivent s'assurer que le fournisseur dispose de processus qui déterminent qui a l'accès et le fournisseur doit permettre au client d'attribuer et de gérer les rôles, les identifications et les autorisations pour chacun de ses utilisateurs, l'essentiel est que les exigences de confidentialité soient traitées.

L'implication d'une bonne protection des données est à cause que les données sont au cœur de toutes les préoccupations de sécurité informatique. Le Cloud ne change pas cette préoccupation mais apporte de nouveaux implications en raison de la nature du Cloud Computing tels que l'opération d'audit et le processus métier, l'évaluation des différentes considérations de stratégie de sécurité, l'évaluation des contrôles de sécurité et l'infrastructure physique et les réseaux et les connexions, etc [93].

II.4 Menaces de la sécurité dans le calcul

La principale préoccupation dans les environnements de Cloud Computing est de fournir une sécurité autour de la multi-location et de l'isolation qui offre ainsi aux clients plus de confort et plus de confiance dans le Cloud [94]. Plusieurs études ont été rapportées pour classer les menaces à la sécurité dans le Cloud en fonction de la nature des modèles de déploiement de services d'un système de Cloud Computing [95].

Le modèle de déploiement de services est l'un des nombreux aspects à prendre en compte dans le cadre d'une enquête exhaustive sur la sécurité du Cloud. La sécurité à différents niveaux, tels que le niveau réseau, le niveau hôte et le niveau données ou application, est nécessaire pour que le Cloud reste opérationnel et fonctionne en permanence. En fonction de ces différents niveaux, différents types de violations de sécurité peuvent survenir. Celles-ci ont été classées dans le reste de cette section.

II.4.1 Sécurité de base

Le Web 2.0, une technologie clé permettant l'utilisation de logiciel en tant que service (ex : SaaS). Cette technologie soulage les utilisateurs dans les tâches de la maintenance et l'installation de logiciels. Il a été largement utilisé dans tous les côtés. Alors que la communauté d'utilisateurs utilisant le Web 2.0 s'augmente de plus en plus, la sécurité est devenue plus importante que jamais pour un tel environnement [96, 97, 98].

II.4.1.1 Attaque par injection SQL

Les attaques par injection SQL sont celles dans lesquelles un code malveillant est inséré dans un code SQL standard afin que les attaquants obtiennent un accès non autorisé à une base de données et puissent accéder à des informations sensibles. Parfois, les données entrées par un pirate sont mal comprises par le site Web, car elle le considère comme l'utilisateur original, et le permet d'accéder au serveur SQL, ce qui permet au pirate d'avoir une connaissance du fonctionnement du site Web et d'y apporter des modifications. Diverses techniques pour faire face au ce type d'attaque telles que: éviter l'utilisation de la génération dynamiquement du code SQL dans le code, utiliser des techniques de filtrage pour assainir l'entrée d'utilisateur, etc.

```
Requête SQL :          select title, text from news
                       where id=$id
Requête  SQL injecté : select title, text from news
                       where id=10 or 1=1
```

Figure II-2 : Exemple d'un attaque par injection SQL [99]

II.4.1.2 Attaque XSS

L'attaque XSS (Cross Site Scripting) est un type de faille de sécurité des sites web qui injecte des scripts malveillants dans le contenu Web. Elle est devenue très populaires depuis la création du Web 2.0. Les possibilités des XSS sont très larges puisque l'attaquant peut utiliser tous les langages pris en charge par le navigateur (JavaScript, PHP, Flash...) et de nouvelles possibilités sont régulièrement découvertes notamment avec l'arrivée de nouvelles technologies comme HTML5. Selon le type de services fournis, un site Web peut être classé comme statique ou dynamique. Les sites Web statiques ne souffrent pas des menaces de sécurité comme les sites Web dynamiques en raison de leur dynamisme dans la fourniture de services multiples aux utilisateurs. Il est possible de rediriger un utilisateur vers un autre site pour l'hameçonnage ou encore de voler la session en récupérant les cookies. Dans l'exemple suivant un scripte d'une attaque XSS pour afficher le dernier commentaire d'une base de données [99] :

```
print "<html>"
print "<h1>Most recent comment</h1>"
print database.latestComment
print "</html>"
```

Il a souvent été observé qu'au cours de travaux sur Internet, des pages Web ou des fenêtres contextuelles s'ouvrent à la demande d'un clic pour afficher le contenu qu'elles contiennent. Le plus souvent, inconsciemment (sur les dangers possibles) ou par curiosité, les utilisateurs cliquent sur ces liens dangereux et ainsi, le tiers intrus obtient le contrôle de ses informations personnelles et il pirate ses comptes. Diverses techniques telles que le filtrage du contenu actif, la prévention des fuites de données basée sur le contenu, la technologie de détection de la vulnérabilité des applications Web, etc. ont déjà été proposées pour traiter ces types d'attaques [100].

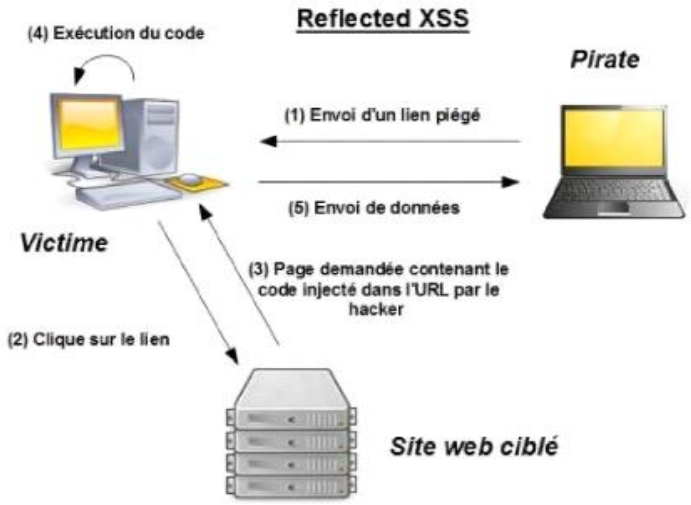


Figure II-3 : L'attaque XSS [101]

II.4.1.3 Attaque de l'intercepteur

Une autre classe d'attaques très connue en SaaS est appelée l'attaque au milieu (Man in the Middle attacks MITM), parfois appelée attaque de l'intercepteur. Dans une telle attaque, l'attaquant a pour but d'intercepter les communications entre les parties, sans qu'elles puissent se douter que le canal de communication entre elles a été compromis. L'attaque de l'intercepteur est particulièrement applicable dans les échanges sans authentification. Plusieurs outils mettant en œuvre des technologies de cryptage puissantes telles que: Dsniff, Cain, Ettercap, Wsniff, Airjack, etc. ont été développés pour assurer la protection. Une étude détaillée sur la prévention des attaques de l'homme au milieu a été présentée dans [102].

Les chercheurs dans [103] ont récemment mentionné, dans un article publié sur Security.com pour s'occuper aux failles de sécurité traditionnelles, l'évaluation des logiciels en tant que sécurité du service, la séparation des processus de sécurité des serveurs et les terminaux et l'évaluation de la virtualisation au niveau du serveur. Par conséquent, la sécurité à différents niveaux est nécessaire pour garantir la bonne mise en œuvre du Cloud Computing, telles que: la sécurité d'accès de serveur, la sécurité d'accès à Internet, la sécurité d'accès aux bases de données, la sécurité des données et la sécurité des programmes. En outre, il doit assurer la sécurité des données au niveau de la couche réseau et la sécurité des données au niveau de la couche physique et de la couche application pour maintenir un Cloud sécurisé.

II.4.2 Sécurité au niveau du réseau

Les réseaux sont classés en plusieurs types tels que: réseaux partagés et non partagés, publics ou privés, réseaux de petite ou de grande surface et chacun d'entre eux doit faire face à un certain nombre de menaces pour la sécurité. Pour assurer la sécurité du réseau, il convient de prendre en compte les points suivants: la confidentialité et l'intégrité dans le réseau, le contrôle d'accès et le maintien de la sécurité appropriés contre les menaces de tiers, tout en assurant une sécurité au niveau du réseau. Les attaques au niveau du réseau comprennent les suivantes: l'attaque de DNS,

attaques par Sniffer, problème ou la réutilisation d'adresse IP, l'attaque par déni de service (DoS) et l'attaque par déni de service distribué (DDoS), etc.

II.4.2.1 Attaque de DNS

Un serveur de nom de domaine (DNS) met en œuvre les mécanismes de transfert d'un nom de domaine vers une adresse IP. Les serveurs DNS sont nécessaires parce que les noms de domaine sont beaucoup et n'est pas faciles à les retenir. L'objectif de ce type d'attaque est de rediriger des Internautes vers des sites pirates en utilisant des faiblesses du protocole DNS et/ou de son implémentation à travers des serveurs de nom de domaine. Il existe deux principales attaques de type DNS: le DNS ID Spoofing et DNS Cache Poisoning.

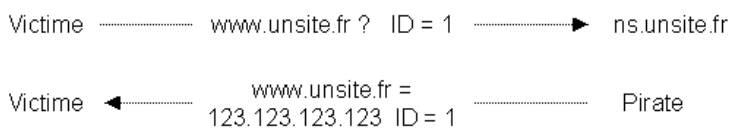


Figure II-4 : L'attaque de DNS ID Spoofing [104]

Dans le DNS ID Spoofing, l'attaque va consister à récupérer le numéro d'identification d'après une requête DNS (en sniffant, lorsque l'attaque est effectuée sur le même réseau physique, ou en utilisant une faille des systèmes d'exploitation ou des serveurs de type DNS) puis envoyer des réponses malveillantes à la victime avant le serveur DNS [104]. Ainsi, la victime utilisera, sans le savoir, l'adresse IP du pirate. Le schéma ci-dessus illustre simplement le principe du DNS ID Spoofing. Dans le DNS Cache Poisoning, les serveurs DNS ont un cache permettant de sauvegarder pendant un certain période la correspondance entre un nom de domaine et son adresse IP.

L'utilisation de mesures de sécurité DNS telles que: les extensions de sécurité du système de noms de domaine (DNSSEC) réduit les effets des menaces DNS, mais il peut arriver que ces mesures de sécurité s'avèrent inadéquates lorsque le chemin entre l'expéditeur et le destinataire est redirigé via une connexion malveillante. Il peut arriver que même après que toutes les mesures de sécurité DNS aient été prises, la route sélectionnée entre l'expéditeur et le destinataire provoque des problèmes de sécurité [105].

II.4.2.2 Balayage de port

Le balayage de ports (Port scanning) est une technique servant à rechercher les ports ouverts sur un serveur de réseau. Généralement cette technique est utilisée par les administrateurs des systèmes informatiques pour contrôler la sécurité des serveurs de leurs réseaux. Les pirates informatiques utilisent cette technique pour tenter de trouver des failles dans des systèmes informatiques. Ce type d'attaque permet à celui-ci de découvrir des ports de communication exploitables. Les ressources du Cloud sont sensibles à l'attaque par balayage de port si celle-ci est effectuée en parallèle. Cette attaque peut être évitée en utilisant des systèmes de sécurité comme un système de détection d'intrusion IDS ou encore un pare-feu [106, 107].

II.4.2.3 Attaque SNIFFER

L'attaque par Sniffing ou reniflement de trafics est une forme d'attaque sur le réseau qui est couramment utilisée par les pirates pour espionner le trafic sur le réseau. Ces types d'attaques sont lancés par des applications capables de capturer des paquets circulant sur un réseau. Si les données transférées via ces paquets ne sont pas cryptées, elles peuvent être lues. Il est donc possible que des informations vitales circulant sur le réseau puissent être tracées ou capturés. Dans la pratique, les pirates ont généralement recours à cette méthode pour détecter tous les messages circulant sur le réseau en retrouvant des mots de passe et des données sensibles. Il existe des plateformes de détection de sniffing malveillante basées sur ARP (Address Resolution Protocol) et RTT (round trip time) qui peuvent être utilisées pour détecter un système de sniffing s'exécutant sur un réseau [108].

II.4.2.4 Problème d'adresses IP réutilisées

Une adresse IP est fournie à chaque nœud d'un réseau. La liste des adresses IP a principalement un nombre déterminé. Dernièrement, plusieurs problèmes liés à une adresse IP réutilisée ont été observés. Lorsqu'un utilisateur particulier quitte un réseau, son adresse IP qui lui est associée (auparavant) est affectée à un nouvel utilisateur. Cela risque parfois de compromettre la sécurité du nouvel utilisateur, car il s'écoule un certain délai entre le changement d'une adresse IP dans le DNS et le nettoyage de cette adresse dans les caches DNS, dans ce temps l'ancienne adresse IP serait attribuée au nouvel utilisateur, et par conséquent, la possibilité d'accès aux données par un autre utilisateur ne sont pas négligeables car l'adresse existe toujours dans le cache DNS. Les données de l'utilisateur peuvent devenir accessibles à un autre utilisateur violant la vie privée de l'utilisateur initial [109].

II.4.2.5 Détournement de BGP

Le protocole BGP (Border Gateway Protocol) est utilisé pour diriger le trafic sur Internet. Les réseaux utilisent le protocole BGP pour échanger des informations d'accessibilité. Tout réseau connecté à Internet dépend éventuellement d'un BGP pour atteindre à d'autres réseaux. Le détournement ou le hijacking de BGP est donc un type d'attaque réseau. Il y a aujourd'hui deux types d'attaques BGP. La première se traduit par l'annonce erronée concernant les adresses IP associées à un système autonome (AS), permettant ainsi à des tiers malveillants d'avoir accès aux adresses IP qui ne leur appartiennent pas. Le second type d'attaques se produit lorsqu'un ASN injecte une route qui recouvre une annonce existante. Dans ce cas, la route " plus spécifique " prend la priorité, et le nouvel ASN peut recouvrir une portion de l'espace d'adressage d'un tiers. Quand les spammers le font, ils annoncent la route pour quelques minutes ou quelques heures seulement. Un système de sécurité autonome pour les systèmes autonomes a été expliqué dans [109].

II.4.3 Sécurité au niveau de l'application

La sécurité au niveau de l'application fait référence à l'utilisation des ressources logicielles et matérielles pour assurer la sécurité des applications de telle sorte que les attaquants ne puissent avoir aucun contrôle ou modification sur les applications. De nos jours, les systèmes sont face aux plusieurs attaques déguisées comme des utilisateurs confiés. Les politiques de sécurité au niveau du réseau autorisent uniquement les vrais utilisateurs à accéder à une adresse IP spécifique. Mais ces politiques sont devenues obsolètes, car la sécurité des systèmes ait été violée par des utilisateurs déguisés.

Avec les récents progrès technologiques, différentes méthodes et techniques a été réalisés pour faire face aux problèmes de sécurité augmentés. Les menaces au niveau des applications se produisent dynamiquement et de façon adaptable aux contrôles de sécurité. Un travail dans [110] consiste à développer un dispositif ASIC orienté tâche, il est capable de gérer une tâche spécifique offrant des niveaux de sécurité plus élevés avec des performances plus élevées.

Même dans l'environnement virtuel, des entreprises telles que VMware, etc. utilisent la technologie de virtualisation Intel pour améliorer les performances et la sécurité. Il a été observé que le plus souvent des sites Web sont sécurisés au niveau du réseau et disposent des mesures solides de sécurité, mais il peut exister des failles de sécurité au niveau de l'application qui peuvent permettre l'accès non autorisé.

Les menaces sur la sécurité des applications incluent les attaques XSS, l'empoisonnement des cookies, la manipulation de champs cachés, les attaques par injection SQL, les attaques par déni de service, les options de porte dérobée et de débogage, la rupture de CAPTCHA, etc., résultants à l'utilisation non autorisée des applications [111].

II.4.3.1 *Problèmes de sécurité liés à l'hyperviseur*

Le Cloud Computing repose principalement sur le concept de virtualisation. Dans un monde virtualisé, l'hyperviseur est généralement défini comme un contrôleur qui est connu par le gestionnaire de la machine virtuelle qui permet à plusieurs systèmes d'exploitations de s'exécuter sur un système à la fois, en fournissant des ressources à chaque système d'exploitation de manière non interférent. Alors que le nombre de systèmes d'exploitation fonctionnant sur une unité matérielle est augmenté, les problèmes de sécurité liés à ces nouveaux systèmes d'exploitation doivent être également pris en compte.

En addition, il peut arriver qu'un système invité essaie d'exécuter un code malveillant sur le système hôte qui est devenu inactif, et par conséquent il perd le contrôle total et bloque l'accès aux autres systèmes d'exploitation invités. On ne peut nier qu'il existe des risques associés au partage d'une même infrastructure physique entre plusieurs utilisateurs. Même un utilisateur malveillant peut entraîner des menaces pour les autres utilisateurs de la même infrastructure [112].

Par conséquent, la sécurité vis-à-vis de l'hyperviseur est un sujet de grande préoccupation, car il contrôle tous les systèmes invités. Sur la base de l'apprentissage du comportement des différents

composants de l'architecture d'hyperviseur, un système de protection de Cloud avancé peut être développé en surveillant les activités des machines virtuelles invitées et l'intercommunication entre les différents composants de l'infrastructure [113, 114].

II.4.3.2 *Attaque par déni de service*

Une attaque par déni de service DoS est une tentative visant à rendre les services attribués aux utilisateurs autorisés inutilisables. Dans une telle attaque, le serveur fournissant le service est surchargé par un grand nombre de requêtes et, par conséquent, le service devient indisponible pour l'utilisateur autorisé. Parfois, lorsque nous essayons d'accéder à un site, nous constatons qu'en raison d'une surcharge du serveur avec les demandes d'accès au site, nous ne pouvons pas accéder et nous ne constatons pas d'erreur. Cela se produit lorsque le nombre de demandes pouvant être traitées par un serveur dépasse sa capacité.

L'attaque DoS pourrait utiliser certaines des techniques suivantes de submerger les ressources d'une Cloud telles que le remplissage de l'espace disque de stockage à l'aide d'énormes fichiers, ou l'envoi d'un message qui réinitialise un masque de sous-réseau de l'hôte cible, provoquant une perturbation de sous-réseau de routage du Target [101]. L'utilisation d'un système de détection d'intrusion (IDS) est la méthode de défense la plus répandue contre ce type d'attaques [115]. Une fédération de défense est utilisée dans [116] pour se protéger contre de telles attaques. Chaque Cloud est chargé avec un IDS séparé. Les différents systèmes de détection d'intrusion fonctionnent sur la base d'un échange d'informations. Si un Cloud spécifique est attaqué, la coopérative IDS alerte le système dans son ensemble.

II.4.3.3 Attaque par déni de service distribué

Le "Distributed denial-of-service" ou déni de service distribué peut être appelé une version avancée de DoS, c'est un type d'attaque qui vise à rendre muette une machine en la fermant de trafic inutile. Plusieurs machines à la fois sont à l'origine de cette attaque (c'est une attaque distribuée) visant à anéantir des sous-réseaux, des serveurs, etc. Le Cloud, spécialement les logiciels SaaS, ont été les cibles les plus fréquentes de ce type d'attaque, elle reste très difficile à éviter ou à contrer. C'est pour cela que cette attaque représente une menace que beaucoup craignent [117].

Il est nécessaire d'étudier les outils les plus importants dans ce domaine pour mieux comprendre ce type d'attaque, qui doivent leur notoriété à des célèbres attaques ayant visé des grands sites sur l'internet. Dans DDoS, l'attaque est relayée par différents réseaux dynamiques qui ont déjà été compromis contrairement au DOS. Les ttaquants ont le pouvoir de contrôler le flux d'informations en permettant à certaines informations d'être disponibles à certains moments. Ainsi, le volume et le type d'informations disponibles pour un usage public sont clairement sous le contrôle de l'attaquant. Un réseau typique dans un attaque DDoS se compose en trois unités fonctionnelles: Un Maître (point central), des esclaves (de nombreux hôtes distants ou démons) et une victime. Pendant la phase de déroulement de l'attaque, les hackers se connectent au maître qui envoie alors un ordre à tous les hôtes distants qui procèdent comme un tableau de bord pour le

Maître (via TCP, UDP ou ICMP). Toutes les communications entre ces parties peuvent également dans certains cas être chiffrées. Ensuite, les hôtes distants vont attaquer la cible finale suivant la technique choisie par les hackers [117]. Ils vont par exemple se mettre à envoyer un maximum de paquets UDP sur des ports spécifiés de la machine cible. Le nombre important des paquets va submerger la cible qui ne pourra plus répondre à aucune autre requête (d'où le terme de déni de service distribuée). D'autres exemples d'attaques existent, comme le SYN flood (TCP), les attaques de type smurf, l'ICMP flood, les attaques de déni de service dites agressives (dont le but est bel et bien de faire crasher complètement la cible), les attaques dites furtives, ou encore des attaques de type "stream attack" (TCP ACK sur des ports au hasard), etc.

En général, les approches utilisées pour défendre contre les attaques *DDoS* impliquent une modification en profondeur du réseau. Ces modifications deviennent souvent coûteuses pour les utilisateurs. Les chercheurs dans [117] ont proposé une logique basée sur les essaims pour surveiller contre les attaques DDoS. Cette logique fournit une couche de transport transparente, grâce à laquelle les protocoles courants tels que HTTP, HTTPS, SMTP, etc. peuvent fonctionner facilement. L'utilisation d'IDS dans la machine virtuelle est proposée dans [118] pour protéger le Cloud contre les attaques DDoS. Un mécanisme de détection d'intrusion de type SNORT est chargé sur la machine virtuelle pour détecter tous les trafics entrés ou sortis. Une autre méthode couramment utilisée pour protéger contre ces attaques DDoS consiste à installer des systèmes de détection d'intrusion sur toutes les machines physiques contenant les machines virtuelles de l'utilisateur [119].

II.4.3.4 *Empoisonnement par les cookies*

Un cookie est un fichier texte, qui contient essentiellement les informations d'identification liées à l'identité de l'utilisateur et qui est stocké par un site web qui est visité sur le disque dur. Ce stockage est réalisé par un navigateur (voir Figure II-5). Les cookies sont donc des données sensibles d'un point de vue de la sécurité. Il faut donc les considérer comme des données personnelles que personne ne doit obtenir.

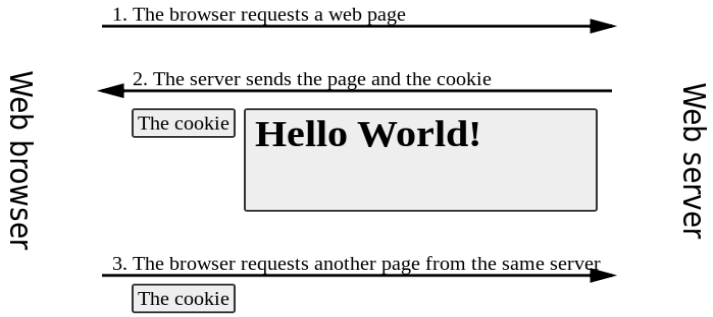


Figure II-5 : Empoisonnement par les cookies

Le but du l'attaquant, est donc généralement de voler le cookie de sa victime pour exploiter son contenu. Cela implique de changer ou de modifier le contenu du cookie pour permettre un

accès non autorisé à une page Web ou à une application. Malheureusement, plusieurs failles permettent de voler des cookies ; à titre d'exemple le vol par accès physique à la machine. Si le hacker a un accès physiquement à la machine, rien de plus simple que de récupérer les cookies. Il va dans le répertoire de stockage des cookies en fonction du système d'exploitation et copie les cookies. Autre exemple c'est le vol par sniffing ou par Man-in-the-middle où le les cookies passent par les requêtes HTTP. Si l'attaquant peut intercepter les requêtes HTTP, soit à l'aide d'un sniffer, soit par attaque par le milieu, il peut donc récupérer tous les cookies sans aucun problème ; à condition bien sûr que le flux HTTP ne soit pas chiffré (HTTPS) [120]. Ce type d'attaque peut être évité en effectuant un nettoyage régulier des cookies ou en mettant en place un système de chiffrement des données du cookie [121].

II.4.3.5 *Attaque de manipulation de champ caché*

La manipulation cachée est principalement centrée sur les sites Web de commerce électronique. Lors de l'accès à une page Web, certains champs sont masqués et contiennent les informations relatives à la page. Ils sont essentiellement utilisés par les développeurs. Cependant, ces champs sont fortement objectifs aux attaques de hackers car ils peuvent être facilement modifiés et postés sur la page Web. Cela peut entraîner de graves violations de sécurité [122].

Les applications Web utilisent les entrées des requêtes HTTP (et parfois des fichiers) pour déterminer comment répondre. Les attaquants peuvent modifier n'importe quelle partie d'une requête HTTP, y compris l'URL, la chaîne de requête, les en-têtes, les cookies, les champs de formulaire et les champs cachés, pour tenter de contourner les mécanismes de sécurité du site. Lorsqu'un utilisateur effectue une sélection sur une page HTML, la sélection est généralement stockée sous forme de valeurs de champ et envoyée à l'application sous forme de requête HTTP (GET ou POST). L'attaquant peut examiner le code HTML de la page et modifier les valeurs des champs masqués afin de modifier les requêtes envoyées vers le serveur, l'exemple suivant explique ces étapes de l'attaque :

Code HTML vulnérable

```
<form method="post" action="page.aspx">
<input type="hidden" name="PRICE" value="200">
Product name : <input type="text" name="product" value="shop" name="product"
value="Shirt"><br>
Product price:200.00"><br>
<input type="submit" value="submit">
</form>
```

Requête normale :

```
http://www.shop.com/page.aspx?product=Shirt&price=200
```

Requête malveillante :

```
http://www.shop.com/page.aspx?product=Shirt&price=2
```

II.4.3.6 Options de Backdoor et de débogage

Les développeurs, dans l'habitude, activent l'option de Backdoor (porte dérobée) avant de la publication d'un site Web, cela leur permettent d'apporter des modifications dans le développement du code et de les implémenter dans le site Web. Puisque ces options facilitent l'entrée des développeurs dans l'arrière-plan, et que parfois ces options sont ignorés, Cela pourrait être la cause de l'attaque des pirates [123]. Par conséquent, une attention particulière doit être portée à ces options afin d'éviter ce problème.

II.4.3.7 CAPTCHA rupture

La définition de CAPTCHA est un test dans lequel l'utilisateur d'un site Web est obligé à déchiffrer une image ou un audio déformées qui sont utilisés pour protéger ce site Web contre les attaques automatisées. Les CAPTCHA ont été développés pour empêcher les bots ou les processus robots d'utiliser les ressources. Ces bots sont généralement des spams ou des surexploitations des ressources du réseau. Dernièrement, les spammeurs sont capables de casser le CAPTCHA [115]. Ils utilisent un système audio capable de lire les caractères CAPTCHA pour les utilisateurs malvoyants et utilisent un logiciel de conversion de parole en texte pour faire échouer le test.

II.4.4 Sécurité physique

La sécurité physique est très importante plus que tout autre contrôle qui vise à protéger la sécurité du Cloud. Les locaux ou le bâtiment dans lequel les Datacenters du Cloud sont hébergés, font aussi l'objet de différentes menaces, y compris les actions humaines et les dangers naturels. Poser les Datacenter dans une zone de désastres est aussi dangereux qu'accorder des accès privilégiés à tous les utilisateurs. Pour cela la sécurité physique d'un local doit être vue comme un système de protection pour mettre en place une défense multi-niveaux ou multicouches dont chaque couche est associée à un contrôle général automatisé.

L'organisation d'une sécurité physique efficace nécessite une conception environnementale qui prend en compte les activités normales et les situations imprévues. Les éléments de la sécurité physique doivent être soutenus par des procédures appropriées, par exemple chaque centre de données ou zone sensibles (salles de serveurs, armoires réseau, armoires utilitaires) requiert au moins une authentification basée sur un lecteur de badge. Les tentatives d'accès infructueuses répétées déclenchent une alerte pour les gardes de sécurité. Les essais d'accès sont enregistrés, et les journaux de vidéosurveillance et les journaux de l'historiques d'accès ont aussi conservés pendant des périodes déterminés. L'étendue des difficultés de sécurité physique est vaste et implique de nombreuses mesures pour empêcher, découvrir et répondre aux accès non autorisés aux locaux, aux ressources ou aux informations présentes dans les locaux [124].

II.4.5 Sécurité des données

La protection de données dans le Cloud s'apparente comme la sécurité de données de façon traditionnelle, mais à cause des caractéristiques de l'ouverture et de la multi-locataires du Cloud, la sécurité de données contenues dans le Cloud a ses particularités. Le problème de la sécurisation et de la confidentialité de données est montré dans le cycle de vie de la sécurité des données [125] [126]. Le cycle de vie des données concerne l'ensemble du processus, de la génération à la destruction de données. Le cycle de vie de données est divisé en six étapes [127, 128] :

II.4.5.1 Cycle de vie des données

- **Création**

  La création est la génération d'un nouvel élément ou la modification d'un élément de données numériques existe. On peut donc l'appeler également en tant que phase de création / mise à jour. Il peut s'agir de tout type de contenu, pas seulement d'un document ou d'une base de données c'est-à-dire peut être structuré ou non structuré. Dans cette phase, les données sont classées et les droits appropriés sont déterminés.

- **Stockage**

  Le stockage est l'action pour former les données numériques selon un type référentiel de stockage structuré ou non structuré (base de données ou fichier). Cette opération se produit généralement en même temps que la création. Ici, la classification et les droits des contrôles de sécurité doivent être mappés, y compris les contrôles d'accès, le chiffrement et la gestion des droits.

- **Utilisation**

  Les données sont visualisées, traitées ou utilisées dans une manière où ces données originales ne sont pas modifiées. Ces activités s'appliquent généralement aux données stockées au moment de l'utilisation d'après d'un PC ou d'une application de l'utilisateur. Pour assurer ce type d'activité, il existe des contrôles de détection tels que la surveillance d'activité, des contrôles préventifs tels que la gestion des droits et des contrôles logiques qui sont généralement appliqués dans les bases de données et les applications.

- **Partage**

  Les données sont rendues accessibles aux autres, et elles sont échangées entre les utilisateurs, les clients et les partenaires. Les contrôles de cette phase incluent une combinaison des opérations de détection et de prévention, de cryptage pour un échange sécurisé des données, des contrôles logiques ainsi que la sécurité des applications.

- **Archivage**

  Les données restent sans utilisation et entrent dans la mémoire à long terme doivent être archivées, ici, la protection des données et leur disponibilité sont assurées par une combinaison de gestion de cryptage et de gestion des bénéfices.

- **Suppression**

  Les données sont détruites de manière permanente à l'aide des moyens physiques ou logiques. Les données doivent être supprimées de manière sécurisée et doivent être utilisées des outils pour retrouver les copies permanentes.

II.4.5.2 Localisation et accès

La grande lacune dans le cycle de vie de données est qu'elle ne montre pas correctement l'emplacement de données lorsqu'elles se déplacent entre les entrepôts, les environnements et les organisations, ni comment on y accède au cours de ces phases. Le but de la localisation peut être représenté en considérant le cycle de vie comme une série de cycles de vie plus petits s'exécutant dans des environnements d'exploitation différents et non comme une opération linéaire unique. Les données peuvent être donc déplacées dans, hors et entre ces environnements à n'importe quelle phase de leur cycle de vie.

Une sécurité élevée de données peut être obtenue en identifiant ces mouvements et en appliquant les contrôles appropriés aux limites de sécurité appropriées. Ces environnements peuvent être des Clouds internes, externes, publics ou privés, des fournisseurs de Cloud ou des sous-traitants traditionnels, etc. Pour cela, il est très important de comprendre les emplacements logiques et physiques des données [127].

II.4.5.3 Aspects de la sécurité des données

Pour assurer la sécurité des données dans le Cloud, il faut une bonne compréhension des différentes étapes et approches de la transmission de données. Dans ce stade, on parle sur les notions : Données en transit, Données en reste, Données en traitement (y compris la multi-location), Lignage de données, Provenance de données et Rémanence de données. Les trois premières notions peuvent être comprise dans l'exemple d'utilisation d'un courrier électronique, où l'envoi d'un message électronique représente les données en transit, les données en reste sont les messages électroniques dans la boîte aux lettres et le traitement des données désigne la saisie d'une réponse [129].

Il est recommandé de protéger les données en transit à l'aide d'un algorithme de chiffrement, en particulier lors de l'utilisation d'un Cloud public. L'utilisation de données chiffrées avec un protocole non sécurisé peut fournir la confidentialité, mais ne garantit pas l'intégrité des données. Par conséquent, il est essentiel d'utiliser un protocole assurant l'intégrité, tel que FTPS, HTTPS, etc. [130]. Le chiffrement de données en reste n'est pas aussi simple que celui des données en transit. Le chiffrement est possible s'il n'est pas nécessaire d'utiliser l'indexation, le traitement et la recherche aux données. Toutefois, lorsque les applications fonctionnent avec le traitement de données, ces données doivent être non chiffrées. Un schéma de chiffrement complètement homomorphique est développé dans [131], ce qui permet de traiter les données sans les déchiffrer.

Lorsque les données sont dans le Cloud, il est recommandé de suivre ses historiques, pour savoir exactement où et quand les données ont été localisées dans le Cloud. Une analyse de lignage de données peut être réalisée avec les outils ETL15 et permet de suivre les modifications de données en présentant une série de dépendances entrée-sortie de données dans un environnement tel qu'un graphe de nœuds et liens. Cette visualisation de chemin de données s'appelle lignage de données. Bien que la fourniture du lignage de données soit très importante pour l'assurance de l'audit, ce processus prend beaucoup de temps et n'est pas vraiment possible pour un Cloud public. Un problème plus complexe pour les clients est de fournir une provenance plus précise des données. La provenance des données représente une sorte de métadonnées, contenant l'historique de dérivation d'une production de données à partir de ses sources d'origine dans un entrepôt de données [132].

Le dernier aspect de la sécurité des données est la rémanence des données ou la représentation résiduelle des données numériques. Par exemple, les données ne doivent pas être disparus même après des tentatives d'effacement. Certains systèmes d'exploitation ne suppriment pas les données immédiatement lorsque l'utilisateur demande, mais les déplacent vers une zone de stockage afin de pouvoir les récupérer facilement en cas d'écrasement ou d'erreur [133]. Le Cloud Computing avec ses caractéristiques de la virtualisation complique la rémanence des données. Il est pratiquement impossible de remplacer le support physique, car l'infrastructure en Cloud peut répartir le stockage du client ou l'instance de machine virtuelle sur plusieurs lecteurs physiques.

De ce qui précède, on peut dire que la seule option réalisable pour faciliter la sécurité des données consiste à s'assurer que les données sensibles ne seront pas stockées dans un Cloud public ou on utilise la technique de chiffrement avec des clés stockées localement.

II.5 Conclusion

Le défis de sécurité devrait être réglés avant que les utilisateurs puissent profiter de tous les avantages du Cloud Computing et y placer leur confiance. En effet, le renforcement de la sécurité et les pratiques de confidentialité vont attirer plus d'entreprises au monde de l'informatique dans le Cloud.

Dans ce chapitre, nous avons identifié les différentes exigences de sécurité du Cloud. La sécurité est généralement liée aux trois aspects principaux qui sont la confidentialité, l'intégrité et la disponibilité (***CID***), et cela du côté client comme du côté fournisseur du Cloud. Nous avons ensuite identifié les problèmes de gestion et de contrôle de la sécurité du Cloud et les principaux impactes de la sécurité.

Nous avons également identifié les menaces de la sécurité du Cloud qui sont classé à différents niveaux, tels que le niveau de base, le niveau réseau, le niveau application, le niveau physique ou données. Et selon ces différents niveaux, des différentes violations de sécurité peuvent survenir.

Actuellement, nous aborderons d'autres problématiques de la sécurité du Cloud Computing notamment, la sécurité des données et les bonnes stratégies pour avoir un niveau adéquat de la confiance dans le Cloud Computing. Nous pensons aussi à créer deux approches : la première permet de garantir la confiance entre les fournisseurs et les utilisateurs du Cloud, et la deuxième permet de proposer une approche solide pour assurer une communication sécurisée dans le Cloud.

Chapitre III

Contribution 1 : Une technique de chiffrement complétement homomorphique préservant l'ordre adaptée au Cloud Computing

III.1 Introduction

La génération volumineuse de données, dans la dernière décennie, a créée de nombreux défis qui ont conduit les chercheurs à concevoir de nouvelles techniques permettant de stocker et de protéger cette vaste quantité de données. Selon un rapport d'IBM [134]: « 90% des données dans le monde ont été créées au cours des dernières années et dans chaque jour il y a une création de plus de 2,5 milliards octets ». En effet, cette augmentation de données générées offre des avantages particuliers dans les systèmes dynamiques et évolutifs. Le principe du service Cloud est que les entités génératrices de données doivent partager leurs données générées entre elles. Cette raison a conduit à utiliser les services de Cloud tel qu'un serveur en cloud. Les serveurs cloud ou généralement les XaaS offre de nombreux avantages tels qu'un grand espace de stockage, une vitesse de calcul mais ils sont généralement considérés comme des entités non confiantes surtout pour des données critiques. Par conséquent, les recherches dans ce contexte traitent la problématique de protection de données stockées sur le serveur Cloud en proposant des solutions adaptatives aux environnements Cloud pour répondre aux besoins requis.

Pour sécuriser les données stockée dans le cloud, plusieurs techniques de chiffrement ont été proposées [135, 136]. Dans [137] les données sont uniquement chiffrées lors de la phase du transfert, celles-ci doivent être en clair lorsqu'elles sont stockées dans le Cloud. Cela garantit la confidentialité durant la phase transfert mais les données sont stockées en clair dans le Cloud. Dans [138] les données sont chiffrées et stockées chiffrée dans le Cloud. Dans ce cas, on ne peut pas exécuter des opérations dans le Cloud ; par exemple, une recherche ou des opérations arithmétique sur les données chiffrées. Pour effectuer des opérations récupérer les données, déchiffrer, appliquer les opérations, rechiffrer et renvoyer s'il est nécessaire. Ce mode de fonctionnement sur les données chiffrées n'est pas efficace et engendre des latences importantes de traitement et consomme les ressources matériels et réseaux (Bande passante, capacité de calcul, RAM, routage…). Pour répondre à ces limites, plusieurs solutions théoriques ont été proposées ; il s'agit des systèmes de chiffrements homomorphiques [139, 140, 141, 142, 143]. Ces systèmes de chiffrement permettent de traiter des requêtes et des opérations sur les données chiffrées sans

devoir les récupérer localement, les déchiffrer, ensuite les chiffrer de nouveau et les renvoyer au Cloud. La limite principale de ces techniques est qu'elles ne garantissent pas l'ordre entre les données.

L'objectif principal de travail présenté dans ce chapitre est de proposer une solution pour protéger les données en fournissant un schéma de chiffrement permettant de conserver certaines opérations telles que l'addition, la multiplication et la préservation de l'ordre sur les données chiffrées dans le Cloud. Cette méthode permettra de traiter les opérations sur les données en utilisant des langages standard, tel que SQL.

III.2 Techniques de sécurité de données dans le Cloud

Plusieurs techniques et solutions ont été proposées pour sécuriser les données dans le Cloud. Ces solutions peuvent être classées selon plusieurs critères, comme illustré dans la *Figure III-1*, où la taxonomie des techniques de sécurité repose sur la stratégie de sécurité et le principe de chiffrement inspiré.

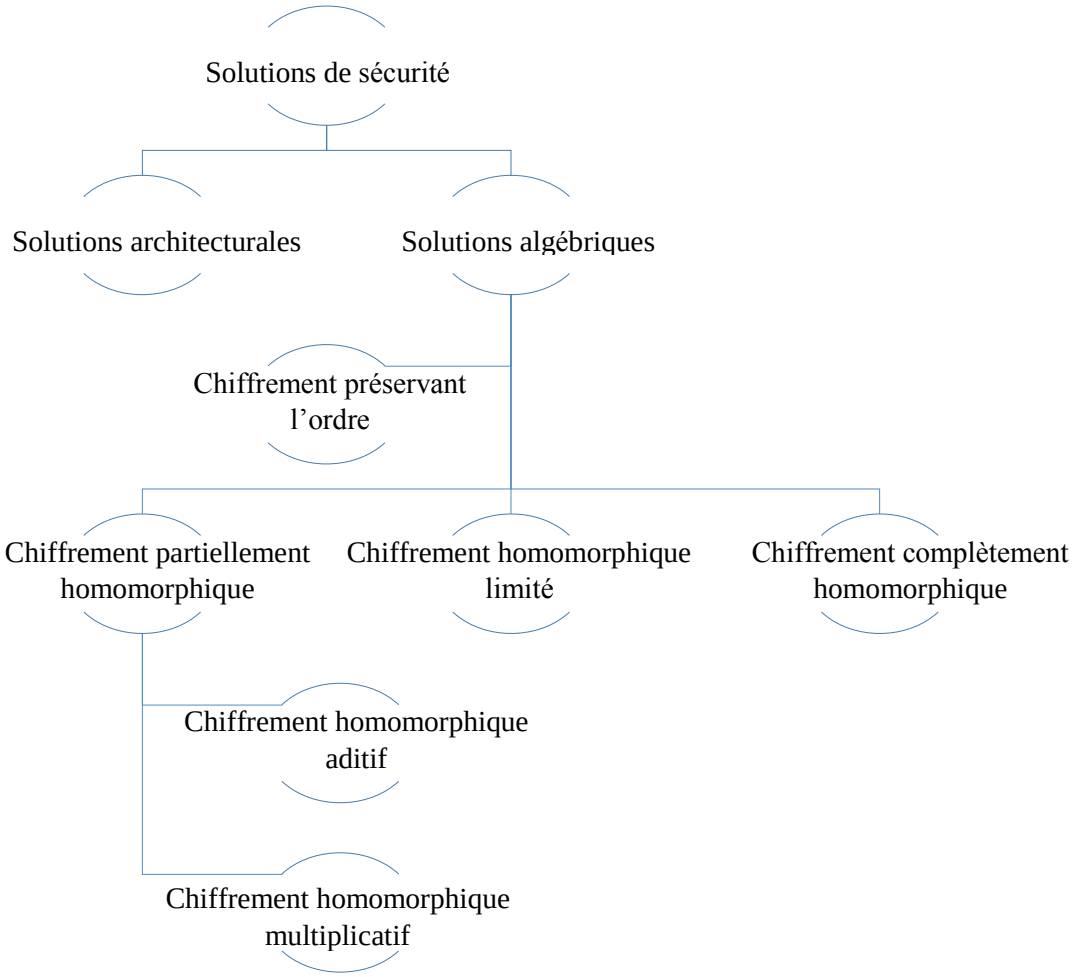


Figure III-1 : Taxonomie sur les solutions de sécurité des données dans le Cloud

III.2.1 Solutions architecturales de multi-Cloud

Des travaux récents [144, 145, 146, 147] définissent des architectures de multi-Cloud permettant de maintenir la sécurité et d'atténuer les limites de sécurité dans le Cloud. Ces architectures sont adaptées aux plates-formes Cloud et n'introduisent pas généralement de proxy intermédiaire ou de serveur broker entre le client et le fournisseur de Cloud. L'utilisation de mécanisme multi-Cloud peut révéler diverses théories afin de cibler différents aspects de la sécurité de confidentialité, intégrité, consistance et cohérence de données stockées dans différents Clouds. Cette classe utilise le principe de distribution de manière systématique des données utilisateur sur plusieurs fournisseurs Cloud.

Le travail dans [148] fournit une construction de deux Clouds avec une série de protocoles de communication pour un service de base de données externalisé. La technique proposée garantie la protection et la confidentialité de données, les propriétés statistiques et la conception des requêtes. Le système proposé comprend un administrateur de base de données et deux Clouds non complotés. Dans ce modèle, l'administrateur de base de données peut être implémenté au côté du client. Les deux Clouds (notés Cloud A et Cloud B) fournissent le stockage et le service de calcul. Les deux Clouds travaillent ensemble pour répondre à chaque demande d'interrogation du client / des utilisateurs autorisés.

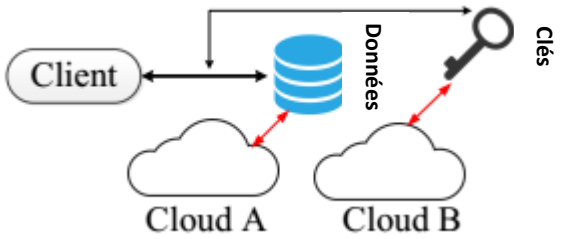


Figure III-2 : Architecture avec deux Clouds

Comme le montre la Figure III-2, pour agir avec une base de données sécurisée, les données sont chiffrées et externalisées pour être stockées dans un Cloud (Cloud A) et les clés privées sont stockées dans l'autre (Cloud B). Les résultats expérimentaux démontrent que l'approche répond aux exigences de confidentialité et la méthode proposée est efficace dans les bases de données stables, tandis que l'efficacité de l'approche diminue dès que le volume de bases de données augmente. Dans ce cas la complexité de l'approche est plus élevée et l'espace mémoire nécessaire est très augmenté et le traitement des requetés ne couvre pas certaines opérations telles que l'opération Somme.

Les auteurs dans [149] ont proposé un schéma de base de données en tant que service (SecureDBaaS) dans lequel les données sont conservées dans un état confidentiel et sécurisé. Ce travail prend en charge l'exécution des opérations simultanées et indépendantes sur une base de données chiffrée dans le Cloud. SecureDBaaS implémente de nombreuses techniques pour convertir les données de texte en clair vers données, métadonnées ou structures de données

chiffrées. Dans SecureDBaaS, il existe principalement trois opérations dont la première est la phase d'installation, dans laquelle la table de stockage de métadonnées est créée. Cette métadonnée contient les types de données, les clés principales ainsi que les techniques de chiffrement. La deuxième phase est la phase de l'utilisation qui consiste à générer les requêtes SQL exécutable et qui ont lieu après l'authentification d'un utilisateur. Un client de SecureDBaaS analyse l'opération et récupère les métadonnées pour réaliser la communication. Les informations des métadonnées sont utilisées pour convertir le SQL en clair vers la requête exécutable dans le Cloud. Enfin, dans la phase finale, les résultats sont ensuite déchiffrés et remis à l'utilisateur. Cette approche est basée sur l'exécution d'opérations de chiffrement via des algorithmes de chiffrement SQL-aware, qui avait été initialement proposée dans [150]. Il s'agit donc des techniques de chiffrement coûteux, et la table de stockage des métadonnées est située sur un serveur non confiant, ce qui pose d'énormes problèmes pour la sécurité de ce schéma.

III.2.2 Solutions algébriques

Les limitations, essentiellement la confidentialité, obligent les chercheurs à proposer d'autres techniques de protection de données. Certains travaux exploitent un ou plusieurs techniques de chiffrement en Cloud et par conséquent la notion la plus connue « Homomorphique ». Dans les techniques proposées, nous pouvons étudier les cryptosystèmes de chiffrement partiellement homomorphique, chiffrement homomorphique limité et chiffrement complètement homomorphique. Le homomorphique est une propriété qui permet de confier des calculs dans le Cloud, sans que les données ni les résultats soient accessibles au fournisseur de Cloud.

III.2.2.1 Chiffrement partiellement homomorphique

Le chiffrement partiellement homomorphique (PHE) permet d'exécuter quelques opérations sur des données chiffrées telles que la multiplication ou l'addition, mais pas les deux. Dans le passé, la communauté cryptographique disposait des systèmes de chiffrement avancés et qui sont partiellement homomorphique, dans lesquels certaines opérations arithmétiques sont effectuées sur des données chiffrées. Nous pouvons classer ces cryptosystèmes en deux catégories. La première est donnée par des schémas de chiffrement homomorphique additifs qui permettent d'effectuer des additions limitées ou illimitées sur des données chiffrées, comme dans dans [151, 152, 153, 154, 155]. La seconde contient des schémas de chiffrement homomorphique multiplicatifs qui permettent d'effectuer des multiplications limitées ou illimitées sur des données chiffrées, comme dans [156, 157, 158]. Cependant, les systèmes PHE ne sont pas vraiment utiles car pratiquement toutes les requêtes du monde réel impliqueraient plusieurs opérations allant au-delà de ce qui peut être contrôlé par les systèmes PHE.

Une méthode de préservation de la confidentialité sécurisée (ESPP) a pour objectif d'externaliser les données d'un dispositif mobile contrôlé par des ressources dans le Cloud Computing a été proposée dans [159]. La technique est basée sur une méthode de chiffrement à clé publique probabiliste. Le procédé demande moins d'efforts de calcul et permet de récupérer le texte en clair sans laisser de fuites d'informations. Il utilise une recherche par mot-clé avec un

classement de rappel sur les données chiffrées afin de récupérer le fichier d'origine dans le Cloud, ce qui permet aux fournisseurs de Cloud de conclure le fichier requis et de le renvoyer sans aucune information expressive. Cette approche préserve la confidentialité des données de tout menace.

La recherche par mot-clé classé montre un système pratique, qui fournit un classement basé sur le score de pertinence afin de l'utiliser dans le processus de recherche. Le client reçoit le fichier le plus pertinent plutôt que tous les fichiers, ce qui réduit la bande passante et confirme la précision et l'intégrité des données extraites. Cette approche garantit également la confidentialité des données et une faible complexité de calcul. La technique produit un index pour un fichier et conserve les index et les fichiers dans le Cloud de manière chiffrée. Pour récupérer les données, l'utilisateur autorisé génère une requête vers le serveur. Le serveur recherche ensuite les fichiers correspondants et les renvoie à l'utilisateur. La mise en œuvre de la méthode de recherche par mot clé classée de cette approche est inefficace lors de la recherche dans de données volumineuses. Cependant, toute modification, suppression et ajout au données ou index représente un nouveau risque en ce qui concerne l'approche sans surcharge de calcul, ce qui nécessite un processus dynamique et efficace sur les données et les index sans menacé la confidentialité. De plus, du côté serveur, l'approche ne fournit pas la citation de résultat ordonnée car elle ne peut pas être utilisée d'une manière efficace, de manière que le client ne reconnaît pas les données récupérées qui seraient les plus appropriées.

III.2.2.2 Chiffrement homomorphique limité

Le chiffrement homomorphique limité (SomeWhat Homomorphic Encryption SWHE) permet d'exécuter plusieurs multiplications et de plusieurs additions, mais le nombre et le mode des opérations est limité. En termes spécifiques, les systèmes SWHE permettent un nombre illimité d'additions et de multiplications sur des données chiffrées. Le cryptosystème [160] était le premier algorithme permettant de calculer avec des schémas SWHE sur des textes chiffrés. L'addition est autorisée plusieurs fois tandis que la multiplication n'est possible qu'une seule fois. Un autre exemple de schémas SWHE est présenté dans [161]. Il permet de calculer des polynômes multivariés de degré ***d***. Il s'agit d'un système de chiffrement complétement homomorphique à ***d*** multiplication, c'est-à-dire qu'il permet ***d*** multiplications et un nombre illimité d'additions sur des données chiffrées.

Dans [162], une nouvelle technique de gestion de base de données CryptDB, est utilisée pour protéger la confidentialité des données et traiter les requêtes SQL de manière efficace. CryptDB vise à exécuter des requêtes sur une base de données chiffrée et qui est gérée par le fournisseur de Cloud, à l'instant de l'exécution de requêtes sur une base de données non chiffrée. L'architecture CryptDB comprend une partie proxy de base de données et une partie SGBD chiffrée dans le Cloud. La partie proxy est considéré comme un serveur confiant, et qui stocke la clé principale secrète et le schéma de la base de données. Le proxy est utilisé comme couche intermédiaire qui chiffre et déchiffre toutes les données et modifie tous les opérateurs des requêtes. Dans cette approche, les données sont chiffrées en couches, appelées oignon [150]. Le mot « oignon » fait

référence à des couches de chiffrement qui se chevauchent comme les couvertures d'un oignon. Ces oignons ont différentes couches, chacune chiffrée à l'aide d'algorithmes et de clés différents. CryptDB pose un problème majeur, qui consiste à effectuer des calculs au niveau du serveur sur des données chiffrées pour différents objectifs, car les textes en clair sont chiffrés avec des clés différentes. La solution traditionnelle consiste à exécuter le calcul après le déchiffrement des données. CryptDB est beaucoup plus efficace mais ne peut pas prendre en charge la plupart des requêtes analytiques sur des données chiffrées. De plus, il est trop coûteux pour certains calculs, comme pour les agrégats à grande échelle.

Les auteurs dans [163, 164] ont proposé un chiffrement noté dodrantencryption qui utilise un chiffrement déterministe limité en conjonction avec un chiffrement semi-homomorphique et des tables logarithmiques et anti-logarithmiques afin de développer l'ensemble des requêtes SQL numériques. Ils ont proposé une solution palliative pour le chiffrement homomorphique qui utilise une variante déterministe du schéma de chiffrement de Paillier [165] et les grandes tables. Ce travail a permis de calculer des sommes et des produits de sommes, mais pas toutes les requêtes SQL impliquant des valeurs numériques. En particulier, il ne peut pas comparer deux valeurs numériques chiffrées.

III.2.2.3 Chiffrement complètement homomorphique

Le chiffrement complètement homomorphique (Fully homomorphism encryption FHE) est un nouveau concept de sécurité. Ce système peut calculer n'importe quel type de fonction sur des données chiffrées. Pour obtenir des propriétés complètement homomorphique, la plupart des travaux FHE étaient, jusqu'à présent, conçus pour le calcul avec un bruit lors de chiffrement. L'évaluation homomorphique amplifie le bruit de chiffrement pour augmenter la marge d'erreur. L'erreur grandit jusqu'à ce qu'il soit impossible d'obtenir une évaluation homomorphique plus précise ou que le déchiffrement soit impossible. Pour réduire l'augmentation de bruit dans le texte chiffré, plusieurs techniques sont utilisées.

Le bootstrapping [166] est une technique permettant de « rafraîchir » périodiquement les texte chiffrés associés aux nœuds intérieurs d'un circuit de chiffrement. En pratique, cette technique est coûteuse car elle sera appelée après chaque multiplication. La technique de [167] a introduit une nouvelle procédure de gestion du bruit. Il consiste à réduire le bruit en convertissant un texte chiffré par réduction modulo. Cette technique permet de transformer un texte chiffré en un autre texte chiffré en utilisant deux clés différentes. En raison de cette réduction, les schémas de chiffrement complétement homomorphiques sont toujours coûteux et lents en termes de calcul.

Les auteurs de [168] proposent une solution pour le traitement de requêtes algébriques de données chiffrées et d'autres défis liés aux données chiffrées avec chiffrement homomorphique. En outre, ils fournissent une base de données sécurisée en tant que service pour les clients. Le travail décrit comment effectuer et traiter toute opération algébrique sur des données chiffrées. Le document montre les avantages du chiffrement FHE, qui permet de réaliser tout processus sur des données chiffrées. De plus, il indique comment appliquer FHE pour obtenir un facteur de sécurité

robuste afin que le fournisseur de service puisse effectuer toute requête. En fait, les auteurs se concentrent sur deux facteurs à atteindre pour réaliser FHE: 1) la garantie de sécurité et 2) les capacités de traitement des requêtes. Le travail fournit une bonne solution pour la recherche qui comprend quatre parties: (a) un modèle de données (b) un modèle de calcul, (c) des algorithmes pour différents opérateurs d'algèbre relationnelle et (d) des méthodes pour transférer les résultats des requêtes vers le client. Cette solution fournit un facteur de sécurité par rapport à l'attaque en texte clair, et l'utilisateur est élaboré dans la phase de chiffrement des données et des requêtes et la phase de déchiffrement des résultats de requête. Cependant, le travail a des manques dans les aspects pratiques du FHE et dans les issues d'authentification des utilisateurs, ainsi que dans la manière de construction des index. Le fournisseur ne peut traiter que des processus partiels et le client effectue le reste du traitement. De plus, l'approche conduit à un espace mémoire important et à une complexité de calcul élevée. La récupération des données chez le client nécessite de nombreuses étapes pour obtenir les fichiers obligatoires.

III.2.2.4 Chiffrement préservant l'ordre

La propriété du chiffrement préservant l'ordre garantit la relation d'ordre entre les éléments de données en fonction de leurs valeurs chiffrées, sans révéler les données elles-mêmes (c'est-à-dire que l'ordre entre les valeurs de texte en clair est conservé après le chiffrement). Cela permet de créer des index sur des données chiffrées qui peuvent être utilisées pour des requêtes d'ordre. Il existe de nombreuses approches pour établir cette propriété, comme [169, 170, 171] qui utilisent des fonctions linéaires et non linéaires pour indexer les données.

En conclusion, les techniques de sécurité homomorphiques peuvent être assurées par des mécanismes cryptographiques et mathématiques distribués. La sélection du mécanisme le plus approprié est l'une des parties vitales du système. Le mécanisme sélectionné doit répondre aux contraintes des utilisateurs telles que la taille des clés, la taille des données, le temps de traitement et les caractéristiques des données tel que l'ordre, la possibilité de réalisation des opérations arithmétiques, etc.

Dans ce chapitre, nous proposons un chiffrement complètement homomorphique qui préserve l'ordres dans un schéma construit par des simples expressions modulaires et linéaires de la forme $(a * x + b) \bmod p$. La forme des expressions est publique, mais les coefficients a, b et p sont gardés secrets (non connus par une troisième partie). Notre schéma de chiffrement et d'indexation est théoriquement sécurisé, puisqu'un attaquant ne peut pas obtenir suffisamment d'informations pour résoudre les équations linéaires à partir des valeurs d'entrée et des informations générées. La section suivante présente le détail de la proposition.

III.3 Solution proposée

Les besoins indispensables d'utilisation d'une technique de chiffrement qui assure spécialement l'homomorphique et la propriété de l'ordre entre les données chiffrées est obligatoire. Cette obligation est justifiée par la nécessité de résoudre les problèmes de confidentialité et de manque de confiance entre les utilisateurs et les fournisseurs de Cloud. A titre d'étude, la base de données peut rester confidentielle tant qu'elle est traitée par un utilisateur authentifié d'une manière confiante. Pour cet objectif, nous proposons des processus utiles pouvant être accomplis dans une base de données résidant dans des environnements Cloud qui sont non confiants. Par conséquent, les données doivent être chiffrées et les opérations dans le Cloud doivent s'exécuter aussi de manière chiffrées.

Pour cette raison, cette section a pour but de trouver une méthode englobant le calcul sur les données chiffrées. Principalement, la technique FHE proposée permet des calculs arbitraires sur des données chiffrées. Cette section explique les détails de notre proposition qui sera présenté dans deux parties : la première partie décrit le modèle formel de notre proposition et la deuxième explique l'architecture de la proposition.

III.3.1 Modèle préliminaire et formel

La définition d'un homomorphisme est donnée par l'explication suivante :

Soit deux groupes (M, Δ) *et* $(C, \circ)$. On note e_M (respectivement e_C) l'élément neutre de M (respectivement de C). Une application $f : M \rightarrow C$ est un homomorphisme de groupe si:

- Si x et y sont deux éléments de M, alors $f(x \Delta y) = f(x) \circ f(y)$.
- $f(e_M) = e_C$

Propriété :

Soit $f : M \rightarrow C$ un homomorphisme de groupes.
Alors :

1) $f(1_M)=1_C$
2) Pour tout élément x de M, $f(x^{-1})=f(x)^{-1}$
3) Pour tout entier non nul n, $f(x^n)= f(x)^n$
4) Pour tout entier non nul n, $f(x^{-n})=f(x)^{-n}$

Un système de chiffrement complètement homomorphique n'est rien d'autre qu'un système de chiffrement homomorphe où toute fonction peut être évaluée sur les données chiffrées. Comme toute fonction peut être exprimée comme un polynôme et qu'un polynôme consiste en une série d'additions et de multiplications, un système de chiffrement sera complètement homomorphe dès qu'il permettra d'évaluer un nombre arbitraire d'additions et de multiplications sur les données chiffrées.

Formellement, si c_1 et c_2 sont deux éléments d'un groupe noté C et qui sont les chiffrés de m_1 et m_2 respectivement, où m_1 et m_2 deux éléments d'un groupe noté M. Une fonction de chiffrement $f : M \rightarrow C$ est un homomorphisme s'il existe deux opérations Δ et $\circ$ telles que :

$$f^{-1}(c_1 \Delta c_2) = f^{-1}(c_1) \circ f^{-1}(c_2) = m_1 \circ m_2 \qquad (1)$$

Typiquement, Δ sera une addition ou une multiplication modulaire, mais ce n'est pas toujours le cas. L'utilisons de l'opération arithmétique modulaire est pour gagner plus d'efficacité qui permet de réaliser algébriquement un chiffrement homomorphique. De plus, les axiomes suivants doivent être satisfaits:

Clôture: Pour chaque $m_1, m_2 \in M$ le résultat de l'opération $m_1 \circ m_2 \in M$.
Elément d'identité: Il existe un élément $e \in M$, tel que pour tout élément $m \in M$, l'égalité $m \circ e = e \circ m = m$ est vraie. Un tel e est un élément unique, nous appelons donc e l'élément identité.
Elément inverse: Pour chaque $m_1 \in M$, il existe un élément $m_2 \in M$ tel que : $m_1 \circ m_2 = m_1 \circ m_2 = e$ où e est l'élément neutre.

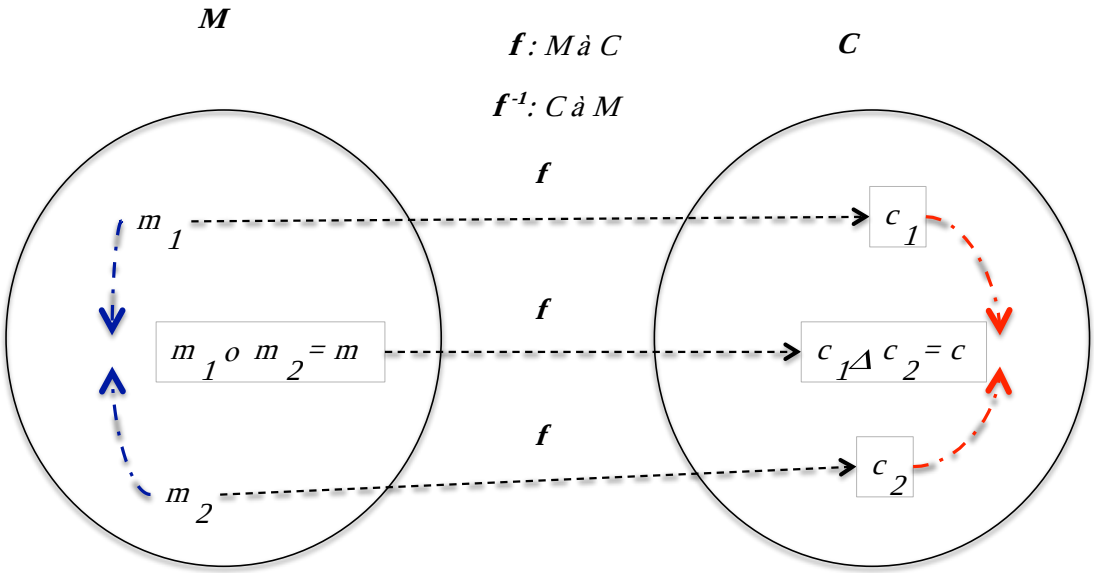


Figure III-3 : Chiffrement homomorphique

La Figure III-3 illustre l'homomorphisme de groupe f: $M \rightarrow C$ tel que (M; C; K; f; f^{-1}) les paramètres de la technique de chiffrement proposée, où M et C sont les pools de messages utilisés et leurs textes chiffrés, K est la clé de chiffrement, f et f^{-1} sont les algorithmes de chiffrement et de déchiffrement exploités. Un chiffrement homomorphique assure que si les textes en clair forment un groupe (M; Δ) et les textes chiffrés forment un groupe (C; $\circ$), alors l'algorithme de chiffrement E doit être une carte du groupe M au groupe C, c'est-à-dire f_k: $M \rightarrow C$, où $k \in K$ est une clé secrète qui vérifie l'égalité suivante:

$$f_k(m_1 \circ m_2) = f_k(m_1) \Delta f_k(m_2) \qquad (2)$$

III.3.1.1 Schéma de chiffrement et déchiffrement

Nous présentons les définitions mathématiques et les concepts importants requis dans la technique proposée. Dans ce travail, notre système de chiffrement conçu est inspiré des fonctions de chiffrement homomorphes proposées par James Dyer et al. dans [172]. Ce chiffrement est basé sur l'opérateur modulaire qui fonctionne sur le groupe cyclique Z_n, où ses séquences arithmétiques ne sont pas ordonnées telles quelles dans Z. Nous définissons donc notre fonction FHE comme indiquée dans l'équation (3).

$$f: c_i = (m_i + rand_i * k) \bmod (k * p) \quad (3)$$

où

$$rand_i = (m_i * p) \bmod k \quad (4)$$

et nous utilisons la notation k, la clé de chiffrement utilisée, car dans notre cas d'utilisation, la même entité qui a chiffré les données le déchiffrera ultérieurement. Cependant, le processus de déchiffrement suit l'équation (5).

$$f^{-1}: m_i = c_i \bmod k \quad (5)$$

Configuration et conditions : Les sous-groupes cycliques, l'élément d'identité, l'inverse et les anneaux constituent les notions de base du schéma conçu pour lequel l'opérateur modulaire fonctionne dans le groupe cyclique Z_n et ses séquences arithmétiques ne sont pas ordonnées comme elles le sont dans Z. Par conséquent, les conditions nécessaires de ce système de chiffrement sont les suivants:

1) Les valeurs choisies de k et p doivent être gardées secrètes et des grands nombres premiers et $p \leq k$, où l'utilisateur seulement fait connaitre la valeur des (k , p);
2) Pour tout m_i et m_j donné, la condition ci-dessous devrait être satisfait:

$$\forall m_i \, \forall m_j, \; m_i + m_j < k \text{ et } m_i * m_j < k$$

Par conséquent, dans ces conditions, nous pouvons chiffrer et déchiffrer à l'aide de modulaire de k. L'homomorphique additif est réalisé en appliquant $f(m_1 + m_2) = f(m_1) + f(m_2)$, et l'homomorphique multiplicatif par $f(m_1 * m_2) = f(m_1) * f(m_2)$, ces propriétés sont démontrées dans la section 3.2. Dans la suite, nous définirons l'algorithmes de la génération de clé, l'algorithme de chiffrement et l'algorithme de déchiffrement.

Génération de clé: Soit λ un paramètre de sécurité. Soit k et p deux grands nombres premiers distincts choisis de manière aléatoire tels que $p \in [2^{\lambda-1}, 2^{\lambda}]$ et $k \in [2^{\eta-1}, 2^{\eta}]$, où $\eta \approx \lambda^2 / \rho - \lambda$ et ρ est l'entropie de distribution. Ici, λ doit être assez grand pour annuler la factorisation directe de $k*p$, et k et p sont choisis pour annuler l'attaque de Coppersmith [173].

Chiffrement: pour chiffrer un message donné $m \in M$, l'utilisateur du Cloud applique la fonction f de l'équation (3) pour obtenir c.

Déchiffrement: pour déchiffrer un message chiffré $c \in C$, l'utilisateur du Cloud applique la fonction f^{-1} de l'équation (5) pour obtenir m, où k est secret.

Algorithme 1 ***Notre_schema***

Génération de clés KeyGen *(λ, ρ)*
Entré: λ, ρ
1 Calculer : $p \in [2^{\lambda-1}, 2^{\lambda}]$ *un grand nombre premier*
2 Calculer: $\eta \approx \lambda^2 / \rho - \lambda$
3 Calculer: $k \in [2^{\eta-1}, 2^{\eta}]$ *un grand nombre premier*
Sortie ***: (****k, p****)***
Chiffrement ***Chiff(m, k, p)***
Entré: m, k, p
1 Calculer: $rand = (m * p) \bmod k$
2 Calculer: $c = (m + rand * k) \bmod (k * p)$
Sortie : c
Déchiffrement ***Déchiff (c, k)***
Entré: c, k
1 Calculer: $m = c \bmod k$
Sortie: m
Calcule index ***CalIndex(m, k, p)***
Entré: m, k, p
1 Calculer: $rand = (m * p) \bmod k$
2 Calculer: $I = (m*k + rand)$
Sortie : I

III.3.1.2 Chiffrement complétement homomorphique

Nous allons maintenant entamer notre discussion sur le concept homomorphique de la technique de chiffrement proposée, ce qui nous amènera immédiatement à transférer ses propriétés et ses avantages vers les exigences du Cloud Computing. Supposons que l'utilisateur du Cloud ait calculé les clés *k* et *p*. Soit *f* une fonction de *M* dans *C*, et m_1, m_2 une paire d'éléments de *M*, si la somme de $m_1 + m_2$ appartient à *M*, on peut appliquer *f* pour obtenir l'élément $f(m_1 + m_2)$ de C. D'autre part, on peut d'abord appliquer *f* pour obtenir les éléments $f(m_1)$ et $f(m_2)$; la somme de ces nouveaux éléments dans *C* est $f(m_1) + f(m_2)$.

Preuve : $f(m_1)+f(m_2) = (m_1+rand_1*k) \bmod k*p + (m_2+ rand_2*k) \bmod k*p$
$= (m_1+m_2+(rand_1+rand_2)*k) \bmod k*p$
$= f(m_1 + m_2)$

Dans la représentation ci-dessus, l'addition est simplement une addition polynomiale sur *C*. De plus, la multiplication en polynôme peut être définie en appliquant $c_1 * c_2 \bmod P$. Supposons que l'utilisateur du Cloud ait calculé la clé secrète (*k, p*). Lorsque m_1, m_2 sont un couple d'éléments de *M*, et que la multiplication de $m_1 * m_2$ appartient à *M*, nous pouvons appliquer la multiplication simple sur f pour obtenir l'élément $f(m_1*m_2)$ de *C*.

Preuve : $f(m_1)*f(m_2) = (m_1+rand_1*k) \bmod k*p * (m_2+ rand_2*k) \bmod k*p$
$= (m_1*m_2+(m_1*rand_2+m_2*rand_1+ rand_1 * rand_2 * k) * k) \bmod k*p$
$= f(m_1 * m_2)$

Remarque: La propriété symétrique de *f*, c'est-à-dire que les ensembles *M* et *C* ont les mêmes éléments, assure l'égalité :

$$((m_1 + rand_1 * k) \bmod k*p \ (+|*)(m_2 + rand_1 * k) \bmod k*p) \bmod k*p =$$
$$(m_1 + rand_1 * k) \bmod k*p \ (+|*)(m_2 + rand_1 * k) \bmod k*p$$

qui est vrai si et seulement si $(m_1 \ (+|*) \ m_2) \in M$ et $m_1, m_2 \in M$.

III.3.1.3 Préservation de l'ordre

Une technique de préservation de l'ordre est une solution d'indexation de données chiffrées. Son rôle est de garantir la possibilité de réalisation des requêtes d'ordre. L'objectif de préservation de l'ordre consiste à conserver les séquences entières des éléments d'origine et à randomiser chaque index dans un intervalle donné, de sorte que les éléments d'origine ne puissent pas être révélés à partir des index. Ce travail gère les requêtes d'ordre sur les bases de données chiffrées de Cloud en incluant un schéma optimisé de chiffrement préservant l'ordre. Dans cette section, nous proposerons un schéma d'indexation efficace préservant l'ordre. Une expression linéaire simple de la forme $a*x + b$ est utilisée. Afin de masquer la valeur des entiers utilisés, le coefficient b est gardé secret (connu de l'utilisateur du Cloud uniquement) et la valeur de $a*x$ sera masquée dans b à l'aide de l'opérateur de l'addition. Les expressions linéaires doivent respecter l'ordre d'indexation, ce qui nous amène à définir la fonction de préservation de l'ordre comme suit:

$$index_i = k * m_i + rand_i \qquad (6)$$

Par sa définition, $rand_i$ est borné par $0 \leq rand_i < k$, ce qui garantit que l'expression linéaire augmente strictement. Par conséquent, $\forall m_1, m_2$ si $m_1 < m_2$, alors $k * m_1 + rand_1 < k*m_2+rand_2$. Ainsi, l'expression linéaire de base respecte l'ordre des *m* utilisés dans *M*. Ce mécanisme d'indexation empêche les attaquants de casser les indices s'ils ne connaissent pas les $rand_i$ de tout m_i donné.

III.3.2 Architecture adaptée à l'approche proposée

Dans ce travail, nous proposons une description d'une architecture de sécurité de haut niveau des services de stockage et de communication de bases de données Cloud. La Figure III-4 montre une représentation schématique de l'architecture proposée. Le cadre a été construit en utilisant deux niveaux. Le premier est le niveau de fournisseur de service de base de données, qui se trouve dans un Cloud public non confiant. Le deuxième est le niveau de client qui se déploie dans l'environnement de la cliente par un proxy client.

Pour interroger la base de données chiffrée, le client dispose d'un proxy qui gère la communication entre la base de données chiffrée et les applications client. Lorsque le client exécute une requête, le proxy la convertit en une requête chiffrée qui s'exécute directement dans le Cloud. Lorsqu'un résultat de requête est repris du Cloud, le proxy le déchiffre avant de le transmettre au client. Le proxy dépend d'un module de métadonnées, qui contient les schémas de base de données et les clés de chiffrement et de déchiffrement.

Plus précisément, avant qu'un tuple soit inséré dans la base de données, le proxy utilise le mécanisme de chiffrement FHE et d'indexation proposés au-dessus pour générer un index et un texte chiffré (IA, CA) pour chaque attribut A du tuple. Le tuple est ensuite stocké dans la base de données chiffrée. Lorsqu'une requête est reçue du client, le proxy analyse sa syntaxe. Pour les paramètres d'une opération de classement, le proxy calcule l'index correspondant à la valeur dans la condition de requête et, pour les paramètres d'une opération homomorphique, le proxy calcule son chiffrement. Ensuite, le proxy envoie la requête transférée au Cloud pour qu'elle soit exécutée dans la base de données chiffrées. Une fois le résultat renvoyé depuis le Cloud, le proxy le déchiffre.

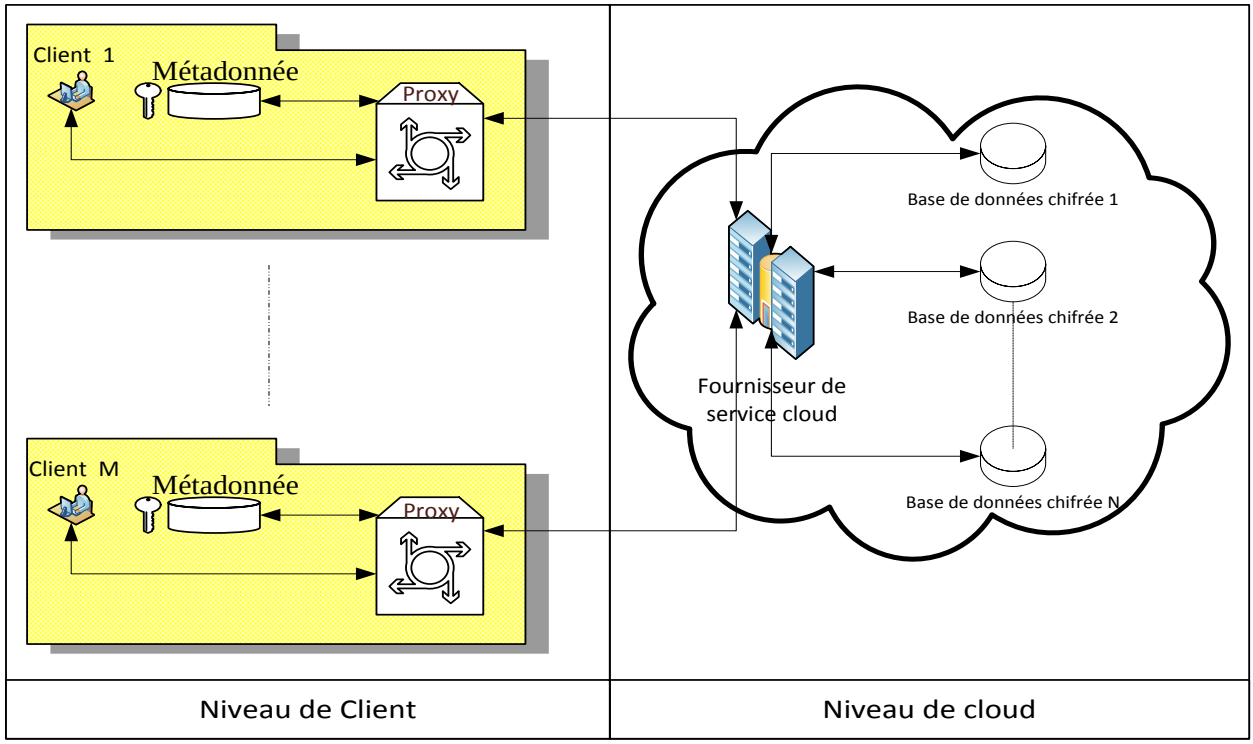


Figure III-4 : L'architecture proposée

III.3.2.1 Data structures

III.3.2.1.1 Structures de métadonnées

Au cours de cette phase, le proxy interroge le client pour créer sa base de données en tant qu'administrateur du système de gestion de base de données (SGBD). Le résultat de cette opération est une métadonnée, où la structure de métadonnée est conçue par le processus de chiffrement de proxy. Le schéma crée (chiffré) est différent de celui utilisé dans la couche client. Les métadonnées contiennent tous les schémas de base de données du client et, pour chaque schéma de base de données, un autre schéma de base de données chiffrée est généré. Les opérations de l'architecture de transformation décrite sont présentées dans l'algorithme 2.

Algorithme 2 ***Génération de schéma de base de données chiffrées***

Entré: schéma de base de données, *k, p*

1 Créer nouveau schéma de base de données
2 NouveauSchéma.name=BaseDeDonnées.ChiffNom(SchémaEntré.name, k, p);
3 **Pour tout** Structure T_i dans SchémaEntré
4 Créer Structure ET_i tel que
5 ET_i.name="T_"+ Structure.ChiffNom (T_i.name, k, p);
6 **Pour tout** Donnée R_j dans T_i
7 Add Donnée ERI_j dans ET_i tel que
8 ERI_j.name="RI_"+ Donnée.ChiffNom (R_j.name, k, p) et
9 ERI_j.val= Donnée.CalIndex(R_j.val, k, p);
10 Add Donnée ERV_j dans ET_i tel que
11 ERVj.name="RV_"+ Donnée.ChiffNom(Rj.name, k, p) et
12 ERVj.val= Donnée.ChiffVal(R_j.val, k, q);
13 **Fin pour**
14 **Fin pour**

Sortie : nouveau schéma de base de données

III.3.2.1.2 Structure de données

Pour créer une Donnée *D* dans une structure *T* dans la base de données Cloud, le proxy chiffre le nom de la structure *T* par le *T_ET*, de sorte que le résultat du nom de la structure soit sans signification pour les administrateurs de Cloud non confiant. *T_* est un préfixe ajouté par le proxy et *ET* est calculé par l'opération *CfiffNom* de structure. Pour la donnée *D* dans *T*, le proxy crée deux données *RI_ER* et *RV_ER*. *RI_* et *RV_* sont des préfixes ajoutés par le proxy, et *ER* est calculé par l'opération *ChiffNom*. La notation *ER* représente le chiffrement du nom *D*. La Figure III-5 montre un exemple d'une structure de données, dans laquelle la structure est générée par le proxy. Les données *RI_ER* et *RV_ER* sont utilisées pour traiter les conditions de requête impliquant des comparaisons d'égalité et d'ordre. Si les conditions de la requête sont satisfaites, la valeur de la donnée *RV_ER* sera renvoyée.

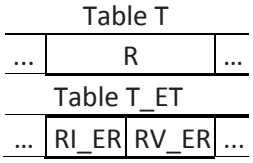


Figure III-5 : Les données d'origine et les données chiffrées

III.4 Implémentation et analyse

III.4.1 Outils de développement

Nous validons l'applicabilité de notre approche dans différentes solutions de Cloud en implémentant et en gérant les opérations de base de données chiffrées sur des systèmes de gestion de base de données Cloud. La version actuelle de notre prototype supporte les bases de données PostgreSQL. Nos tests sont effectuées dans une base de données de Postgres Plus Cloud [174]

avec des limites gratuits, ce qui nécessite une configuration manuelle. Ce fournisseur de Cloud propose des solutions prêtes à l'emploi aux locataires, mais ils ne permettent pas un accès complet au système de base de données. Il propose des interfaces SQL standard et des API propriétaires qui simplifient l'évolutivité et la disponibilité de la base de données. Cela empêche l'installation de logiciels supplémentaires et l'utilisation d'outils et toute personnalisation.

Du côté positif, l'approche proposée utilise des commandes SQL standard qui peuvent chiffrer les données utilisateur sur n'importe quel service de base de données. De plus, certains calculs avancés sur les données chiffrées nécessitent l'installation de bibliothèques personnalisées sur l'infrastructure client. C'est le cas de Postgres Plus Cloud qui fournit un accès par SSL (Secure Socket Layer) pour enrichir la base de données par des fonctionnalités supplémentaires. Le comportement du proxy consiste en une bibliothèque Java et un package de sécurité. La communication entre le client et la base de données en Cloud s'effectue via le proxy, à l'aide du protocole SSL. Cela permet une protection solide contre les tentatives malveillantes de détection de paquets. D'autres composants sont inclus dans notre implémentation système, qui consiste en plusieurs lignes de code Java, avec plusieurs lignes supplémentaires de code de test.

III.4.2 Structure de métadonnées

Pour évaluer les performances de l'approche proposée, nous étudions le scénario suivant: la création d'un Cloud Computing avec un centre de données contenant une base de données SQL. Ensuite, nous créons un groupe de compte utilisateur et pour chacun nous configurons un proxy avec sa métadonnée (Figure III-6).

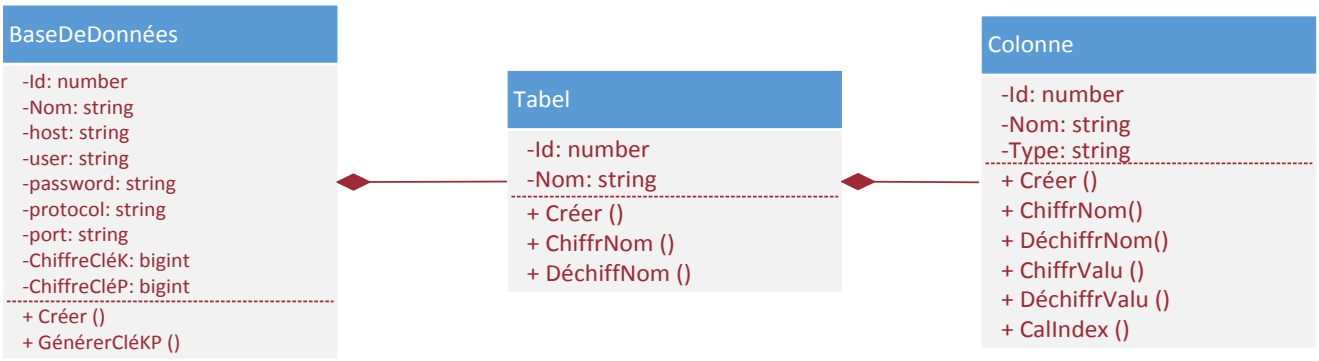


Figure III-6 : Structure de métadonnées

III.4.3 Etude de cas

Nous avons évalué la complexité de la modification d'une application pour qu'elle fonctionne selon notre approche. Nous décrivons le système chiffré de gestion d'employeur CRYPTEMSYS. Dans CRYPTEMSYS nous avons considéré un scénario simple dans lequel nous supposons que la base de données Cloud est accessible uniquement par un client. L'objectif visé ici est de savoir comment mettre en évidence les principales étapes du traitement. Par conséquent, nous ne

tiendrons pas compte des problèmes d'optimisation des performances qui seront abordés par la suite. Le client se connecte à sa base de données Cloud via le proxy. Il crée, insère et interroge sur ses données chiffrées. Pour une meilleure explication, nous avons utilisé une grande table Employer, qui est utilisé pour gérer le système d'information de l'employeur EMSYS. Dans le cas général, la base de données de EMSYS se compose de nombreuses tables. Dans cet exemple, nous allons les réunir dans une table Employer. Elle comporte de nombreuses colonnes parmi lesquelles nous en considérons trois: ID, Nom et Salaire. ID représente le numéro d'identification de l'employé, Nom représente le nom de l'employé et Salary est son salaire. Étant donné que les requêtes au-dessous sont exécutées, le proxy crée la table chiffrée qui est présentée dans la Table III-1 en utilisant l'algorithme 2 telle que Structure dans ce cas est une table et Donnée est une colonne. Cette table représente la base de données du nouveau système CRYPTEMSYS.

Create table Employee (ID int, Name varchar, Salary float,...);

Insert into Employee (ID, Name, Salary) values (1, "Amine", 100000);

Insert into Employee (ID, Name, Salary) values (2, "Ali", 50000);

Insert into Employee (ID, Name, Salary) values (3, "Yahya", 40000);

Dans la Table III-2 au-dessous, nous reprenons les requêtes de base d'après le standard SQL qui sont utilisables dans EMSYS et CRYPTEMSYS.

Table original *: Employer*			**Table chiffrée *: T_16...***					
ID	***Name***	***Salary***	***RI_37…***	***RV_37…***	***RI_68…***	***RV_68…***	***RI_54…***	***RV_54…***
1	*Amine*	*100000*	*112…*	*532…*	*424…*	*419...*	*648…*	*434…*
2	*Ali*	*50000*	*159…*	*337…*	*424…*	*419...*	*324…*	*217…*
3	*Yahya*	*40000*	*207…*	*142…*	*577…*	*104...*	*259…*	*610…*

Table III-1: Table originale et table chiffrée

III.4.4 Génération des requêtes

Le proxy reçoit la requête d'origine envoyée par le client de EMSYS et la modifie selon la fonctionnalité et la structure souhaitées afin d'obtenir une nouvelle requête chiffrée pour l'exécuter dans CRYPTEMSYS. Les opérations de chiffrement et d'indexation de la colonne calculée sont réalisées au niveau du client. Les opérations de transformation de toute instruction sont effectuées de manière flexible en utilisant les algorithmes 2, 3,4 proposés. Dans la suite nous traitons les requêtes de base dans le processus de gestion de base de données.

III.4.4.1 Requête d'insertion

Lorsqu'une requête d'insertion arrive au proxy, il la traite en utilisant l'algorithme 3 pour la transformer en une autre requête d'insertion exécutable dans le système CRYPTEMSYS. Dans ce cas le proxy chiffre les noms de la table et des colonnes de la table puis chiffre les valeurs.

Algorithme 3 ***Génération de requête d'insertion***

Entré: requête, *k, p*

1 Créer nouveau requête
2 Add table ET.nom="T_"+Table.ChiffNom (T.name, k, p)
3 **Pour tout** colonne R_i dans requête
4 Add colonne ERI_i.nom="RI_"+Colonne.ChiffNom(R_i.name, k, p) ;
5 Add colonne ERV_i.nom="RV_"+ Colonne.ChiffNom(R_i.name, k, p);
6 Add value VI_i= Colonne.CalIndex(V_i, k, p);
7 Add value VV_i= Colonne.ChiffVal(V_i, k, p);
8 **Fin pour**

Sortie : nouveau requête

III.4.4.2 Requête simple

Toute requête ne contenant pas de condition, comme illustré ci-dessous, sera considérée comme une requête simple. Dans ce type de requêtes, il n'y a généralement aucune analyse de l'instruction de requête. Un exemple pour ce type de requête est l'instruction suivante:

$$Select \quad R_1, \ ... \ , R_n \quad from \ T$$

La création de nouvelle requête est décrite dans l'algorithme 3 qui est similaire à l'algorithme 2, mais sans chiffrement de valeur.

Algorithme 3 ***Génération de requête simple***

Entré: requête, *k, p*

1 **Si** requête est insertion **alors**
2 ***Génération de requête d'insertion***
3 **Sinon**
4 Créer nouveau requête
5 Add table nom ET.nom="T_"+Table.ChiffNom(T.name, k, p);
6 **Pour tout** colonne R_i dans requête
7 Add colonne nom ERV_i.nom="RV_"+ Column.ChiffNom(R_i.nom, k, p);
8 **Fin pour**
9 **Fin Sinon**

Sortie : nouveau requête

III.4.4.3 Requêtes d'ordre

Pour permettre l'utilisation de requêtes de sélection ou d'ordre, nous supposons qu'une requête du client prend la forme de base suivante:

$$\begin{array}{lll} select & R_1, ..., R_n & \\ from & T_i & \\ where & R_{c1} \ [\ > \ | \ < \ ] \ V_1 & \{ C_1 \} \\ [AND \ | \ OR] & ..... & \\ [AND \ | \ OR] & R_{cm} \ [\ > \ | \ < \ ] \ V_m & \{ C_m \} \end{array}$$

La condition de requêtes est définie par une composition des formules logiques comme $R_i < V_i$ ou $R_i > V_i$, où V_i est une valeur du domaine de la colonne R_i, à l'aide des connecteurs logiques (c'est-à-dire, et, ou). Lors de la traduction de la condition C_i, il suffit de remplacer chaque expression logique par celle traduite. La condition R_{ci} [> | <] V_i est traduit en:

"RI " + Column.ChiffNom(R_i.name, k, p) [> | <] CalIndex(V_i, k, p).

La contrainte "order by R_i" est fréquemment utilisée dans les requêtes. Elle est traduite en "order by *"RI " + Column.ChiffNom(R_i.name, k, p)*".

III.4.4.4 Requête homomorphique

Dans cette classe de requête, la propriété homomorphique est utilisée. La requête généralement contient des opérations arithmétiques ou des agrégations comme SUM ou AVG. Généralement, une requête doit contenir une condition telle que R_i *op* $R_j = V$ (op représente les opérations arithmétiques). La requête du client peut prendre la forme de base suivante:

select R_1, ... , R_n, *SUM*(R_i), ... *from T where* $R_c = V_c$;

Le proxy chiffre les valeurs et les noms des colonnes pour la création de nouvelle requête pour et l'envoyer au Cloud. L'opération de création de requêtes de propriétés homomorphique et d'ordre est décrite dans l'algorithme 4.

Algorithme 4 ***Génération de requête***

Entré: requête, *k, p*

Si requête est simple **alors**
Génération de requête simple
Sinon
  Créer nouveau requête
  Add table nom ET.nom="T_"+Table.ChiffNom(T.name, k, p);
  Pour tout colonne R_i dans requête
    Add colonne nom ERV_i.nom="RV_"+ colonne.ChiffNom(R_i.nom, k, p);
  Fin pour
  Pour tout condition C_i dans requête
    Si C_i dans {> , < } **alors**
      Add "RI_"+colonne.ChiffNom(R_i.nom, k, p) [>|<] colonne.CalIndex(V_i, k, p);
    Sinon
      Si C_i dans {=} **alors**
      Add "RV_"+colonne.ChiffNom(R_i.nom, k, p) [=] colonne.ChiffVal(V_i, k, p);
      Sinon
Add "RV_"+colonne.ChiffNom(R_{i1}.nom, k, p) op "RV_"+colonne.ChiffNom(R_{i2}.name, k, p) [=] colonne. ChiffValu(V_i, k, p);
      Fin Sinon
    Fin Sinon
  Fin pour
Fin Sinon

Sortie : nouveau requête

Cependant, certains calculs ne peuvent pas être adaptés aux données chiffrées. Une analyse approfondie montre quelques cas. Par exemple, la requête de la dernière ligne de la Table III-2 avec la condition (*Salaire * 10 < 100000*) n'est pas adaptée pour le calcul et la comparaison sur la même colonne. Le serveur de Cloud peut traiter cette requête, mais cela nécessiterait également un traitement sur le proxy. Dans le proxy, la requête doit être réécrite en sous-requête qui sélectionne une colonne, Select *RV_54…*Chiffval(10),* calculé à l'aide de la propriété homomorphique. Dans la deuxième étape, rechiffré dans le proxy, une nouvelle colonne *NCol* est créée, qui contient les nouvelles valeurs chiffrées. Enfin, la requête d'origine avec le condition *Where NCol < 100000* doit être exécutée. Les principaux traitements et transformations du proxy sont décrits dans la Table III-2.

Requête original	Requête chiffrée
Create table Employee (ID int, Name varchar, Salary float, . . .);	*Create table T_163... (RI_37… varchar, RV_37... DECIMAL, RI_68... varchar, RV_68... varchar, RI_54... varchar, RV_54... DECIMAL);*
Insert into Employee (ID, Name, Salary) values (1,"Amine",100000);	*Insert into T_163... (RI_37… RV_37..., RI_68..., RV_68..., RI_54..., RV_54...) VALUES ("11...", "53...", "42...", "41...", "64...", "43...");*
Select Name from Employee where ID = 1; (Select with equal)	*Select RV_68... from T_163… WHERE RV_37…="532…";*
Select SUM (Salary) FROM Employee; (Select with Hom)	*Select SUM (RV_54…) from T_163…;*
Update Employee set Name ="Yahya2" where id < 3;	*Update T_163… set RI_68... = "577…", RV_68…= "104…" where RI_37...< "207…";*
Select ID, Name from Employee order by ID; (Select with order);	*Select RV_37…, RV_68… from T_163… order by RI_37…;*
Delete from Employee where ID = 1;	*Delete from T_163… where RV_37…= "53…";*
Select min(Salary) from Employee;	*Select RV_54…, min(RI_54…) from T_163…;*
*Select Name from Employee where Salary*10 < 100000; (Select with Order and Hom)*	*Select RV_37… as ID, RV_54…*"152…" as NCol from T_163…;* *For each ID Select RV_68… from T_163… where ID= RV_37… and CalIndex(NCol, k, p) < "112…";*

Table III-2 : Requête originale et Requête chiffrée

III.4.5 Comparaison et analyse

Pour comprendre l'impact introduite par notre proposition, nous mesurons les temps de réponse du serveur et du proxy et les comparons avec les résultats de CryptDB [162] pour les mêmes types de requêtes SQL. La Table III-3 montre les résultats dans trois colonnes. Nous avons constaté qu'il existe une augmentation globale de la latence des serveurs cloud de 20% avec le proxy proposé, ce qui est considéré comme modeste par rapport à notre objectif d'assurance de la confidentialité des données. Le proxy ajoute en moyenne 0,24 ms à une requête; sur ce temps, 43% sont dépensés dans le proxy tel que 25% sur le chiffrement et le déchiffrement, et les 57% restants

sur l'analyse et le traitement des requêtes hors le proxy. La surcharge cryptographique est relativement faible car notre technique de chiffrement est efficace. L'amélioration de vitesse de calcul de la méthode proposée est remarquable lorsque nous comparons notre proposition avec CryptDB.

Type de quêter	**EMSYS temps de réponse (ms)**	**CRYPTEMSYS temps de réponse (ms)**		**CryptDB temps de réponse (ms)**
		Proxy	**server**	
Create	0.30	0.19	0.34	/
Insert	0.06	0.07	0.07	0.47
Delete	0.05	0.05	0.05	0.36
Update	0.09	0.10	0.10	0.50
Select avec égale	0.08	0.16	0.08	0.97
Select avec ordre	0.14	0.22	0.20	1.00
Select avec Hom	0.09	0.18	0.34	1.45
Select avec ordre et Hom	0.14	0.42	0.64	Not resolved
General	0.08	0.14	0.18	/

Table III-3 : Temps de réponse de EMSYS, CRYPTEMSYS et CryptDB

Dans la Figure III-7, nous évaluons l'adaptabilité de l'exécution de notre approche lorsqu'un client exécute des requêtes SQL sur une base de données chiffrée. Pour évaluer les coûts de chiffrement, nous mesurons le temps de réponse des 50 requêtes à l'aide de différentes clés possibles. Les coûts ou le temps de réponse dépendent du type de la requête, qui est regroupé en fonction de la classe de transaction : Requêtes d'insertion et création, Requêtes select égale, Requêtes d'ordre, Requêtes Hom et Requêtes d'ordre et Hom. Le temps de réponse est indiqué dans l'histogramme de la Figure III-7. Il ressort de cette figure que le temps de chiffrement est inférieur à 0,7 seconde pour toutes les opérations et inférieur à 0,3 seconde pour la majorité des opérations. Ce résultat est important car il confirme que notre approche proposée est une solution valide et pratique pour garantir la sécurité des données avec différents niveaux de sécurité en fonction de la taille de la clé de chiffrement.

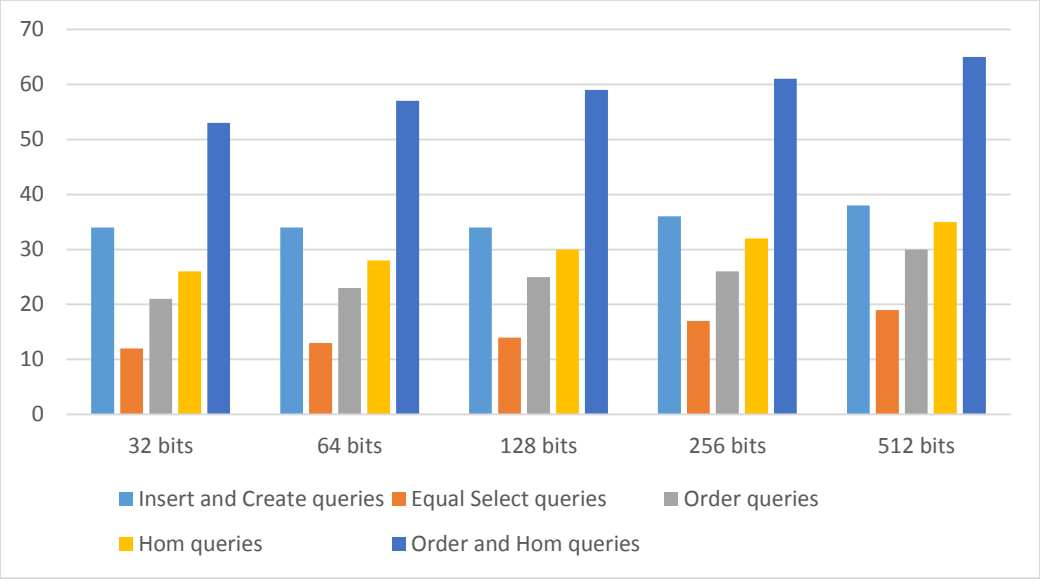


Figure III-7 : Temps de réponse (ms) des requêtes avec clés de 32 bits, 64, 128, 256 et 512 bits

III.5 Conclusion

Dans ce chapitre, nous avons proposé une méthode pour générer un chiffrement complètement homomorphique en assurant la préservation de l'ordre entre les données chiffrées qui sont stockées sur le Cloud, afin de permettre l'exécution des différentes opérations sur ces données protégées. Cette technique est simple à utiliser car elle est basée sur des expressions linéaires et modulaires. Nous avons évalué la complexité de la modification d'une application pour qu'elle fonctionne selon notre approche. Les types de requêtes et d'applications peuvent prendre en charge un niveau de sécurité prédéfini et l'impact sur les performances obtenues à l'aide du calcul linéaire.

Nous avons aussi décrit SQL CRYPTEMSYS en prenant un scénario simple dans lequel nous avons supposé que la base de données Cloud est chez un fournisseur non confiant. Dans la section de l'implémentation, nous avons analysé la technique proposée et validé son applicabilité pour différentes solutions de Cloud. Les tests importants sont orientés pour vérifier sa fonctionnalité dans l'environnement de bases de données en Cloud.

Notre proposition est une technique plus formelle qu'architecturelle pour cela elle est adaptable à la base de données NoSQL qui peut remédier aux limitations des bases de données relationnelles. Il peut répondre aux besoins d'écriture et de lecture à haute fréquence, d'accès et de stockage à haute efficacité, ainsi qu'à une disponibilité et une évolutivité élevées. Comme NoSQL demande un schéma rigide et se trouve généralement dans une structure distribuée et contrairement aux bases de données SQL, les données de la base de données NoSQL n'impose aucune structure de données et il n'est donc pas nécessaire de déclarer un schéma de table avant l'insertion.

Chapitre IV

Contribution 2 : Une approche SMA pour sécuriser les communications et les données dans le Cloud

IV.1 Introduction

Garantir la préservation de la vie privée, la confidentialité et l'intégrité des données deviennent la préoccupation essentielle, en veillant à ce que les données et le calcul soient gardés secrètes et inaltéré [175]. Les services de stockage de données dans le Cloud apportent de nombreux problèmes de conception complexes en raison de la nature structurelle de ces environnements et la perte de contrôle physique. Ces défis ont une influence notable sur la sécurité des données et les performances des systèmes du Cloud.

D'autre côté, la préservation de la confidentialité des données, dans des environnements multi-tenants et à distant, devient plus difficile et conflictuelle, surtout dans la phase de communication ou de transfert de données. Dans [4] nous avons proposé d'utiliser le chiffrement des données, du côté client. Ainsi, le client conserve les clés de déchiffrement, loin de la portée du fournisseur de Cloud. Néanmoins, cette approche suscite plusieurs problèmes de gestion de clés, tels que, le stockage et la disponibilité des clés. En outre, la préservation de la confidentialité devient plus compliquée avec un partage de données dynamique entre un groupe d'utilisateurs.

La protection des données dans le Cloud ne concerne pas seulement la sécurité des données, mais également le processus de transmission de données. Face à ces défis de sécurité et ces faiblesses et vulnérabilités, nous présentons, dans ce chapitre, une approche SMA de sécuriser des données dans le Cloud [3]. L'approche proposée conforme aux hypothèses proposées dans cette étude. C'est une approche basée sur les systèmes multi agents et les techniques de chiffrements pour faciliter l'exactitude de la confidentialité, la disponibilité et l'intégrité de la sécurité de données du Cloud de manière flexible. Elle sépare les opérations d'authentification de communication et de stockage. En plus, pour chaque opération nous appliquons un type de chiffrement différent. Par conséquent, notre proposition permet un déploiement flexible et évolutif de la solution, ainsi qu'une sécurité renforcée pour les données externalisées dans les serveurs de Cloud.

Nous pouvons au départ considérer que les deux paradigmes distribués SMA et Cloud Computing sont incompatibles. Par contre, des analyses détaillées montrent qu'ils sont en réalité non seulement complémentaires, mais ils partagent entre eux une association considérable.

Le Cloud fournit des services à grande flexibilité, de hautes performances et un stockage de données évolutif à un nombre croissant d'utilisateurs. Actuellement, les services Cloud sont principalement utilisés pour gérer des charges de calcul très intensives et pour fournir la facilité de stockage de données de manière évolutif et variable [176]. Or que les systèmes multi agents représentent également un paradigme de calcul et de gestion distribuée sous forme d'un ensemble d'agents ou « systèmes multi agents » qui interagissent entre eux dans un environnement multi locataire pour résoudre un problème commun en utilisant les connaissances et les ressources des agents. Une caractéristique clé des agents est l'intelligence qui peut être utilisée dans les aspects de prise de décision, la distribution des tâches de l'exécution, la communication et surtout l'évolution des agents vers des composants logiciels de plus en plus autonomes capable d'agir sans intervention de l'utilisateur. Ceci qui peut être extrêmement bénéfique en termes de sécurité, sécurité distribuée et prévention d'attaques.

Nous allons présenter le travail réalisé en commençant par un état de l'art décrivant les axes que nous avons fixés pour aborder les aspects communs entre le Cloud et les SMA. Cette partie explicative sera suivie d'un aperçu des architectures et quelques solutions existantes. Nous présenterons ensuite notre proposition concernant la conception du système multi agents, suivie d'une revue de l'implémentation et discutions des résultats des premiers tests du notre proposition.

IV.2 Cloud Computing et Systèmes Multi-Agents

Le Cloud Computing et les systèmes multi agents sont deux modèles de calcul distribués. Il existe plusieurs différences entre les deux modèles mais aussi il existe une grande synergie qui peut être bénéfique dans les deux sens.

D'une part, des recherches dans le domaine du Cloud Computing tendent vers des solutions pratiques pour affranchir les contraintes de l'outil informatique traditionnel à savoir les problèmes de l'espace de stockage, la portabilité, l'élasticité et l'agilité des données. L'objectif donc est d'avoir une haute disponibilité du service avec des coûts réduits. Par conséquent, les environnements Cloud peuvent couvrir les besoins de calcul persistant requis par les systèmes SMA pour différentes applications telles que l'exploration de données, la gestion de services complexes, etc.

D'autre part, généralement, les systèmes multi agents sont utilisés dans la recherche dans les trois domaines suivants: la résolution de problèmes, la simulation de phénomènes complexes et la conception de programmes tout en mettant l'accent sur l'aspect d'intelligence individuelle et collective. Le SMA est donc utilisé pour la conception adaptable des environnements Cloud Computing et par conséquence les systèmes Cloud Computing bénéficient à ce paradigme de nouvelles caractéristiques telles le contrôle distribué, l'apprentissage et l'intelligence, qui

permettent de développer des environnements informatiques beaucoup plus avancés dans tous les domaines (services intelligents, interopérabilité entre les ressources partagées, etc.).

En effet, dans les deux sens les deux paradigmes restent complémentaires. Le nombre d'études que l'on peut trouver sur le Cloud Computing avec la technologie des agents est tout à fait en croissance [177, 178, 179, 180, 181, 182, 183]. Dans ce contexte, nous allons discuter les SMA et la sécurité par SMA. Une étude bibliographique de la technologie est menée et l'intégration des agents dans le domaine de la sécurité est mise en évidence.

IV.3 Agents, agents mobiles et SMAs

Récemment, l'informatique devient de plus en plus distribuée, en particulier les systèmes Clouds. Toutefois, la programmation de telle application distribuée demeure difficile. La plupart de ces applications sont mal adaptées à l'aspect dynamique et à la diversité de tels systèmes. Le paradigme « Agent » semble une alternative intéressante à ces caractéristiques. De nos jours, on ne parle pas sur le mot Agent mais sur les systèmes multi-agents. Les systèmes multi-agents ont pris une place de plus en plus importante en informatique, que ce soit dans le domaine de l'intelligence artificielle, dans les systèmes informatiques distribués et de la robotique et du génie logiciel. Tous ces concepts sont détaillés par la suite.

IV.3.1 Définition d'un agent

Le concept d'agent comme plusieurs autres concepts est défini de plusieurs manières différentes. La définition la plus générale et la plus fréquemment utilisée est celle de Jacques Ferber qui définit " *Un agent est une entité autonome abstraite ou physique qui est capable d'agir sur elle-même et sur son environnement, et qui dans un univers Multi-Agents peut communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances, et des interactions avec les autres agents*" [184].

A partir de cette définition, on comprend que l'agent est une entité physique ou virtuelle qui est capable d'exécute un ensemble de tâches ou de processus de fonctionnement. Ces processus comprennent trois phases successives essentielles qui sont la phase de *perception* qui permet d'élaborer une idée sur l'état actuel de l'environnement, la phase de *délibération* qui permet de décider quelle action à exécuter suivant l'état de l'environnement et l'état interne de l'agent. C'est dans cette dernière phase que le comportement de l'agent est décrit donc c'est la phase la plus importante. Finalement, la phase *d'action* c'est l'exécution de l'action par l'actionneur correspondant.

IV.3.2 Caractéristiques d'un agent

La communauté d'agent décrit plusieurs groupements des caractéristiques de l'agent [185, 186, 187]. En pratique, et dépendamment des domaines considérés, certaines caractéristiques sont

plus importantes que d'autres. Parmi ces formes nous citons les plus importantes dans notre domaine, le Cloud Computing :

- **Autonomie :** C'est la caractéristique la plus intéressante de l'agent dans lequel il est capable d'agir sans l'intervention d'un tiers extérieure et contrôle ses propres actions ainsi que son état interne afin de prendre des décisions. L'agent est dit autonome dans le sens où le concepteur du système ne pilote pas son comportement c'est-à-dire l'agent décide lui-même quelle action à entreprendre parmi celles qui sont possibles.
- **Situation :** L'agent est situé dans un environnement contenant également des entités passives. Il est capable d'agir sur son environnement qu'il peut percevoir grâce à ses entrées sensorielles. L'agent doit s'adapter sans cesse aux changements de son entourage qui pourraient modifier de façon pertinente son comportement à tous les niveaux (objectif, plan, action…etc.).
- **Flexibilité** : L'agent dans ce cas est capable de percevoir son environnement puis réagir et répondre à temps. Il doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au "bon" moment ; donc il est capable de prendre des initiatives afin de s'adapter au changement de son environnement pour atteindre les objectifs qui lui ont été fixé. La réflexibilité signifie dans ce cas: la réactivité, la proactivité et le comportement social.

IV.3.3 Classification des agents

Deux grandes classes peuvent être distinguées : les agents cognitifs et les agents réactifs [188] :

IV.3.3.1 Agents réactifs

L'agent réactif est un agent simple qualifié non intelligent et de plus de bas niveau. Il ne dispose que d'un protocole et d'un langage de communication réduite, il n'a pas une représentation globale de leur environnement et n'est pas capables de tenir compte de leur actions passées. Leur comportement est basé sur le principe de « Stimulus – action », c'est-à-dire, ils répondent d'une manière opportune aux changements de leurs environnements. Le système d'agents réactifs est caractérisé par le nombre important des agents qui sont capables de réaliser des processus complexes.

IV.3.3.2 Agents cognitifs

L'agent cognitif possède une représentation explicite de leur environnement et des autres agents. C'est un agent "intelligent" qui tienne compte de leur passé et dispose d'un but explicite. L'agent possède une base de connaissances qui contient l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec les autres agents et avec son environnement. On dit aussi que l'agent est "intentionnel", c'est-à-dire qu'il possède des buts et des plans explicites leur permettant d'accomplir leur but.

La combinaison des deux classes d'agents donne une nouvelle architecture hybride d'agents qui est un ensemble de modules organisés dans une hiérarchie, chaque module étant soit une composante cognitive, soit une composante réactive. De cette manière, le comportement proactif de l'agent, dirigé par les buts, est combiné avec un comportement réactif afin d'obtenir les avantages des architectures cognitives et réactives, tout en éliminant leurs limitations.

IV.3.4 Système multi-agents

Un système multi-agent (SMA) est un ensemble d'agents mettant en commun leurs compétences et connaissances. Ces agents interagissent et coopère entre eux indirectement (en agissant sur l'environnement) ou bien directement (à travers la communication et la négociation) afin de résoudre un problème commun [188].

Les systèmes multi-agents et les agents autonomes et mobiles représentent une approche pour l'analyse, la conception et l'implantation des systèmes informatiques complexes. La vision basée sur les caractères de SMA qui offre un puissant répertoire d'outils, de techniques, et de métaphores qui y ont le potentiel d'améliorer considérablement les systèmes logiciels [186].

IV.3.4.1 Propriétés des systèmes Multi-Agents

Dans un système Multi-Agent chaque agent possède des capacités à résoudre des problèmes limités ou des informations incomplètes, donc chaque agent à un point de vue partiel. Il n'y a pas de contrôle global du système où les données sont décentralisées et le calcul est asynchrone [189]. Généralement, les SMAs sont conçues pour modéliser des systèmes complexes qui ont des ensembles constitués d'un grand nombre d'entités en interaction direct ou indirect. Ces SMAs sont utilisés de manière performante dans plusieurs domaines; ces performances sont représentées selon les caractéristiques (non exclusives et non exhaustives) suivantes [190]:

- **Distribution spatiale et fonctionnelle :** les agents du système ainsi que leurs fonctionnalités associées peuvent être situées à des endroits séparés.
- **Décentralisation :** les contrôles centralisés permettant de gérer tous les traitements du système ne sont pas souhaité.
- **Hétérogénéité :** les données qui sont traitées et les décisions qui sont prises peuvent concerner différents domaines complètement différents et indépendants les uns des autres.
- **Ouverture et extensibilité :** les agents peuvent s'insérer ou se retirer du système à tout moment en cours de fonctionnement. Ainsi, les agents ont l'indépendance les uns vis à vis des autres et ont la possibilité de modifier leurs comportements.
- **Fiabilité et efficacité :** correspond à la distribution des actions sur les agents en utilisant la capacité de communication entre agents qui peuvent faciliter la résolution des problèmes de façon rapide et efficace.

- **Maintenabilité :** cette caractéristique est due à l'inter-indépendance entre les agents qui permet également de maintenir chaque agent séparément des autres sans affecter le fonctionnement global du système.

IV.3.5 Agents Mobiles

La mobilité des agents est considérée (par quelques chercheurs) comme étant une des caractéristiques des agents. Un agent mobile est un logiciel autonome et auto-adaptable qui est capable de se déplacer avec ses données propres, son code et son état d'exécution au cours de son exécution dans le réseau. L'agent donc a la possibilité de migrer d'un site à un autre et de suspendre ou changer son comportement selon les évènements et les circonstances qu'il rencontre [191].

Généralement, un agent mobile est un paradigme de plus en plus utilisé dans les systèmes distribués. Par exemple lorsqu'un client donne une mission à un SMA, le système active un agent mobile avec la description de sa mission, ce dernier se déplace dans le réseau en accédant localement aux services offerts par les hôtes du réseau et exécute sa mission et récupère éventuellement des résultats. Dans cette proposition nous avons basé sur ce scénario.

IV.3.5.1 Types de mobilité

La mobilité d'un agent mobile peut être classée sur plusieurs modèles. Selon [192] la mobilité soit faible ou forte. La mobilité faible ne permet de transférer avec l'agent que son code et ses données. Par contre dans la mobilité forte, l'agent une fois arrivé à sa destination, il reprend son exécution au point précédent. Une autre classification des agents mobiles dans [193] selon la nature de la tâche qu'ils sont en mesure d'accomplir. Il s'agit d'agent statique, agent visiteur, agent collecteur, agent fusion, agent ordonnanceur et agent identificateur.

IV.3.5.2 Attributs d'un agent mobile

Un agent mobile est constitué par cinq composants ou attributs qui sont : un état, une implémentation (code), une interface, un identifiant et une autorité [194], ces attributs sont transportés par l'agent en se déplaçant à travers le réseau. L'état d'un agent peut être considéré comme une image instantanée de son exécution. Notons que l'agent mobile possède un identifiant unique et un code exécutable. Quand il se déplace à travers le réseau, l'agent fournit une interface qui permet aux autres agents et systèmes d'interagir avec lui. Il peut soit emporter son code soit se déplacer à destination puis voir le code qui est disponible sur la machine distante et récupérer le code manquant à partir du réseau.

IV.3.6 Travaux connexes

La sécurité des données est le défi ultime du Cloud, et les données sensibles devront être protégées au niveau des clients, et non au niveau du fournisseur Cloud. Dans le chapitre précèdent, tous les recherches et les travaux proposés dans ce contexte mentionnent que la sécurité devra passer au niveau des données pour que les entreprises puissent s'assurer que leurs données sont

protégées partout où elles se trouvent. De ce fait, il est intéressant pour les clients de prendre leurs propres mesures de sécurité, indépendamment de ce que les fournisseurs proposent. Ces mesures peuvent être utilisées des méthodes de protection de données à savoir les méthodes de chiffrement, d'authentification, les méthodes de transport sécurisés et les architectures sécurisées basées sur des agents intelligents adaptatifs et capables d'agir en cas d'intrusion.

Dans ce sens, les chercheurs ont proposé de nombreuses travaux utilisant l'approche multi-agents dans un contexte Cloud en vue de dominer les limites de ce dernier telles que la sécurité et la confidentialité des données. Par exemple, dans l'article [195] l'auteur montre que l'utilisation des SMA dans l'environnement Cloud peut bénéficier de hautes performances pour des applications intelligentes et des systèmes complexes. Les SMA ont été aussi utilisés pour proposer un service de gestion d'accès dans le Cloud. Cette recherche prouve ainsi qu'on peut obtenir une infrastructure fiable et évolutive sur laquelle on exécute une application à grande échelle.

Les auteurs de l'article [196] soulignent que les systèmes traditionnels ne sont pas suffisamment efficaces pour réaliser la fonctionnalité du contrôle d'accès dans le Cloud. En raison de la grande extensibilité de l'environnement de Cloud, ils ont utilisé un système multi-agents pour gérer le fonctionnement de leur modèle afin d'améliorer le système de contrôle d'accès.

D'autres utilisations des systèmes multi-agents pour fournir des services de sécurité basées Cloud, ont proposés une architecture SMA dans le contexte de la sécurité du Cloud pour assurer la confidentialité et la disponibilité pour un service de stockage collaboratif [197]. L'architecture proposée consiste en deux couches principales, la première couche est la couche Cloud et l'autre c'est la couche agent. Cette architecture comprend cinq types principaux d'agents. Plus récemment, en 2014, les mêmes auteurs ont discuté la possibilité d'une jointure entre le Cloud et le « MAS-based CBR » et l'article explique comment ceci peut être réalisé [198]. Dans [199], les auteurs ont proposé un framework de sécurité des données en utilisant la flexibilité, la capacité d'interaction et d'apprentissage des systèmes multi-agents. Cette architecture comprend quatre agents, à savoir « l'agent de confidentialité », « l'agent d'exactitude », « l'agent de disponibilité » et « l'agent d'intégrité ». Ce SMA permet d'assurer la sécurité du framework global.

Les auteurs de [200] ont proposé une sécurité à base d'agents du Cloud Computing utilisant la technique de l'obscurcissement. Ils ont traité le problème d'un fournisseur de Cloud Computing malveillant, par le biais d'un agent de protection de données sensibles d'un utilisateur. Ce travail a permis de réduire la surcharge des sites des utilisateurs en utilisant un agent assurant la sécurité par la définition de différents algorithmes permettant de masquer et de récupérer les données. En outre, cette technique d'obfuscation permet d'améliorer l'aspect de complexité des algorithmes utilisés et étend le modèle de sécurité proposé à un autre service de Cloud. Cependant, dans de tels cas, il n'existe pas de base de données pour tous les concepts de cette technique d'obscurcissement et le travail ne montre pas en détail comment intégrer les nouveaux concepts.

Il est à noter que les modèles SMA ont été utilisés pour simplifier et améliorer la gestion et le contrôle de sécurité des architectures des service Cloud. A notre connaissance, aucun service de

sécurité totale du Cloud n'a été traité. La majorité des recherches se sont concentrées sur l'une des méthodes d'authentification, de stockages ou de chiffrements des données dans le Cloud. Notre travail traite de la sécurité des données dans un environnement Cloud sur toutes les phases de l'authentification jusqu'à la réponse au requête ou le stockage dans le Cloud.

IV.4 Approche Proposée

La conception et la structuration d'un système de sécurité et de contrôle sûre dans un environnement distribué nécessite l'utilisation de techniques d'intelligence artificielle pour pouvoir intégrer les tâches permettant une adaptation dynamique aux changements et attaques possibles de données. L'adaptation dynamique aux changements qui surviennent dans l'environnement nécessite les capacités de détection, de réaction et d'apprentissage. Par conséquente, une architecture répartie de composantes, de système et des modèles de raisonnement avancés. Dans ce sens, les SMA permettent d'incorporer des théories, des modèles, des mécanismes, des méthodes et des outils facilitant le développement de systèmes dotés de capacités de réorganisation et pouvant s'adapter automatiquement face aux défis, risques et modifications futures des environnements.

Dans les chapitres précédents, nous avons formalisé le contexte de notre environnement, dans lesquels l'architecture proposée sera exécutée. Compte tenu de la complexité associée à un environnement Cloud Computing, ainsi que des différentes composantes artificielles et humaines impliquées dans ce contexte, il est nécessaire de définir la manière dont les services seront représentés au niveau technique. Pour cette raison, chaque service peut être déployé simultanément sur différents modules virtuels.

Cette étude est basée sur des aspects organisationnels, il est nécessaire d'identifier l'architecture organisationnelle à utiliser. Pour ce faire, la première étape consiste à identifier les composants de l'architecture, ce qui permet d'établir le modèle d'interaction sur la base d'une analyse des besoins en terme des composants potentiels du système. Sur la base de cette analyse (expliquée dans la Figure IV-1), il est possible de déduire les rôles des utilisateurs et les composants participant au système ainsi que la manière dont ils échangeront des informations.

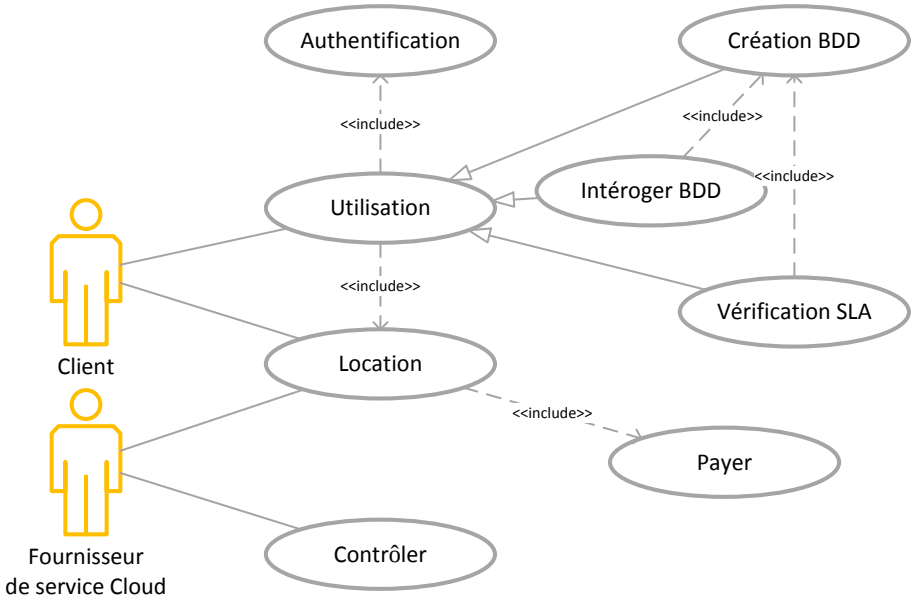


Figure IV-1 : Les besoins fonctionnels du système proposé

Le développement d'un système assurant la sécurité pour la gestion d'environnement Cloud Computing qui est basé sur un modèle SMA diffère des modèles traditionnels basés sur les processus décisionnels centralisé. La portée de cette étude suit un modèle alternatif basé sur la théorie des agents dans laquelle les responsabilités, le contrôle, la surveillance et l'exécution des tâches, sont réparties entre les composants de la plate-forme. Cette architecture permet d'assurer le suivi et de préciser de circulation de données dans le Cloud, ce qui a permis de concevoir des processus de contrôle agiles et sécurisés.

Dans ce travail nous proposons une architecture multi-agents basée sur les techniques de chiffrement et la virtualisation qui sont spécialement conçu pour la gestion des environnements Cloud Computing [3]. Ainsi, nous décrivons l'architecture de sécurité de haut niveau des services de stockage et de communication de données dans le Cloud. La Figure IV-2 montre une représentation schématique de cette architecture. Le cadre est construit en utilisant deux couches, tel que les fonctionnalités de ces couches peuvent être résumées comme suit:

- La couche d'utilisateur: cette couche comporte cinq modules: le module d'interface utilisateur, le module de gestion de clés, le module de chiffrement/déchiffrement, le module d'obfuscation et le module de communication dans le Cloud.
- La couche de fournisseur de services Cloud FSC: la couche FSC peut être identifiée en tant que module de communication dans le coulds, module de gestion de clés, module de chiffrement/déchiffrement et le centre de données.

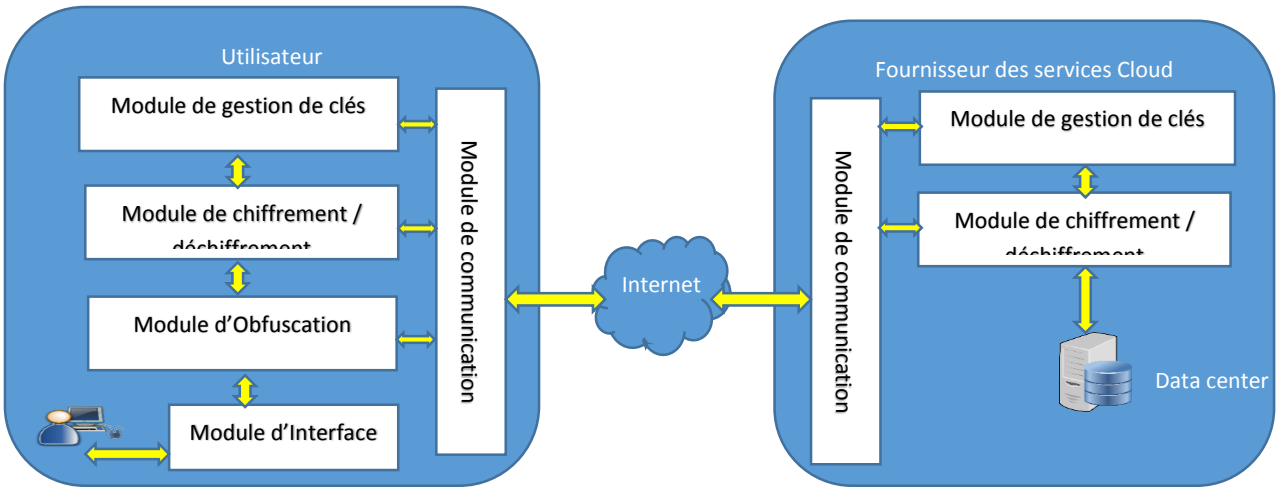


Figure IV-2 : L'architecture de sécurité proposée

IV.4.1 Couche utilisateur

IV.4.1.1 Module d'interface utilisateur

Ce module est situé chez le client. Le but principal de cette unité est d'introduire les opérations du client et d'afficher les résultats. Ce composant présente un avantage en ce qui facilite l'interaction entre les utilisateurs et le système via l'interface graphique. Dans ce cadre, nous assignons un agent pour contrôler les données et normaliser les demandes. Cet agent s'appelé l'Agent d'Interface utilisateur (UIA). UIA agit comme un pont efficace entre l'utilisateur et les autres agents. L'UIA permet donc de :

- Créer un compte d'utilisation le système et de générer une clé de chiffrement de stockage (storage encryption key SEK), puis le donne à l'utilisateur pour l'utiliser à chaque connexion.
- De se connecter au système via son compte (dans le cas d'un compte existant),
- De recevoir des requêtes ou d'afficher des résultats.

IV.4.1.2 Module d'obscurcissement (Obfuscation)

Ce module est aussi situé au niveau client, tel que présenté en Figure IV-3. Il comporte deux agents :

- L'UIA qui fournit l'opération ou la requête sous forme d'un texte en clair.
- L'agent d'obfuscation (ObA) est un agent cognitif, il est le responsable de l'exécution de la technique d'obscurcissement. ObA peut, automatiquement et de manière intelligent, lorsqu'il y a une menace sur l'interface de l'utilisateur, masquer/démasquer une partie ou l'intégralité du texte en clair jusqu'au contrôle de la situation.

L'obfuscation et la récupération des données sont effectués à l'aide du certificat proposé par l'utilisateur. Pour construire ce certificat, nous utilisons une base de concepts d'obfuscation enregistrée dans la couche de Cloud aidant à la proposition des concepts. L'objectif essentiel de cette technique est de sécuriser l'interface utilisateur contre les attaques personnelles à l'interface d'utilisateur.

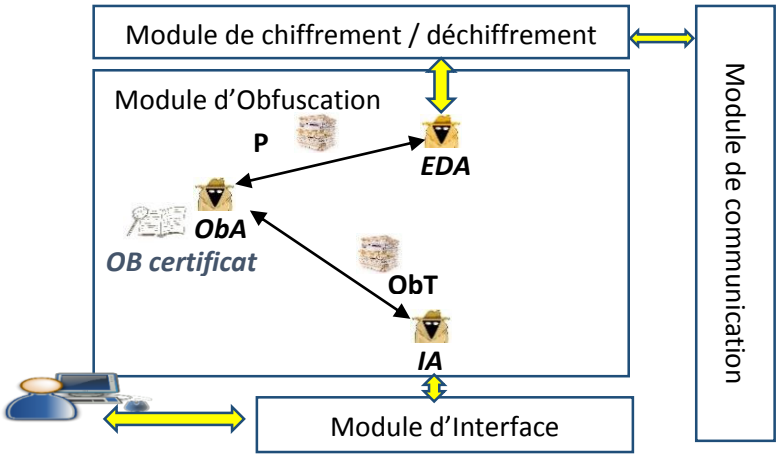


Figure IV-4 : Module d'obscurcissement

IV.4.1.3 Module de gestion de clés

Ce module de la couche d'utilisateur avec l'autre module dans la couche de fournisseur sont responsable de générer les clés de chiffrement utilisées dans l'opération de transport. Les agents de gestion de clés (Key Management Agent KMA) dans les deux couches génèrent les clés nécessaires :

- le KMA de la couche fournisseur génère une clé de chiffrement d'une technique de chiffrement symétrique (Encryption Symmetric Key ESK) qui est utilisé dans le transport de données au Cloud.
- Le KMA de la couche utilisateur génère deux clés d'une technique de chiffrement asymétrique tel que la première est publique (encryption Key Asymmetric PUblic KAsPU) transférée au Cloud par un agent de transport AT, et la deuxième est une clé privée (KAsPr).

Pour accomplir le transfert des clés nous proposons d'utiliser le scénario de transfert illustrées dans la Figure IV-5. Ce scénario est une combinaison entre deux techniques de chiffrement qui sont RSA et Blowfish. La combinaison de ces deux algorithmes est utile en termes d'une méthode hybride, rapide, compacte, moins coûteuse et facile à mettre en œuvre. Le problème principal qui doit être pris en considération est la taille des clés, qui doivent être maintenue très grande, de sorte qu'on ne puisse pas les découvrir par substitution directe des clés. Cela rendra le processus beaucoup plus lent. Par contre, la technique hybride permet d'utiliser une petite clé pour la technique symétrique, parce que la substitution directe ne fonctionnera pas lorsqu'on la combine

avec RSA qui a une grande clé. Par conséquent, la vitesse de traitement est beaucoup plus rapide que le traitement séparé.

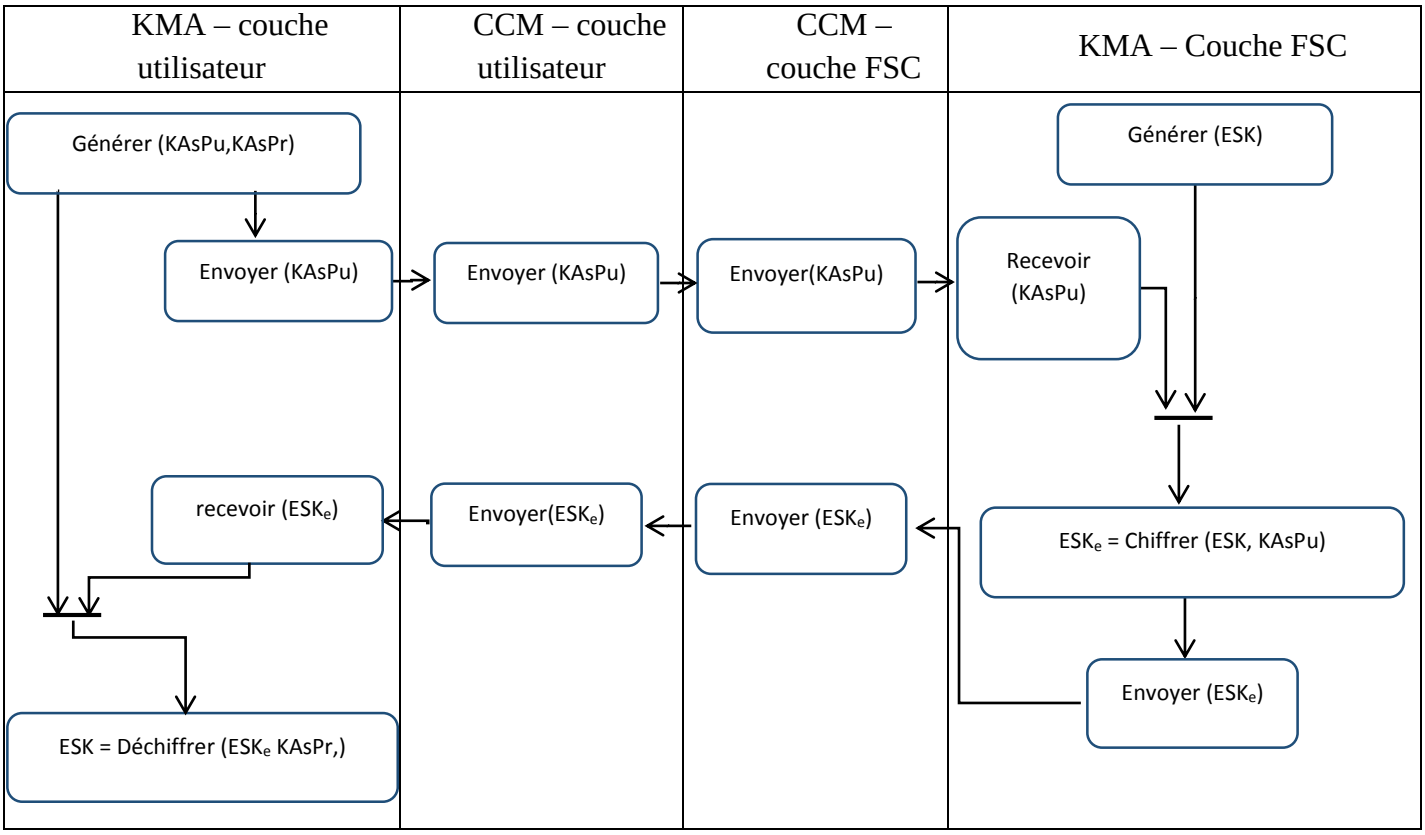


Figure IV-5 : Diagramme d'activités pour la génération de ESK

IV.4.1.4 Module de chiffrement / déchiffrement

Comme illustré dans la Figure IV-6, ce module comporte quatre agents:

- Le KMA qui transfère le ESK.
- Le deuxième agent est l'agent de gestion des clés de chiffrement des données stockées (Storage Key Management Agent). Il gère les opérations de transmission des clés de chiffrement et les métadonnées entre l'utilisateur et l'agent.
- L'agent de chiffrement/déchiffrement (Encrypt/Decrypt Agent EDA) est le responsable de l'opération de chiffrement du texte en clair. L'opération de chiffrement est réalisée en deux étapes. Dans la première étape, EDA analyse la requête et l'adapte en utilisant la métadonnée. Puis le chiffrement de stockage s'effectue selon la technique proposée dans IV.3. Le texte en clair est chiffré par la clé fournie par l'agent SKMA et le résultat est un texte de stockage chiffré (Stored Ciphertext SCit). Dans la deuxième étape, l'algorithme de chiffrement Blowfish est utilisé pour chiffrer la sortie de la première étape SCit par la clé ESK pour obtenir un texte de transport chiffré (TCit).

- Le quatrième est l'agent mobile de transport (TA), il transfère le TCit au module de communication de Cloud.

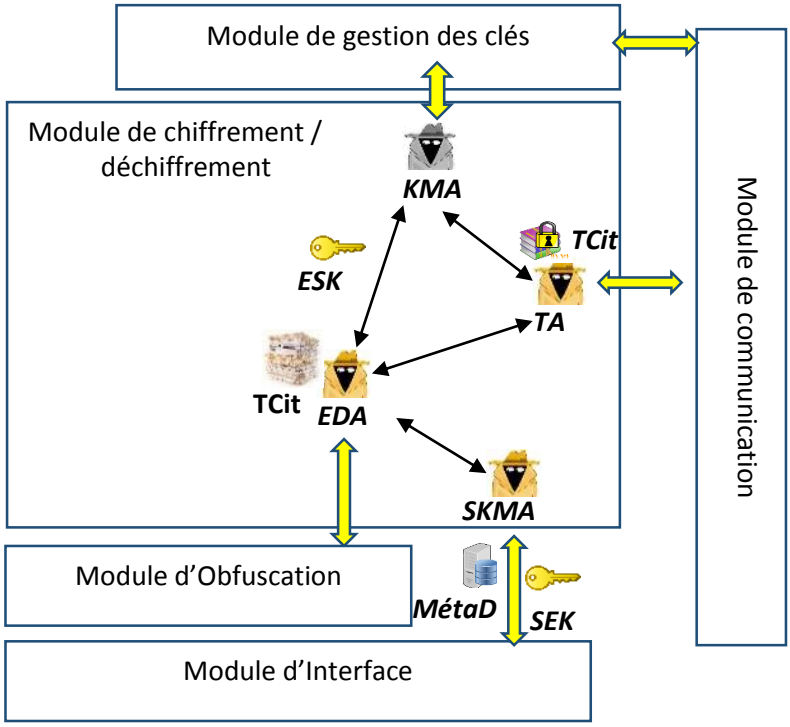


Figure IV-6 : Module de chiffrement / déchiffrement

IV.4.1.5 Module de communication de Cloud (MCC)

Il existe deux modules de communication MCC, le premier dans le site de l'utilisateur et l'autre dans le site de fournisseur de Cloud. Leur rôle est de d'organiser et réaliser tous les opérations de transfert des données. L'agent mobile de transport TA est responsable de tout échange entre les deux couches concernant le transfert des clés, les données à stocker, les concepts d'obfuscation...

IV.4.2 Couche de fournisseur de services Cloud

IV.4.2.1 Module de gestion de clés

Ce module est pareil au celui de couche utilisateur, il possède un agent KMA qui a aussi les mêmes activités de KMA de la couche utilisateur. Cependant, la différence réside dans le fait que si l'utilisateur souhaite envoyer le texte en clair, la couche FSC est responsable de l'envoi de la ESK, sinon c'est l'inverse.

IV.4.2.2 Module de chiffrement / déchiffrement

Ce module (voir Figure IV-7) comporte quatre agents: l'agent KMA qui donne le KSE. Cet agent est généré précédemment dans cette couche. L'agent de chiffrement / déchiffrement déchiffre

le texte chiffré transporté donné par l'agent de TA avec la clé KSE où SCit = Déchiffrer (TCiT, KSE). Ce texte chiffré stocké doit être transféré au quatrième agent, c'est-à-dire l'agent de stockage (SA).

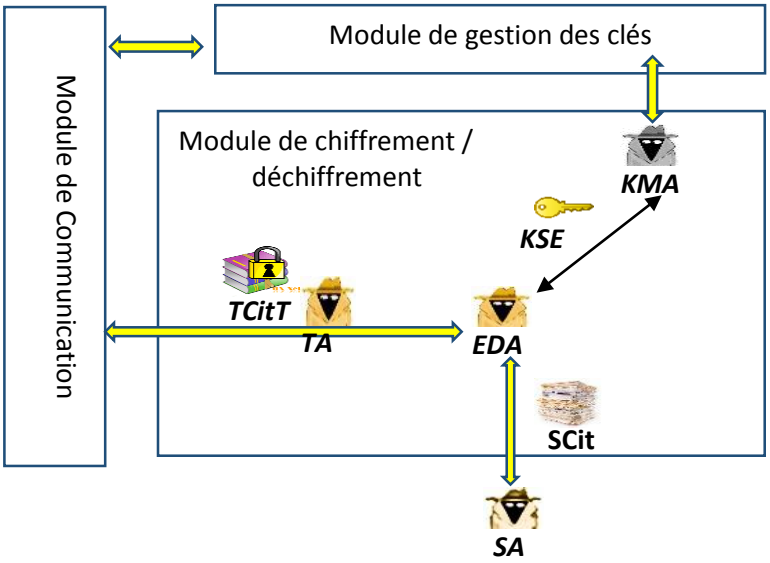


Figure IV-7 : Module de chiffrement / déchiffrement de la couche FSC

IV.4.2.3 Centres de données

L'agent de stockage (SA) exécute la requête chiffrée dans la base de données qui est envoyée par l'utilisateur. En cas de résultat, l'agent SA envoie ie résultat chiffré à l'EDA et le même scénario de transmission se répète mais de l'autre côté.

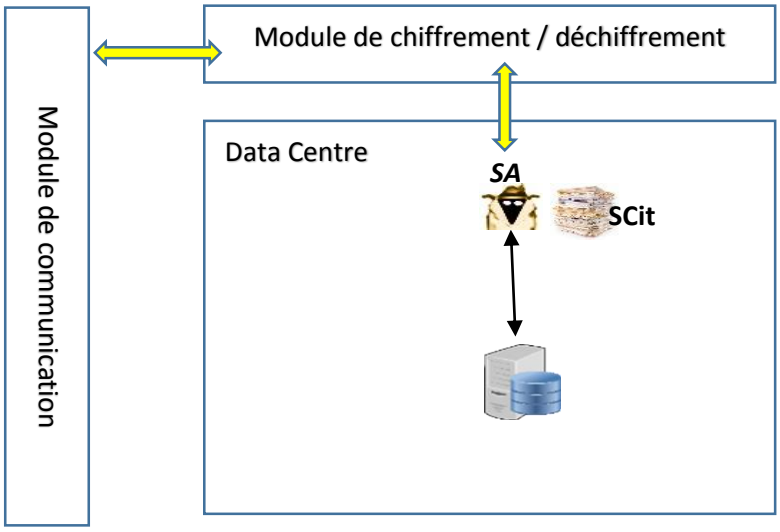


Figure IV-8 : Architecture du centre de données

IV.4.3 Avantages de l'intégration des SMA dans la sécurité du Cloud

- Plus d'intelligence: les caractères d'autonomie et l'adaptabilité automatique des agents donnent au système plus d'intelligence dans l'opération de contrôle de sécurité. Cette caractéristique sera plus efficace si le système est renforcé par un module de détection d'intrusions. L'architecture Cloud est généralement basée sur la stratégie de la virtualisation. L'architecture proposée dans le cadre de l'étude [4] repose sur la virtualisation d'agents intelligents, ce qui permet de fournir une nouvelle solution requise par les plates-formes Cloud pour permettre aux composants de s'adapter, changer, entrer ou sortir. L'objectif principal est la capacité de sécuriser et de contrôler un environnement Cloud.

- Évolutivité infinie: avec les agents, la communication dans le Cloud est optimale, car à chaque fois que la requête d'utilisateur nécessite plus de communication avec le Cloud, le système crée tous les types d'agent pour affecter une communication correcte.

- L'optimisation et la réduction de la consommation de bande passante: l'un des problèmes les plus importants de la technologie de l'informatique en Cloud est la nécessité d'une grande bande passante, qui sera renforcée par l'utilisation d'agents de transport dans la communication entre les différentes couches de Cloud Computing.

- Tolérance aux pannes et robustesse: les agents ont la capacité de réagir de manière dynamique aux situations défavorables. L'objectif est de construire un système distribué robuste et tolérant aux pannes. C'est le cas des agents de transport. Si l'un des modules de Cloud Computing est sur le point d'être arrêté, les agents de transport reprend la responsabilité de continuer l'exécution des tâches nécessaires dans cette région et dans l'autre région.

- Adapté aux réseaux hétérogènes: les agents mobiles étant basés sur JAVA, il sera pratique de les utiliser dans les réseaux basés sur des systèmes matériels et logiciels hétérogènes.

IV.5 Implémentation de la solution

Dans le processus de l'implémentation d'une solution vient après un enchainement de plusieurs étapes et son but principal est de réaliser un système capable de résoudre les problèmes posés en utilisant des outils et des algorithmes. Dans le but de réaliser et valider les idées proposées dans ce chapitre, la suite montre les outils et la configuration utilisée afin de développer ce système.

IV.5.1 Outils et Plateformes Utilisées

Afin de faciliter le développement de notre système, nous avons utilisé un ordinateur de processeur Core I5 avec une RAM de 8 Go sous Windows 7, mais on peut implémenter ce système sur n'importe quel système d'exploitation grâce à la machine virtuelle du Java. Pour développer

les agents, nous avons utilisés la plateforme JADE et l'environnement de développement intégré Eclipse. Ce dernier fournit un langage de communication entre les agents en utilisant le FIPA-ACL. Enfin, nous utilisons la plate-forme Cloudsim pour simuler les ressources virtuelles ainsi que les tâches réalisées dans notre système.

IV.5.1.1 Plateforme JADE

JADE (Java Agent Development framework) est une plateforme implémentée avec le langage JAVA par le laboratoire TILAB [201]. Elle est destinée aux développeurs qui s'intéressent aux systèmes multi-agents distribués. La plateforme JADE fournit une interopérabilité sans limite pour les applications quelles prend en charge. Cette plateforme est indépendante du système d'exploitation ainsi que du matériel sur laquelle elle est implémentée [202]. Cette plateforme permet le développement des applications conformes aux normes FIPA qui propose le modèle AMRM (Agent Management Reference Model) comme le modèle de base de l'architecture de la plateforme JADE. Les principaux modules qui composent l'architecture JADE sont : DF, AMS et également le MTS (Massage Transport Service), un moyen pour la communication entre plusieurs plateformes JADE [203].

Nous avons développé notre système à l'aide de JADE-S, la version 2 de JADE. JADE-S est formée par la combinaison de la version standard de JADE et du plug-in de sécurité [204]. JADE-S inclut des fonctionnalités de sécurité telles que l'authentification des utilisateurs/agents, l'autorisation et la communication sécurisée entre agents sur la même plateforme tels que [205]:

- Autorisations et stratégies: l'autorisation est une caractéristique qui décrit la possibilité d'effectuer une action par un agent sur une ressource donnée. La stratégie spécifie les autorisations disponibles pour différentes entités.
- Certificats et autorité de certification: l'autorité de certification (CA) est l'entité qui signe tous les certificats pour la plateforme, à l'aide d'une paire de clés publique / privée.
- Délégation: ce mécanisme permet de donner des autorisations à un agent. Outre le certificat d'identité, un agent peut également posséder d'autres certificats qui lui sont donnés par d'autres agents;
- Communication sécurisée: la communication entre les agents de différents conteneurs / hôtes est réalisée à l'aide du protocole SSL (Secure Socket Layer). Cela permet une protection solide contre les tentatives malveillantes de détection de paquets.

IV.5.1.2 Simulateur *Cloudsim*

A cause de frais élevés et de limitation des tests sur un environnement Cloud réel comme Amazon EC2, nous avons utilisé des simulations pour l'évaluation de notre approche. Un simulateur Cloud est un outil qui permet aux développeurs d'utiliser des ressources et des machines virtuelles puissantes. Pour valider les idées présentées dans ce chapitre, nous avons

utilisé le simulateur Cloudsim qui est l'un des principaux représentants des simulateurs de Clouds. Cloudsim est un outil qui offre aux développeurs un moyen pour modéliser, simuler et réaliser des expérimentations dans le but de créer une infrastructure Cloud. Les principales fonctionnalités de Cloudsim sont détaillées dans [206, 207] *:*

- Supporter la modélisation et la simulation des Datacenter Cloud à grande échelle.
- Supporter la modélisation et la simulation des serveurs virtuels d'hébergement d'application, avec la possibilité de personnaliser la politique de fourniture des ressources d'hébergeme nts aux machines virtuelles.
- Supporter la modélisation et la simulation de ressources virtuelles et logiques pour des tests et assurer l'optimisation de la consommation d'énergie sur le Cloud.
- Supporter la modélisation et la simulation des différentes topologies réseaux, centres de données et l'échange de messages entre applications Cloud.
- Supporter la modélisation et la simulation des systèmes Cloud fédérés.
- Supporter l'insertion dynamique de nouveaux éléments de simulation, avec des capacités d'arrêt et de reprise de simulation.
- Supporter tout type et politiques d'allocation de ressources pour la création de machines virtuelles, et tout type et politiques d'allocation de ressources pour l'hébergement d'application et de services Cloud.

IV.5.2 Présentation du prototype

Nous avons créé une interface qui facilite à l'utilisateur l'accès et la manipulation des ressources Cloud car le Cloudsim n'offre pas une interface graphique qui permet l'affichage graphique des résultats et le déroulement visuel d'une simulation. Le lancement de notre application débute par l'interface de l'utilisateur lorsqu' il demande de louer un service de stockage Cloud. L'interface de l'utilisateur permet à l'utilisateur de créer une base de données et d'introduire ces propres paramètres. La structure de prototype et de la simulation implémentée est présentée dans la Figure IV-9. Principalement, notre implémentation se déroule en 2 étapes :

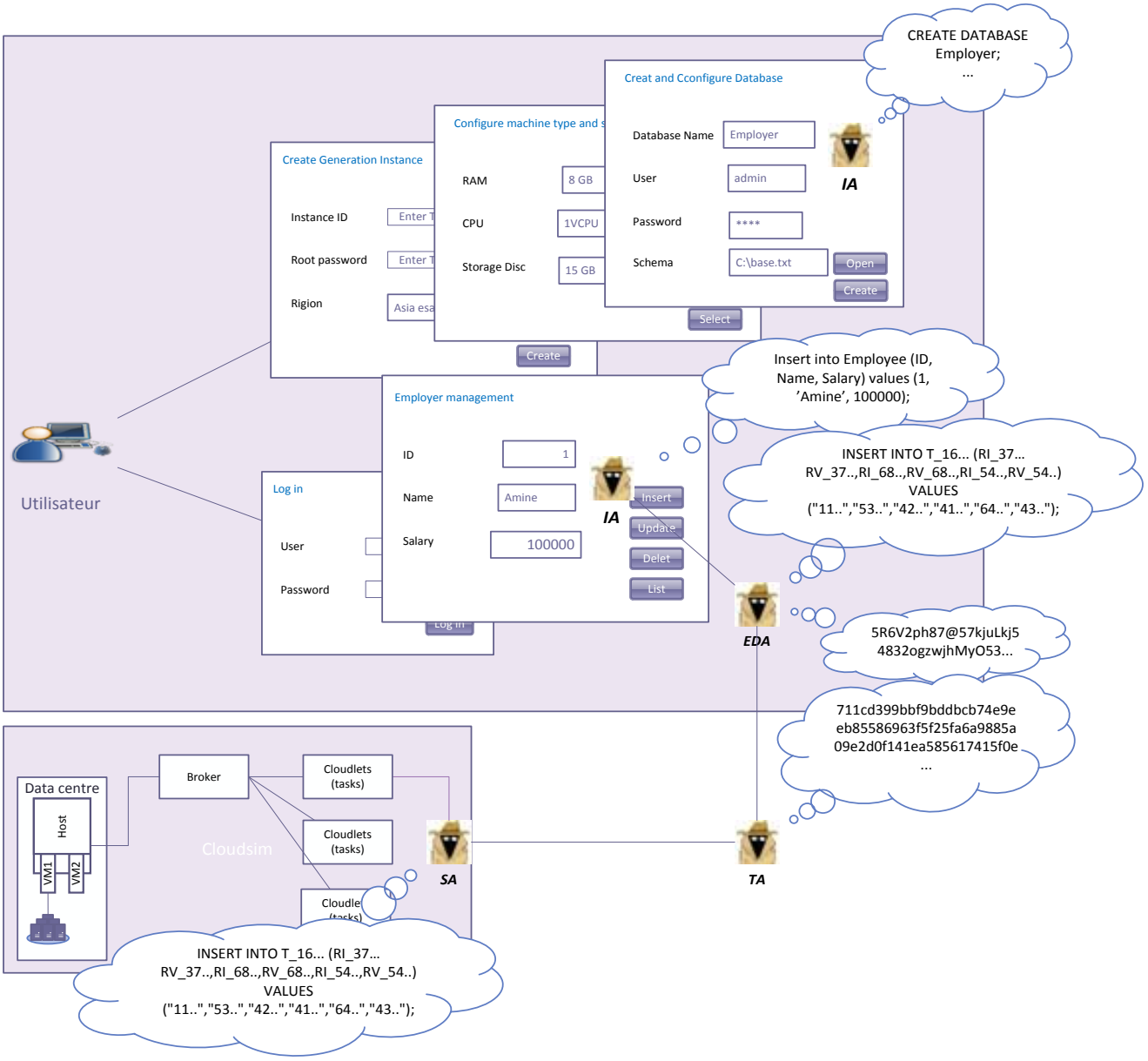


Figure IV-9 : La simulation de l'implémentation

1) **Création et paramétrage d'une simulation :** c'est la configuration des paramètres globaux de la simulation et la configuration de l'environnement de Cloud Computing. Nous validons l'applicabilité de notre approche dans différentes solutions de Cloud en implémentant et en gérant des opérations de base de données cryptées sur des infrastructures de Cloud émulées. La version actuelle de notre prototype prend en charge les bases de données relationnelles PostgreSQL, MySql et SQL Server, ainsi que la base de données NoSQL de MongoDB. L'importances de ces tests visent à vérifier la fonctionnalité de nos travaux sur différents types de bases de données en Cloud. Ces types de bases de données offrent des normes des interfaces SQL et JSON qui simplifient l'évolutivité et la disponibilité de la base de données en Cloud. Du côté positif, notre proposition utilise uniquement des commandes des standards SQL et

notations JSON pour chiffrer les données utilisateur sur n'importe quel service de base de données en Cloud.

Dans un premier temps l'utilisateur crée une instance de data centre après avoir choisir le type de base de données, dans notre cas c'est une base de données MySQL. Puis il précise ses paramètres et donne son schéma. L'agent EDA, lorsqu'il obtient le code de création, il génère un autre schéma du Cloud selon l'algorithme 4 et un méta data pour l'envoyer à l'utilisateur. Dans cet état la base de données du Cloud devient prêt à exploiter.

2) **L'exploitation :** cette étape représente toute la durée vie de la base de données après la création. A chaque connexion d'un utilisateur à la base de données, le système crée l'ensemble des agents et effectue la configuration du SMA. L'application charge automatiquement la plateforme JADE dès son lancement. Nous avons programmé notre modèle JADE sur le diagramme de classes représenté dans la Figure IV-10. Le comportement de l'agent principal EDA consiste en la bibliothèque et le package de sécurité Java. Ce package comporte un analyseur de requête, un chiffreur / réécriture de requête, qui chiffre les données ou les champs de la requête avec tous les algorithmes de chiffrements. L'agent EDA est considéré comme un proxy dans notre architecture, comme le proxy de travail [2].

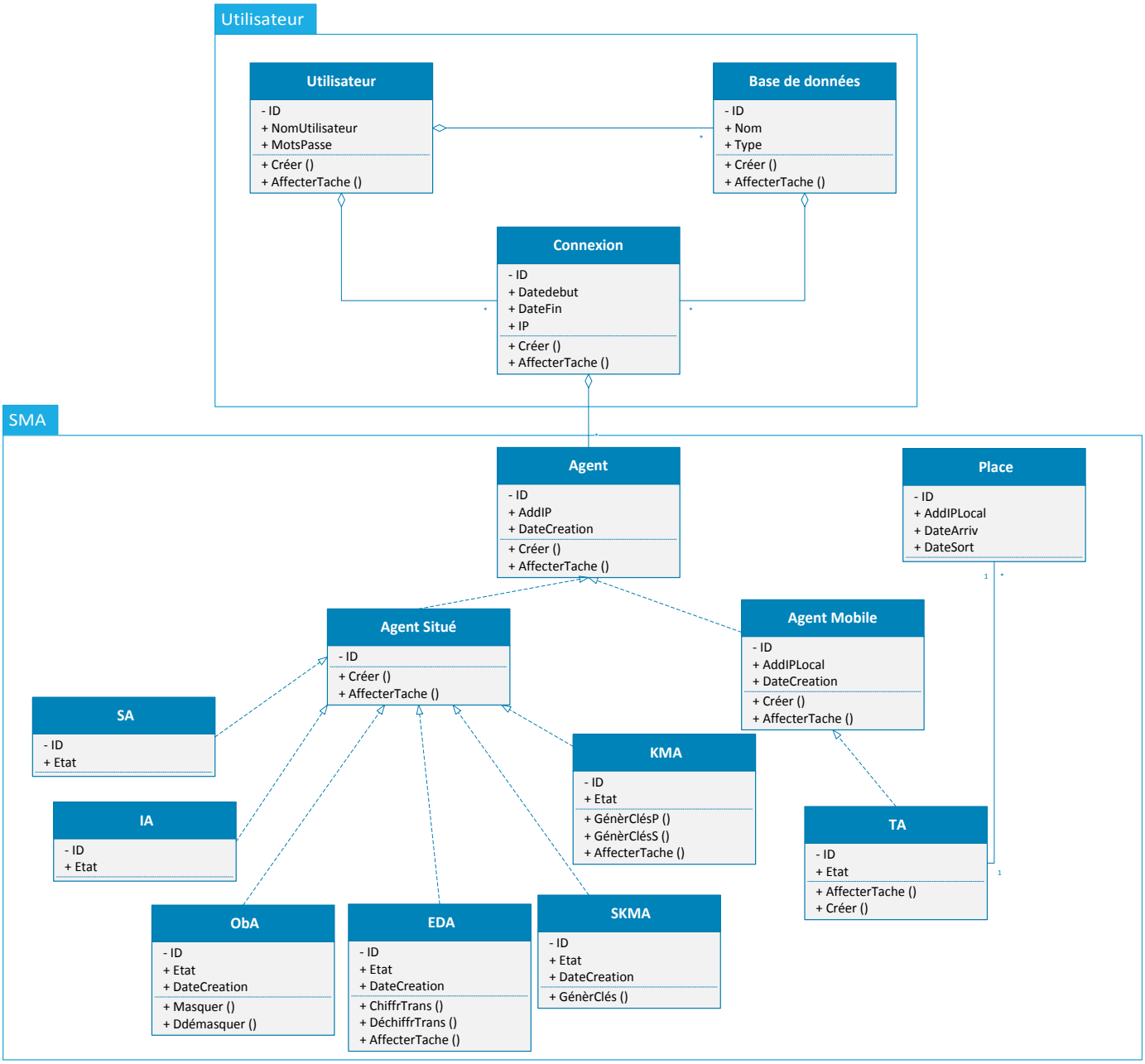


Figure IV-10 : Diagramme de classes de l'implémentation

IV.5.3 Résultats obtenus et discussions

L'utilisateur est connecté au simulateur Cloud via une interface utilisateur contrôlée par les agents. Il crée, met à jour, enquête ou supprime ses données cryptées dans son centre de données. Nos expérimentations évaluent la surcharge du chiffrement et comparent les temps de réponse des opérations effectués sur la base de données chiffrée. Les résultats expérimentaux montrent les performances des algorithmes de chiffrement Blowfish, AES et RSA qui sont utilisés dans le

module de communication. La Table IV-1 montre la comparaison de ces trois algorithmes avec notre algorithme hybride qui sont implémentées sous les mêmes conditions matériel et logiciel. La comparaison montre que l'algorithme proposée est beaucoup plus rapide que les autres algorithmes. Le meilleur temps est obtenu car la clé de Blowfish est très petite, et la clé de RSA, qui est plus puissante, assure un niveau de sécurité optimal.

Algorithme	Taille de paquet	Temps de chiffrement (ms)	Temps de déchiffrement (ms)
Blowfish	100	1.145	1
AES		1.625	1.148
RSA		2.322	2.294
Méthode Proposée		0.815	0.814
Blowfish	500	4.801	3.715
AES		5.347	4.639
RSA		7.712	7.400
Méthode Proposée		1.938	1.911
Blowfish	1000	7.633	7.170
AES		9.392	9.990
RSA		15.381	15.046
Méthode Proposée		2.253	2.032
Blowfish	5000	42.100	38.812
AES		52.460	51.509
RSA		94.166	87.663
Méthode Proposée		8.089	7.690

Table IV-1: Comparaisons des algorithmes de chiffrement

La Figure IV-11 montre les résultats mesurant les temps de réponse de notre solution de SMA et les compare avec les résultats de travail proposé dans le chapitre précédent pour les mêmes types de requêtes SQL. Dans cet ensemble d'expériences, nous évaluons la surcharge introduite lorsqu'un utilisateur exécute des requêtes SQL sur la base de données chiffrée. Ces temps de réponse dépendent de types de requêtes qui sont indiqués dans l'histogramme suivant. Cette figure indique que le temps de réponse est diminué par 5 milliseconde pour la majorité des requêtes. Ce résultat est important car il confirme que notre approche proposée est une solution valide et pratique pour garantir la sécurité des données dans une base de données en Cloud.

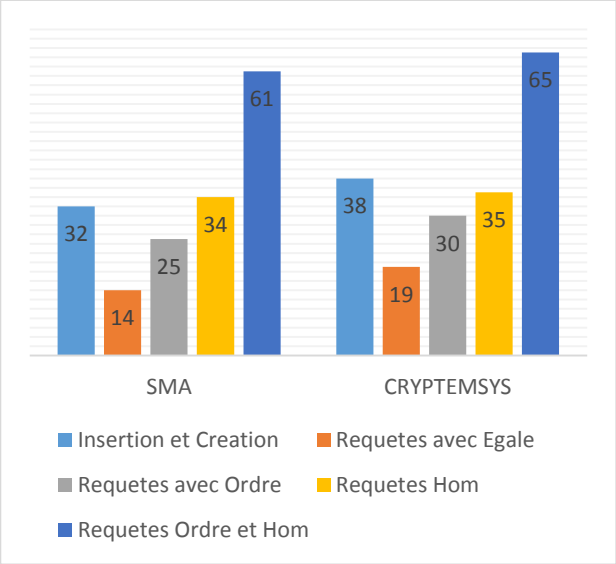


Figure IV-11 : Durée de traitement de la requête (ms) avec clés de chiffrement de 512 bits

IV.6 Conclusion

La sécurité des données est un problème crucial dans l'environnement de cloud computing. Pour protéger les données stockées en cloud, de nombreuses technologies sont proposées en utilisant les techniques de chiffrements et les systèmes multi-agents. Dans ce chapitre, nous avons présenté une approche basée sur les systèmes multi-agent et les techniques de chiffrements afin de fournir une architecture de sécurité multi-niveaux solide et pratique. Le premier niveau est l'interface utilisateur. Il montre la technique de l'obscurcissement pour de garantir la confidentialité des données existantes dans cette interface.

Le deuxième niveau vise à assurer les opérations de communication, qui combinent les algorithmes de chiffrements symétriques et asymétriques. Ces méthodes combinées prouvent que la nouvelle technique proposée garantissant les avantages de l'algorithme symétrique en termes de temps de traitement et garantissant la robustesse de l'algorithme asymétrique en longueur de clé. En fait, cette nouvelle approche est plus rapide que d'autres techniques cryptographiques en termes de traitement de données. De plus, le chiffrement asymétrique est un moyen robuste et sécurisé pour le mécanisme de distribution de clé.

Le dernier niveau est conçu pour assurer les opérations de stockage. Pour cela, nous avons manipulé les données de l'utilisateur en utilisant une méthode de chiffrement complètement homomorphique. En se basant sur la réalisation d'un système multi-agents, l'intégrité et la souplesse inter-agents peuvent être étendus à une structure multi-proxy pour la réalisation des autres systèmes plus complexes tel que les systèmes gérant les Big Data.

Conclusion générale

Les efforts récents visant le développement des technologies Cloud en utilisant des nouvelles techniques, méthodes, politiques et mécanismes efficaces de sécurisation. Cette thèse s'est intéressée à ce challenge du Cloud Computing. En particulier, le travail de cette thèse s'intègre dans le contexte du problème de la sécurité des données. Notre objectif principal était de proposer une approche pour la sécurité de données dans le Cloud Computing. Le but ultime est de satisfaire les besoins des utilisateurs Cloud, tout en proposant une solution pour la sécurisation des données la plus sûre et la plus confiée. La solution proposée doit garantir en premier lieu la confidentialité et l'intégrité.

Dans l'introduction générale et les deux premiers chapitres, nous sommes focalisés sur le concept de la sécurité de Cloud Computing. Nous avons présenté la définition et l'objectif général de notre contexte d'étude. Ensuite, nous avons présenté la définition, les caractéristiques et les principaux défis et challenges de recherche dans le cadre du Cloud. Après, nous avons détaillé les principales exigences, normes, stratégies de la gestion de la sécurité du Cloud Computing. Nous avons discuté aussi les problèmes de sécurité, de confidentialité, de contrôle des données et les menaces de la sécurité. En effet, cette partie représente l'état de l'art de ce travail de thèse.

Dans le troisième chapitre, nous avons présenté notre première contribution. Il met l'accent sur la problématique de la confidentialité des données stockées dans le Cloud. Nous avons mis en place un modèle garantissant la confidentialité des données. Nous avons proposé aussi une technique pour générer un chiffrement complètement homomorphique en assurant la préservation de l'ordre entre les données chiffrées stockées sur le Cloud. L'objectif est de permettre l'exécution des différentes opérations sur ces données protégées. Cette technique est simple à utiliser car elle est basée sur des expressions linéaires et modulaires. Nous avons validé cette solution par un scénario simple décrit par SQL CRYPTEMSYS.

Dans le dernier chapitre (le quatrième chapitre), nous avons introduit notre deuxième contribution en utilisant des systèmes multi-agentes. Plus précisément, Nous avons présenté une approche basée sur les systèmes multi-agent et les techniques de chiffrements. Cette approche a fourni une architecture de sécurité multi-niveaux. Nous avons considéré le niveau interface utilisateur et le niveau supports de communication. Le système proposé utilise une approche de chiffrement hybride rapide et robustesse. En fin, dans le dernier niveau du data centre, nous avons manipulé les données de l'utilisateur en utilisant une méthode de chiffrement complètement homomorphique.

Pour conclure, cette thèse nous a permis d'examiner un large éventail de concepts, de modèles et de technologies dans les domaines de la sécurité des données et de Cloud. Notre objectif était d'étudier les problèmes de sécurité des données stockées dans le Cloud, tout en se concentrant sur la confidentialité des données et l'intégrité des données à distance. Nous avons fourni des nouvelles architectures basées sur les techniques de chiffrements pour répondre à nos objectifs. Nous avons montré aussi que le travail proposé rejoint une thématique de recherche riche et encourageante.

Cette thèse constitue une base de travail à partir du quelle, de nouvelles activités de recherche peuvent être lancées afin d'améliorer le travail présenté. Les perspectives que nous proposons peuvent donc s'orienter vers les directions suivantes:

- Adapter nos propositions à la base de données NOSQL. Contrairement aux bases de données SQL, le NOSQL demande un schéma peu structuré et se trouve généralement dans un environnement distribué. En effet, les données de la base de données NOSQL n'impose pas de déclarer un schéma de table avant l'insertion.
- La réalisation d'un système multi-proxy pour la sécurité de Big Data.
- Nous envisageons le développement d'un système SMA pour la détection d'intrusions dédié et spécifique au Cloud. L'objectif est de bénéficier de caractéristiques de l'autonomie, l'adaptabilité automatique et par conséquence plus d'intelligence dans l'opération de contrôle de sécurité.
- La tendance forte des appareils mobiles et des réseaux de capteurs faire appel aux services de gestion de données en Cloud. Il sera intéressant d'étudier les mécanismes de sécurité des données de faible coût. En effet, notre technique de chiffrement de données peut être adaptable. Elle mérite d'être mis en œuvre sur un matériel mobile réel pour évaluer les coûts de calcul.
- Publier les deux travaux [2] et [4] qui reposent sur la virtualisation d'agents intelligents, ce qui permet de fournir une nouvelle solution requise par les plates-formes Cloud.

Enfin, nous affirmons que la sécurité de données en Cloud est toujours pleine de défis et d'une importance majeure. De nombreux problèmes de recherche restent à identifier et à étudier.

Bibliographie

[1] Y. Mohammed Amine, K. Okba, L. Abdelkader, B. Ahcène, E. Reinhardt et A. Muath, «An Adaptive and Efficient Fully Homomorphic Encryption Technique,» *The 2nd International Conference on Future Networks & Distributed Systems, Amman, Jordan,* 26-27 June 2018.

[2] Y. Mohammed Amine, K. Okba, L. Abdelkader, B. Ahcène et E. Reinhardt, «An Efficient and Adapted Fully Homomorphic and Order Preserving Encryption in Cloud Computing».

[3] Y. Mohammed Amine, K. Okba et B. Mounir, «A multi-agent system approach based on cryptographic algorithm for securing communications and protecting stored data in the cloud-computing environment,» *International Journal of Information and Computer Security ,* 07/Seb/2018.

[4] Y. Mohammed Amine, K. Okba, L. Abdelkader et I. Kerthio, «Intelligent cloud protection based on Multi Agent System Approach Using Advanced Cryptographic Algorithm».

[5] I. Foster et C. Kesselman, The grid: blueprint for a new computing infrastructure, San Francisco: Morgan Kaufmann, 1999.

[6] Z. KARTIT, «Contribution à la sécurité du Cloud Computing : Application des algorithmes de chiffrement pour sécuriser les données dans le Cloud Storage,» Université de mohammed V - Faculté des Sciences -, Rabat , 2016.

[7] M. A Vouk, «Cloud computing–issues, research and implementations,» *Journal of computing and information technology,* vol. 16, n° %14, pp. 235-246, 2008.

[8] S. Ried, K. Holger, M. Pascal, B. Andrew et L. Miroslaw, «Sizing the cloud, understanding and quantifying the future of cloud computing,» Forrester Research, Inc 21 , 2011.

[9] P. Mell et T. Grance, «The NIST Definition of Cloud Computing.,» NIST, [En ligne]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.. [Accès le 07 octobre 2018].

[10] Ibm, «Ibm cloud,» IBM, [En ligne]. Available: http://www.ibm.com/cloud-computing/us/en/what-is-cloud-computing.html. [Accès le 07 octobre 2018].

[11] G. IT, «Gartner IT glossary,» Technology Research, 2013.

[12] Y. Yang, K. Liu, J. Chen, . X. Liu, D. Yua et J. Hai, «An algorithm in SwinDeW-C for scheduling transaction-intensive cost-constrained cloud workflows.,» chez *eScience'08. IEEE Fourth International Conference*, 374-375, Indianapolis, USA, 2008.

[13] «http://www.expertglossary.com/cloud-computing/definition/linearly-scalable,» [En ligne]. [Accès le Septembre 2018].

[14] . M. Armbrust, F. Armando, G. Rean, D. J. Anthony, H. K. Randy, K. Andrew, L. Gunho, D. Patterson et A. Rabkin, «Above the Clouds: A Berkeley View of Cloud Computing,» UCB/EECS, UC Berkeley Reliable Adaptive Distributed Systems Laboratory, EECS Department, University of California, Berkeley, 2009.

[15] S. Subashini et K. Veeraruna, «A survey on security issues in service delivery models of cloud computing,» *Journal of network and computer applications,* vol. 34, n° %11, pp. 1-11, 2011.

[16] «Virtualization,» [En ligne]. Available: http ://en.wikipedia.org/w/index.php ?title=Virtualization. [Accès le 14 Octobre 2018].

[17] . P. Wieder, J. M. Butler et W. Theilmann, Service level agreements for cloud computing, Springer Science & Business Media, 2011.

[18] R. Buyya, K. G. Saurabh et N. C. Rodrigo, «SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions,» *Cloud and Service Computing (CSC), 2011 International Conference onCloud and Service Computing (CSC), Hong Kong, China,* pp. pp. 1-10. IEEE, 2011.

[19] R. Buyya, S. Y. Chee et V. Srikumar, «Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities,» *10th IEEE International Conference on on High Performance Computing and Communications (HPCC-08), Dalian, China,* pp. 5-13, September 2008.

[20] L. Wang, T. Jie, K. Marcel, C. C. Alvaro, K. David et K. Wolfgang, «Cloud Computing : Early Definition and Experience,,» *Proc. of the 10th IEEE Int. Conf. on High Performance Computing and Communications, Washington, DC, USA,* pp. 825-830, September 2008.

[21] [En ligne]. Available: http://www.hosting.com/resources/white-papers/what-cloud-computing-means-to-you-efficiency-flexibility/success-cloud-computing-means/.

[22] C. N. Höfer et G. Karagiannis, «Cloud computing services : taxonomy and comparison,» *Journal of Internet Services and Applications,* vol. 2, n° %12, pp. 81-94, 2011.

[23] I. Foster, Z. Yong, R. Ioan et L. Shiyong, «Cloud computing and grid computing 360-degree compared,» *In Grid Computing Environments Workshop, GCE'08, Ieee ,* pp. 1-10, 2008.

[24] R. N. Calheiros, R. Rajiv, B. Anton, A. D. R. César et B. Rajkumar, « CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,» *Software: Practice and experience,* vol. 41, n° %11, pp. 23-50, 2011.

[25] H. D, «Cloud computing : a taxonomy of platform and infrastructure-level offerings,» chez *Tech Rep GIT-CERCS-09-13,CERCS*, Georgia Institute of Technology., 2009.

[26] S. Weil, «CEPH : Reliable, scalable, and high-performance distributed storage,» PhD thesis, University of California - SANTA CRUZ, 2007.

[27] G. Diana, H. Marcus et A. Streit, «Evaluating the performance and scalability of the ceph distributed storage system.,» *IEEE International Conferenceon Big Data,* 2014.

[28] S. Konstantin, K. Hairong, R. Sanjay et C. Robert, «The hadoop distributed file system,» chez *In Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 2010 .

[29] D. Benjamin, L. M. Gaël et S. Cyril, «Analysis of six distributed file systems,» 2013.

[30] P. Mayur, I. Adriana, R. Matei et G. Simson, «Amazon s3 for science grids : a viable solution ?,» *In Proceedings of the 2008 international workshop on Data-aware distributed computing,* p. 55–64, June 2008.

[31] T. Yusuke, Y. Seiya et H. Takahiro, «A high performance, qos-enabled, s3-based object store,» *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing,* p. 784–791, 2014.

[32] [En ligne]. Available: https://cloud.google.com/developers/articles/building-high-availability-applications-ongoogle-compute-engine/.

[33] L. Qian, L. Zhiguo, D. Yujian et G. Leitao, «Cloud computing: An overview.,» *In IEEE International Conference on Cloud Computing, Springer, Berlin, Heidelberg,* pp. 626-631, 2009.

[34] G. Pierre et S. Corina, «ConPaaS: a platform for hosting elastic cloud applications,» *IEEE Internet Computing,* vol. 5, n° %15 , pp. 88-92, 2012.

[35] P. Guillaume et S. Corina, «Conpaas : A platform for hosting elastic cloud applications,» *IEEE Internet Computing,* p. 88–92, 2012.

[36] M. Azure, «Microsoft Azure,» Microsoft, [En ligne]. Available: https://azure.microsoft.com/fr-fr/overview/what-is-azure/. [Accès le 26 October 2018].

[37] «Microsoft Azure (Windows Azure),» Windows , [En ligne]. Available: https://www.lemagit.fr/definition/Microsoft-Azure-Windows-Azure.

[38] «Elastic Beanstalk,» amazon, [En ligne]. Available: AWS Elastic Beanstalk. https://aws.amazon.com/fr/elasticbeanstalk/.. [Accès le 26 October 2018].

[39] P. Guillaume et S. Corina, «Conpaas : A platform for hosting elastic cloud applications,» *IEEE Internet Computing,* p. 88–92, 2012.

[40] C. N. Höfer et G. Karagiannis, «Cloud computing services : taxonomy and comparison,» *Journal of Internet Services and Applications,* vol. 2, n° %12, pp. 81-94, 2011.

[41] A. Jula, E. Sundararajan et Z. Othman, «Cloud computing service composition : A systematic literature review,» *Expert Systems with Applications,* vol. 41, n° %18, pp. 3809-3824, 2014.

[42] I. Foster, Y. Zhao, I. Raicu et S. Lu, «Cloud Computing and Grid Computing 360-Degree Compared,» *Grid Computing Environments Workshop. GCE '08,* pp. 10-56, 2008.

[43] V. J. Winkler, Securing the Cloud: Cloud computer Security techniques and tactics, Elsevier, 2011.

[44] X. Xu, «From cloud computing to cloud manufacturing",,» *Robotics and computerintegrated manufacturing,* vol. 28, n° %11, p. 75 – 86, 2012.

[45] C. Curino, P. J. Evan, A. P. Raluca, M. Nirmesh, W. Eugene, M. Sam, B. Hari et Z. Nickolai, «Relational cloud: A database-as-a-service for the cloud,» MITLibraries, 2011.

[46] E. Brewer, «Towards robust distributed systems,» chez *In PODC (Vol 7)*, July 2000.

[47] R. Cattell, «Scalable SQL and NoSQL data stores,» *Acm Sigmod Record,* vol. 39, n° %14, pp. 12-27, 2011.

[48] H.-E. Chihoub, I. Shadi, L. Yue, A. Gabriel, P. María et B. Luc, «Exploring energy-consistency trade-offs in cassandra cloud storage system,» chez *In SBAC-PAD'15-The 27th International Symposium on Computer Architecture and High Performance Computing*, 2015.

[49] L. Bing, H. Yutao et X. Ke, «The nosql principles and basic application of cassandra model,» chez *International Conference on Computer Science and Service System*, 2012.

[50] K. Chodorow, MongoDB: The Definitive Guide: Powerful and Scalable Data Storage, O'Reilly Media, 2013.

[51] A. Ortiz, J. Jacques et M. Abdelaziz, « Toward a new direction on data management in grids,» *In 15th IEEE International Conference on High Performance Distributed Computing,* pp. 377-378, 2006.

[52] V. J. Winkler, Securing the Cloud: Cloud computer Security techniques and tactics, Elsevier, 2011.

[53] S. Srinivasan, «Cloud computing basics,» Springer, 2014, p. 44.

[54] L. Wu et Y. Chengwei, «A solution of manufacturing resources sharing in cloud computing environment,» *In International Conference on Cooperative Design, Visualization and Engineering, Springer, Berlin, Heidelberg,* pp. 247-252, 2010.

[55] D. Sarna, chez *Implementing and Developing Cloud Computing Applications*, Taylor & Francis Group, 2010, p. 2.

[56] A. Lee-Post et P. Ram, «Cloud computing: A comprehensive introduction,» *In Security, Trust, and Regulatory Aspects of Cloud Computing in Business Environments, IGI Global,* pp. 1-23, 2014.

[57] M. Abdelhak, Approche de composition de services web dans le Cloud Computing basée sur la coopération des agents, Université Mohamed Khider – BISKRA, Département d'informatique, 20/02/2018.

[58] D. Rountree et I. Castrillo, chez *The Basics of Cloud Computing Understanding the Fundamentals of Cloud Computing in Theory and Practice*, Elsivier , 2014, p. 45.

[59] T. DILLON, C. WU et E. CHANG, «Cloud computing: issues and challenges,» *In : Advanced Information Networking and Applications (AINA), 24th IEEE International Conference on,* pp. 27-33, 2010.

[60] A. Michael, D. J. Anthony, H. K. Randy et A. P. David, «Above the Clouds : A Berkeley View of Cloud Computing,» chez *Technical Report UCB/EECS-2009-28,* , EECS Department , University of California, Berkeley, 2009.

[61] M. Marzolla, B. Ozalp et P. Fabio, «Server consolidation in clouds through gossiping,» *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks,* pp. 1-6, 2011.

[62] R. J. R, Z. L. Z, M. L. M, L. K. L et L. J. L, «Distributed media-aware flow scheduling in cloud computing environment,» *Computer Communications. 35(1),* p. 1819–1827, September 2012.

[63] T. C. T et R. J. R, «Metaheuristic scheduling for cloud : A survey,» *IEEE Systems,* vol. 8, n° %11, p. 279–291, 2014.

[64] B. Rochwerger, D. Breitgand et A. Epstein, «Reservoir-when one cloud is not enough',» *Computer,* vol. 44, n° %13, p. 1–7, 2011.

[65] V. BHAGAWAT et A. KUMAR, «Survey on data security issues in cloud environment.,» *International Journal of Innovative Research in Advanced Engineering,* vol. 2, n° %11, pp. 31-35, 2015.

[66] N. MacDonald, «Security is the top Concern for Public Cloud, but What Does That Really Mean?,» [En ligne]. Available: https://blogs.gartner.com/neil_macdonald/2010/12/16/security-is-the-top-concern-for-public-cloud-but-what-does-that-really-mean/. [Accès le 13 Novembre 2018].

[67] R. McMillan, «Cloud Computing est un «cauchemar de la sécurité»,» Cisco, http://www.pcworld.com/article/163681/article.html, Consulté Novembre 2018.

[68] N. Gohring, «Amazon's data storage service hit by outage,» [En ligne]. Available: https://www.computerworld.com/article/2537118/data-center/amazon-s-data-storage-service-hit-by-outage.html. [Accès le Novembre 2018].

[69] «Google Docs leaks out private data,» [En ligne]. Available: https://www.infosecurity-magazine.com/news/google-docs-leaks-out-private-data/. [Accès le Novembre 2018].

[70] J. Kirk, «Google's gmail hit with two-hour outage,,» [En ligne]. Available: https://www.computerworld.com/article/2531439/networking/google-s-gmail-hit-with-two-hour-outage.html. [Accès le Novembre 2018].

[71] D. Chen et H. Zhao, « Data security and privacy protection issues in cloud computing,» *In : Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on. IEEE,* p. 647–651, 2012 .

[72] J. Brodkin, « Loss of customer data spurs closure of online storage service ’the linkup’,» [En ligne]. Available: http://www.networkworld.com/article/2274737/ data-center/loss-of-customer-data-spurs-closure-of-online-sto html. [Accès le Novembre 2018].

[73] R. CHOW, P. GOLLE, M. JAKOBSSON, E. Shi, J. Staddon, R. Masuoka et J. Molina, «Controlling data in the cloud: outsourcing computation without outsourcing control,» *In : Proceedings of the 2009 ACM workshop on Cloud computing security. ACM,* pp. 85-90, 2009.

[74] H. TIANFIELD, « Security issues in cloud computing.,» *In : Systems, Man, and Cybernetics (SMC) 2012 IEEE International Conference on. IEEE,* pp. 1082-1089, 2012.

[75] D. BORU, D. KLIAZOVICH, F. GRANELLI, P. Bouvry et A. Y. Zomaya, « Energy-efficient data replication in cloud computing datacenters.,» *Cluster computing,* vol. 18, n° %1 1, pp. 385-402, 2015.

[76] D.-W. SUN, G.-R. CHANG, S. GAO, L. Z. Jin et X. W. Wang, «Modeling a dynamic data replication strategy to increase system availability in cloud computing environments.,» *Journal of computer science and technology,* vol. 27, n° %12, pp. 256-272, 2012.

[77] P. X. WEN et L. DONG, «Quality model for evaluating SaaS service,» *In : Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on. IEEE,* pp. 83-87, 2013.

[78] P. RANE, Securing SaaS applications: a cloud security perspective for application providers, Information Security Management Handbook, vol. 5, 2010.

[79] R. SHIREY, « Internet security glossary,» *version 2,* 2007.

[80] A. NAGARAJAN et V. VARADHARAJAN, «Dynamic trust enhanced security model for trusted platform based services,» *Future Generation Computer Systems,* vol. 27, n° %15, pp. 564-573, 2011.

[81] Y. Zhang et J. Joshi, Access control and trust management for emerging multidomain environments, In: S. Upadhyaya and R. O. Rao, eds., Annals of Emerging Research in Information Assurance, Security and Privacy Services, Emerald Group Publishing, 2009.

[82] D. SHIN et G.-J. AHN, «Role-based privilege and trust management.,» *Computer Systems Science and Engineering,* vol. 20, n° %16, pp. 401-410, 2005.

[83] M. KROTSIANI et G. SPANOUDAKIS, «Continuous certification of non-repudiation in cloud storage services,» *In : Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on. IEEE,* pp. 921-928, 2014.

[84] A. REED, C. REZEK et P. SIMMONDS, «Security guidance for critical areas of focus in cloud computing,» *v3. 0. Cloud Security Alliance, ,* pp. 14-44, 2011.

[85] J. G. BRODKIN, «Seven cloud-computing security risks.,» *Infoworld,* vol. 2008, pp. 1-3, 2008.

[86] A. M., G. J., M. I. et e. al., «An analysis of the cloud computing security problem,» *in: Proceedings of APSEC 2010 Cloud Workshop, Sydney, Australia, ,* 30 Nov, 2010..

[87] S. RAMGOVIND, M. M. ELOFF et E. SMITH, « The management of security in cloud computing,» *In : Information Security for South Africa (ISSA). IEEE,* pp. 1-7, 2010.

[88] B. R. Kandukuri et R. Atanu, «Cloud security issues." Services Computing,» *SCC'09. IEEE International Conference on. IEEE,* 2009.

[89] M. David et S. Stuart, «Self Hosting vs. Cloud Hosting: Accounting for the security impact of hosting in the cloud,» *Microsoft Research,* pp. 1 - 18, 2010.

[90] M. Jensen, J. Schwenk, N. Gruschka et L. L. Iacono, «On technical security issues in cloud computing.,» *In Cloud Computing CLOUD'09. IEEE International Conference on,* pp. 109-116, Septembre 2009.

[91] C. Soghoian, «Caught in the cloud: Privacy, encryption, and government back doors in the web 2.0 era,» *J. on Telecomm. & High Tech. L,* vol. 8, p. 359, 2010.

[92] M. Armbrust, A. Fox, R. Griffith, D. A. Joseph, H. R. Katz, A. Konwinski, G. Lee, A. D. Patterson, A. Rabkin, A. Stoica et M. Zaharia, « Above the clouds: A Berkeley view of Cloud Computing,» *UC Berkeley EECS,* Feb 2010.

[93] «Cloud security management,» [En ligne]. Available: https://www.uniprint.net/fr/cloud-security-management-8-steps/. [Accès le 22 Novembre 2018].

[94] J. Networks, «Security Consideration for Cloud Ready DataCentres,» Oct. 2009. [En ligne]. Available: http://www.juniper.net/us/en/local/pdf/whitepapers/2000332-en.pdf. [Accès le 27 Novembre 2018].

[95] S. Subashini et K. Veeraruna, «A survey on security issues in service delivery models of cloud computing,» *Journal of network and computer applications ,* vol. 34, n° %11, pp. 1-11, 2011.

[96] A. A. Noureddine et Meledath Damodaran, «Security in web 2.0 application development,» chez *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, ACM, 2008.

[97] B. Benatallah, F. Casati, F. Daniel et J. Yu, «Mashups, saas, and cloud computing: Evolutions and revolutions in the integration landscape.,» Document based on Tutorial at ICDE , 2009.

[98] R. Pradnyesh, Securing SaaS Applications: A Cloud Security Perspective for Application Providers," Information Systems Security", 2010.

[99] acunetix, «Cross-site Scripting (XSS) Attack,» [En ligne]. Available: https://www.acunetix.com/websitesecurity/cross-site-scripting/. [Accès le 23 Novembre 2018].

[100] «Web 2.0/SaaS Security, Tokyo Research Laboratory,,» IBM Research, [En ligne]. Available: http://www.trl.ibm.com/projects/web20sec/web20sec_e. htm. [Accès le 27 Novembre 2018].

[101] L. NEWCOMBE, Securing cloud services: a pragmatic approach to security architecture in the cloud, IT Governance Publishing, 2012.

[102] J. Katz, Efficient cryptographic protocols preventing" man-in-the-middle" attacks, Columbia University, 2002.

[103] E. Ogren, «Whitelists SaaS modify traditional security, tackle flaws,» 17 Sep. 2009. [En ligne]. Available: http://searchsecurity. techtarget. com/news/column/0,294698,sid14_gci1368647,00.html/. [Accès le Novembre 2018].

[104] «Le DNS Spoofing : explications,» [En ligne]. Available: https://www.securiteinfo.com/attaques/hacking/dnsspoofing.shtml. [Accès le 24 Novembre 2018].

[105] C. Sample, «Senior Scientist, BBN Technologies, Diana Kelley, Partner, Security Curve,"Cloud computing security: Routing and DNS security threats».

[106] R. L. Krutz et D. V. Russell, Cloud security: A comprehensive guide to secure cloud computing, Wiley Publishing, 2010.

[107] «Balayage de ports,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Balayage_de_ports. [Accès le 24 Novembre 2018].

[108] Z. TRABELSI, H. RAHMANI, K. KAOUECH et M. Frikha, «Malicious sniffing systems detection platform.,» *In : Applications and the Internet, 2004. Proceedings. 2004 International Symposium on. IEEE,* pp. 201-207, 2004.

[109] J. KARLIN, S. FORREST et J. REXFORD, «Autonomous security for autonomous systems.,» *Computer Networks,* vol. 52, n° %115, pp. 2908-2923, 2008.

[110] «Scalable Security Solutions, Check Point Open Performance Architecture, Quad-Core Intel Xeon Processors, "Delivering Application-Level Security at Data Centre Performance Levels,» Intel Corporation, http://download.intel.com/netcomms/technologies/security/320923.pdf, 2008.

[111] R. Bhadauria, R. Chaki, N. Chaki et S. Sanyal, « SECURITY ISSUES IN CLOUD COMPUTING.,» *Acta Technica Corvininesis-Bulletin of Engineering, ,* vol. 7, n° %14, pp. 159-177, 2014.

[112] S. Pal, S. Khatua, N. Chaki et S. Sanyal, «A new trusted and collaborative agent based approach for ensuring cloud security.,» *arXiv preprint arXiv:1108.4100,* 2011.

[113] F. LOMBARDI et R. DI PIETRO, «Secure virtualization for cloud computing.,» *Journal of network and computer applications,* vol. 34, n° %14, pp. 1113-1122, 2011.

[114] H. Wu, Y. Ding, C. Winer et L. Yao, «Network security for virtual machine in cloud computing,» *In Computer Sciences and Convergence Information Technology (ICCIT), 5th International Conference on. IEEE,* pp. 18-21, 2010.

[115] K. VIEIRA, A. SCHULTER et C. WESTPHALL, « Intrusion detection techniques in grid and cloud computing environment,» *IT Professional, IEEE Computer Society,* vol. 12, n° %14, pp. 38-43, 2010.

[116] C.-C. Lo, H. Chun-Chieh, K. Joy et ", «A cooperative intrusion detection system framework for cloud computing networks,» chez *Parallel processing workshops (ICPPW), 2010 39th international conference*, IEEE, 2010.

[117] L. Ruiping et C. Y. Kin, «Mitigating DDoS Attacks with Transparent and Intelligent Fast-Flux Swarm Network,» *IEEE Network,* vol. 25, n° %14, pp. 28-33, 2011.

[118] A. BAKSHI et Y. B. DUJODWALA, «Securing cloud from ddos attacks using intrusion detection system in virtual machine,» *In : Communication Software and Networks. ICCSN'10. Second International Conference on. IEEE,* pp. 260-264, 2010.

[119] M. Claudio, B. Roberto et C. Roberto, «Integrating a Network IDS into an Open Source Cloud Computing Environment,» *Sixth International Conference on Information Assurance and Security, USA,* pp. 265-270, 2010.

[120] «Cookies et sécurité,» securiteinfo, [En ligne]. Available: https://www.securiteinfo.com/conseils/cookies.shtml. [Accès le 29 Novembre 2018].

[121] D. GOLLMANN, «Securing web applications.,» *Information Security Technical Report,* vol. 13, n° %11, pp. 1-9, 2008.

[122] R. BHADAURIA, R. CHAKI, N. CHAKI et S. Sanyal, «SECURITY ISSUES IN CLOUD COMPUTING,» *Acta Technica Corvininesis-Bulletin of Engineering,* vol. 7, n° %14, 2014.

[123] Z. TRABELSI, H. RAHMANI, K. KAOUECH et M. Frikha, «Malicious sniffing systems detection platform,» *In : Applications and the Internet. Proceedings. International Symposium on. IEEE,* pp. 201-207, 2004.

[124] V. (. Winkler, La sécurité dans le Cloud, Pearson Education France, 2011 .

[125] D. Chen et H. Zhao, «Data security and privacy protection issues in cloud computing,» *Computer Science and Electronics Engineering (ICCSEE), International Conference on. IEEE,* vol. 1, 2012.

[126] F. OGIGAU-NEAMTIU, «Cloud computing security issues,» *Journal of Defense Resources Management,* vol. 3, n° %12, p. 141, 2012.

[127] «Data security lifecycle 2.0,» [En ligne]. Available: https://www.securosis.com/blog/ data-security-lifecycle-2.0. [Accès le 03 Decembre 2018].

[128] A. REED, C. REZEK et P. SIMMONDS, «Security guidance for critical areas of focus in cloud computing v4,» *Cloud Security Alliance,* pp. 14-44, 2017.

[129] K. RAUBER, «Cloud cryptography,» *International Journal of Pure and Applied Mathematics,* vol. 85, n° %11, pp. 1-11, 2013.

[130] T. Mather, K. Subra et L. Shahed, Cloud security and privacy: an enterprise perspective on risks and compliance, O'Reilly Media, Inc, 2009.

[131] B. PRINCE, « Ibm discovers encryption scheme that could improve cloud security,» *spam filtering. E-Week. com,* 2009.

[132] Y. L. SIMMHAN, B. PLALE et D. GANNON, «A survey of data provenance techniques,» *Computer Science Department, Indiana University, Bloomington IN,* vol. 47405, p. 69, 2005.

[133] «Data remanence,» Wikipedia, [En ligne]. Available: https://en.wikipedia.org/wiki/Data_remanence. [Accès le 03 decembre 2018].

[134] P. BARNAGHI, W. WANG, C. HENSON et K. Taylor, «Semantics for the Internet of Things: early progress and back to the future,» *International Journal on Semantic Web and Information Systems (IJSWIS),* vol. 8, n° %11, pp. 1-21, 2012.

[135] V. P. P., B. P. S., N. P. S., I. B. N. et H. J. R., «Survey of various homomorphic encryption algorithms and schemes,» *International Journal of Computer Applications,* vol. 91, n° %18 , 2014.

[136] S. S. et K. V., «A survey on security issues in service delivery models of cloud computing,» *Journal of network and computer applications,* vol. 34, n° %11, p. 1–11, 2011.

[137] M. T., O. A., C. H., B. I., P. J.-M. et V. V. A., «Cloud computing: Survey on energy efficiency,» *Acm computing surveys (csur),* vol. 47, n° %12, p. 33, 2015.

[138] M. K. I., K. A. et A. M., «Cloud computing security: A survey,» *Computers,* vol. 3, n° %11, p. 1–35, 2014.

[139] D. A., Q. Y., W. Q. et H. S., «An advanced survey on secure energy-efficient hierarchical routing protocols in wireless sensor networks,» *arXiv preprint arXiv,* pp. 1306-4595, 2013.

[140] M. S. J. et P. H. K. A., «A survey on homomorphic encryption techniques in cloud computing,» *Development,* vol. 2, n° %12, 2015.

[141] M. C., O. M., O. E., D. Y. et S. B., «Practical homomorphic encryption: A survey,” in Circuits and Systems (ISCAS),» *IEEE International Symposium on. IEEE,* p. 2792–2795, 2014.

[142] E.-Y. A. et D. E. M., «Fully homomorphic encryption: state of art and comparison,» *International Journal of Computer Science and Information Security,* vol. 14, n° %14, p. 159, 2016.

[143] F. C. et G. F., «A survey of homomorphic encryption for nonspecialists,» *EURASIP Journal on Information Security,* vol. 2007, p. 15, 2007.

[144] Z. O., D. A. et D. H., «Cloud computing security through parallelizing fully homomorphic encryption applied to multi-cloud approach,» *Journal of Theoretical & Applied Information Technology,* vol. 87, n° %12, 2016.

[145] B. A., C. M., Q. B., A. F. et S. P., «Depsky: dependable and secure storage in a cloud-of-clouds,» *ACM Transactions on Storage (TOS),* vol. 9, n° %14, p. 12, 2013.

[146] A. M. A., P. E., S. B. et T. J. A., «Cloud computing security: from single to multi-clouds,» *System Science (HICSS), 2012 45th Hawaii International Conference on. IEEE,* p. 5490–5499, 2012.

[147] D. B. J., J. A. et O. A., «Hail: A high-availability and integrity layer for cloud storage,» *Proceedings of the 16th ACM Conference on Computer and communications Security. ACM,* p. 187–198, 2009.

[148] X. K., L. S., H. J., X. Y., Y. N. et H. P., «Two-cloud secure database for numeric-related sql range queries with privacy preserving,» *IEEE Transactions on Information Forensics and Security,* vol. 12, n° %17, p. 1596–1608, 2017.

[149] F. L., C. M. et M. M., «Distributed, concurrent, and independent access to encrypted cloud databases,» *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, n° %12, p. 437–446, 2014.

[150] A. P. R., R. C., Z. N. et B. H., «Cryptdb: protecting confidentiality with encrypted query processing,» *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM,* p. 85–100, 2011.

[151] G. S. et M. S., «Probabilistic encryption,» *Journal of Computer and System Sciences,* n° %12, p. 270–299, 1984.

[152] N. D. et S. J., «A new public-key cryptosystem,» *International Conference on the Theory and Applications of Cryptographic Techniques. Springe,* p. 27–36, 1997.

[153] O. T. et U. S., «A new public-key cryptosystem as secure as factoring,» *International Conference on the Theory and applications of cryptographic techniques. Springer,* p. 308–318, 1998.

[154] P. P., «Public-key cryptosystems based on composite degree residuosity classes,» *International Conference on the Theory and Applications of Cryptographic Techniques. Springer,* p. 223– 238, 1999.

[155] T. T. et S.-S. K., «Paillier's cryptosystem modulo p2q and its applications to trapdoor commitment scheme,» 2005.

[156] L. R. R., S. A. et A. L., «A method for obtaining digital signatures and public-key cryptosystems,» *Communications of the ACM,* vol. 21, n° %12, p. 120–126, 1978.

[157] E. T., «A public key cryptosystem and a signature scheme based on discrete logarithms,» *IEEE Transactions on Information Theory,* vol. 31, n° %14, p. 469–472, 1985.

[158] S. T., Y. A. et Y. M., «Non-interactive cryptocomputing for nc/sup 1,» *Foundations of Computer Science, 1999. 40th Annual Symposium on. IEEE,* p. 554–566, 1999.

[159] K. P. S., R. S. et B. R., «An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing,» *Journal of Network and Computer Applications,* vol. 64, p. 12–22, 2016.

[160] K. J., « Theory of Cryptography: Second Theory of Cryptography Conference,» *TCC 2005, Cambridge, MA, USA, Proceedings. Springer, vol. 3378,* 10-12 February 2005.

[161] M. A., G. P. et H. J., «Additively homomorphic encryption with d-operand multiplications,» *Annual Cryptology Conference. Springer,* p. 138–154, 2010.

[162] D. Z., «Secure database in cloud computing-cryptdb revis-˘ ited,» *International Journal of Information Security Science,* vol. 3, n° %11, p. 129–147, 2014.

[163] J. S., L. W. et S. T., «Numerical sql value expressions over encrypted cloud databases,» *International Conference on Database and Expert Systems Applications. Springer,* p. 455–478, 2015.

[164] S. T., «Dodrant-homomorphic encryption for cloud databases using table lookup,» *Networks, Computers and Communications (ISNCC), 2017 International Symposium on. IEEE,* p. 1–6, 2017.

[165] C. D., G. R., H.-G. N. et Q. N. P., «Paillier's cryptosystem revisited,» *Proceedings of the 8th ACM Conference on Computer and Communications Security. ACM,* p. 206–214, 2001.

[166] G. C., A fully homomorphic encryption scheme, Stanford : Stanford University, 2009.

[167] B. Z. et e. al., Efficient fully homomorphic encryption from (standard) lwe, focs, 2011.

[168] M. M., S. K. et G. M., «Enabling secure database as a service using fully homomorphic encryption: Challenges and opportunities,» *arXiv preprint arXiv: 1302. 2654,* 2013.

[169] B. A., C. N., L. Y. et O. A., «Orderpreserving symmetric encryption,» *Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer,* p. 224–241, 2009.

[170] L. D. et W. S., «Programmable order-preserving secure index for encrypted database query," in,» *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE,* p. 502–509, 2012.

[171] L. D. et w. S., «Nonlinear order preserving index for encrypted database query in service cloud environments,» *Concurrency and Computation: Practice and Experience,* vol. 25, n° %113, p. 1967–1984, 2013.

[172] D. J., D. M. et X. J., «Practical homomorphic encryption over the integers,» *ArXiv Preprint ArXiv:1702.07588, ,* 2017.

[173] J.-S. CORON, A. MANDAL, D. NACCACHE et M. Tibouchi, «Fully homomorphic encryption over the integers with shorter public keys.,» *Annual Cryptology Conference. Springer, Berlin, Heidelberg,* pp. 487-504, 2011.

[174] «Postgres Plus Cloud Database enterprisedb,» [En ligne]. Available: http://enterprisedb.com/cloud-database. [Accès le 14 12 2018].

[175] J. Rittinghouse et J. Ransome, «Cloud Computing: Implementation, Management, and Security,» Inc. Boca Raton, FL, USA, CRC Press, 2009, p. Chapter 6.

[176] M. Armbrust, A. Fox, G. R. A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, R. Ariel, I. Stoica et Z. Matei, «A view of cloud computing,» *Communications of the ACM,* vol. 53, n° %14, pp. 50-58, 2010.

[177] J. Kang et K. M. Sim, «Cloudle: An ontology-enhanced cloud service search engine,» *In Web Information Systems Engineering–WISE 2010 Workshops. Springer Berlin Heidelberg,* pp. 416-427, 2011, January.

[178] J. Kang et K. M. Sim, «Ontology and search engine for cloud computing system,» *In System Science and Engineering (ICSSE), International Conference IEEE,* pp. 276-281, 2011 .

[179] K. M. Sim, «Agent-based Cloud Computing,» *IEEE TRANSACTIONS ON SERVICES COMPUTING,* 2012.

[180] K. M. Sim, « Complex and concurrent negotiations for multiple interrelated emarkets,» *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetic ,* vol. 43, n° %11, pp. 230-245, 2013.

[181] D. Talia, «Cloud Computing and Software Agents: Towards Cloud Intelligent Services,» *In WOA,* pp. 2-6, 2011.

[182] D. Talia, « Clouds meet agents: Toward intelligent cloud services,» *Internet Computing, IEEE,* vol. 16, n° %12, pp. 78-81, 2012.

[183] S. Venticinque, R. Aversa, B. Di Martino, M. Rak et D. Petcu, «A cloud agency for SLA negotiation and management,» *In Euro-Par 2010 Parallel Processing Workshops, Springer Berlin Heidelberg,* pp. 587-594, 2010 .

[184] F. J., Les systèmes Multi-Agents Vers une intelligence collective., Paris: Inter Edition, 1995.

[185] J. I. et C.-D. B., Aperçu sur les systèmes Multi-Agents, CIRANO: Série Scientifique du centre inter universitaire de recherche en analyse des organisations, 2002.

[186] N. R. Jennings, «On agent-based software engineering,» *Artificial Intelligence Journal,* 2000.

[187] o. Khayati, «Modèles formels et outils génériques pour la gestion et la recherche de composants,» chez *Thèse, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE*, 2005.

[188] J. Ferber, Les systèmes multi-agents, vers une intelligence collective, InterEditions, 1995.

[189] N. R. Jennings, K. Sycara et M. Wooldridge, «A Roadmap of Agent Research and Development,» *Journal of Autonomous Agents and Multi-Agent Systems. Boston, USA: Kluwer Academic Publishers.,* 1998.

[190] K. Sycara, «Multi-agent system,» *AI Magazine,* vol. 19, n° %12, p. 79–92, 1998.

[191] S. O. K. Hamza, «Découverte de services web via le Cloud computing à base d'agents mobiles,» chez *thèse de doctorat, Université de Biskra*, 2015.

[192] S. Perret, «Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure,» chez *Thèse de doctorat de l'Université Joseph Fourier - Grenoble I*, 1997.

[193] L. M. Jeffrey et J. P. Tsai, «Security modeling and analysis of mobile agent systems,» chez *Series in Electrical and Computer Engineering Vol 5*, 2006.

[194] F. A. BARIKA, «Vers un IDS Intelligent à base d'Agents Mobiles,» chez *Mémoire de DEA, Université de Tunis Institut Supérieur de Gestion, SOI2E-ISG-TUNIS, LERIA-EPITECH-PARIS*, 2003.

[195] S. Rodríguez González, Modèle adaptatif pour les organisations d'agents virtuels, 2010.

[196] J. W. Ross et G. Westerman, «Preparing for utility computing: The role of IT architecture and relationship management,» *IBM systems journal,* vol. 43, n° %11, pp. 5-19, 2004.

[197] A. M. R. A. R. A. e. M. A. A. M. Talib, «Security framework of cloud data storage based on multi agent system architecture - a pilot study,» *international conference In International Conference on Information Retrieval and Knowledge Management,* 2012.

[198] A. M. Talib et E. N. E. M., «Multi agent system-based on case based reasoning for cloud computing system,» *In APJES,* vol. 2, n° %12, p. 34–38, 2014.

[199] H. Zhou et Q. S., «Security framework for cloud data storage based on multiagent system.,» *Computer Modelling & New Technologies,* p. 548–553, 2014.

[200] K. Govinda et E. Sathiyamoorthy, «Agent based security for cloud computing using obfuscation',,» *International Conference On Modelling Optimization And Computing Procedia Engineering,* vol. 38, p. 125–129, 2012.

[201] B. F., P. A. et R. G., «Developing multi-agent systems with JADE,» *Intelligent Agents VII Agent Theories Architectures and Languages, Springer,* p. 89–103, 2001.

[202] F. Bellifemine, A. Poggi et G. Rimassa, «JADE—a FIPA compliant agent framework,» *Proceedings of the 4th international conference and exhibition on the practical application of intelligent agents and multi-agents, UK,* p. 97–108, 1999.

[203] M. N. Huhns, «Agents as Web services,» *IEEE Internet computing,* vol. 6, n° %14, pp. 93-95, 2002.

[204] A. Poggi, G. Rimassa et M. Tomaiuolo, «Multi-user and security support for multi-agent systems,» *Proceedings of WOA, Dagli oggetti agli,* 2001 .

[205] F. Bellifemine, A. Poggi et G. Rimassa, «JADE–A FIPA-compliant agent framework,» *Proceedings of PAAM ,* vol. 99, pp. 97-108, 1999.

[206] N. Rodrigo, R. Rajiv, B. Anton, D. César, B. Rajkumar, N. Calheiros, R. Ranjan, A. Beloglazov, A. De-Rose et R. Buyya, «CloudSim : A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,» *Software : Practice and Experience (SPE),* vol. 41, n° %11, pp. 23-50, 2011.

[207] K. Saurabh et B. Rajkumar, «NetworkCloudSim : Modelling Parallel Applications in Cloud Simulations,» *Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing, Victoria, Canada,* n° %1105-113, 2011.