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

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseau & télécommunication

Par :

BOUZARA Ayoub

Sur le thème

Sécurité dans le Cloud Computing

Soutenu publiquement le 26/06/2018 à Tiaret devant le jury composé de :

Mr.Zioual Taher	Université Ibn Khaldoun	Président
Mr.Mestfaoui Kadda Mokhtar	Université Ibn Khaldoun	Encadreur
Mr.Mokhtari Ahmed	Université Ibn Khaldoun	Examinateur

Remerciements

Au premier lieu je tiens à remercier vivement mon encadrant qui m'a réservé toute sa disponibilité, ses orientation de méthodes de travail pour faire dresser et soigner l'aspect d'une bonne tache de recherche.

Je tiens à présenter mes remerciements au monsieur les membres jurys qui ont examiné avec rigueur et ont données leurs meilleur et juste jury.

C'est avec une grande fierté que j'adresse mes remerciements avec égard à tous mes profs qui ont donné de leur meilleur savoir et formation.

Dédicaces

A mes Parents,

A mes sœurs et mes frères,

A toute ma famille,

A mon oncle Deifellah,

Ames très chères amis,

A mes enseignants tous au long de mon cursus scolaire et universitaire.

Sommaire :

Listes des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Le Cloud Computing est apparu comme une technique importante dans le domaine des applications informatiques et de la technologie de l'information. Il s'agit de services de stockage, de traitement et de transmission de données via des ressources partagées, sur Internet. Les ressources utilisées pour ces services peuvent être mesurées et les clients peuvent être facturés pour les ressources qu'ils utilisent.

Le Cloud Computing offre de nombreuses avantages comme la réduction des couts, le déploiement rapide, le paiement à l'usage, l'évolutivité facile, l'accès ubiquitaire aux ressources, etc. En raison de diverses caractéristiques, il devient une solution intéressante pour les entreprises et les chercheurs.

Cependant, son adoption est confrontée à un certain nombre de défis, tels que les problèmes de sécurité, les défis juridiques et organisationnels. Les données stockées dans le Cloud sont confidentielles et peuvent être sensibles à l'entreprise et sont susceptibles d'être exploitées par un tierce non autorisées. 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. Cette faiblesse nous a motivés pour proposer une solution qui permet aux utilisateurs de sécuriser leurs données pour éviter leurs utilisations malveillantes.

Un autre objectif de ce travail est d'approfondir et d'expérimenter nos connaissances sur ce thème de Cloud Computing, puis de faire son état de l'art, en vue de choisir la meilleure solution disponible à l'heure actuelle, et de la déployer.

Ce travail s'articule autour de quatre chapitres, dans le premier chapitre on donnera une vision plus claire sur la notion de Cloud Computing et leurs technologies constructives, les différents modèles de service et différents modes de son déploiement ainsi que les différents avantages et inconvénients.

Le second chapitre, sera consacré pour les principales notions de sécurité autour de Cloud Computing. Dans le troisième chapitre on va présenter une méthode pour augmenter la sécurité des données. Le dernier chapitre va décrire en détail les spécifications techniques matérielles et l'architecture réseau ainsi que toutes les étapes requis pour mettre une solution de Cloud privée en place.

Chapitre I

Généralités sur le Cloud Computing.

1.1. Historique:

La notion de Cloud fait référence à un nuage, tel que l'on a l'habitude de l'utiliser dans les schémas techniques pour représenter Internet. Cette utilisation remonte aux années 90 lorsque les ingénieurs du réseau ont représenté l'ensemble des équipements interconnectés sur internet en utilisant des nuages [1].

Le terme Cloud est une métaphore exprimant la similarité avec le réseau électrique. Au cours des 1910, bien avant l'arrivée des services d'électricité, les usines et les entreprises ; en particulier les unités de production ; devaient disposer d'énormes infrastructures et personnels pour assurer la production de leur propre électricité nécessaire à leur fonctionnement, s'éloignant ainsi de leurs activités principales et dépensant énormément.

Avec l'émergence des réseaux électriques, les entreprises commençaient progressivement à s'approvisionner de l'électricité auprès d'une source extérieure. Les propriétaires bénéficiaient de cette avancée en se débarrassant du fardeau des unités de production, des tâches et dépenses liées aux charbon (livraison et nettoyage) et des recrutements d'ingénieurs mécaniciens et électriciens, Mais la fiabilité de ces réseaux électriques les préoccupaient [2].

La similarité du Cloud Computing avec le réseau électrique est nette, les grandes centrales étant les Datacenter, le réseau étant celui d'Internet et l'électricité correspond aux ressources informatiques.

Un réseau Internet est constitué d'une multitude de systèmes fournissant des services et des informations. Le Cloud Computing est dans cette lignée: un ensemble de services et de données consommables.

Cette notion de consommation a été proposée en 1961, lors d'une conférence au MIT (Massachusetts Institute of Technology), par John McCarthy considéré comme l'un des pionniers de l'intelligence artificielle. John McCarthy suggéra que la technologie informatique partagée (« Time-sharing ») pouvait construire un bel avenir dans lequel la puissance de calcul et les applications spécifiques pouvaient être vendues comme un service public. Cette idée, très populaire dans les années 60, disparaissait au milieu des années 70 : à l'époque, les technologies matérielles, logicielles et réseaux n'étaient tout simplement pas prêts.

En 1999, Salesforce.com a présenté pour la première fois des applications livrées aux entreprises via internet, ensuite en 2002, Amazon a lancé les AWS (Amazon Web Services) qui offrent des services de stockage et de calcul. Le Cloud Computing n'est véritablement apparu qu'au cours de l'année 2006 avec l'apparition d'Amazon EC2 (Elastic Compute Cloud). Début 2008,

OpenNebula et NASA, mettent en valeur le projet RESERVOIR financé par la Commission européenne qui est devenu le premier logiciel open source pour déployer des services de Clouds privés et hybrides. Ce n'est qu'en 2009 que la réelle explosion du Cloud survint avec l'arrivée sur le marché de sociétés comme Google (Google App Engine), Microsoft (Microsoft Azure), IBM (IBM Smart Business Service), Sun (Sun Cloud) et Canonical Ltd (Ubuntu Enterprise Cloud) [3].

D'après Cisco Global Cloud Index, le trafic du Cloud Computing devrait atteindre plus de 19 Zettabytes d'ici 2021. Figure 1.1: représente la croissance de trafic du Cloud Computing:

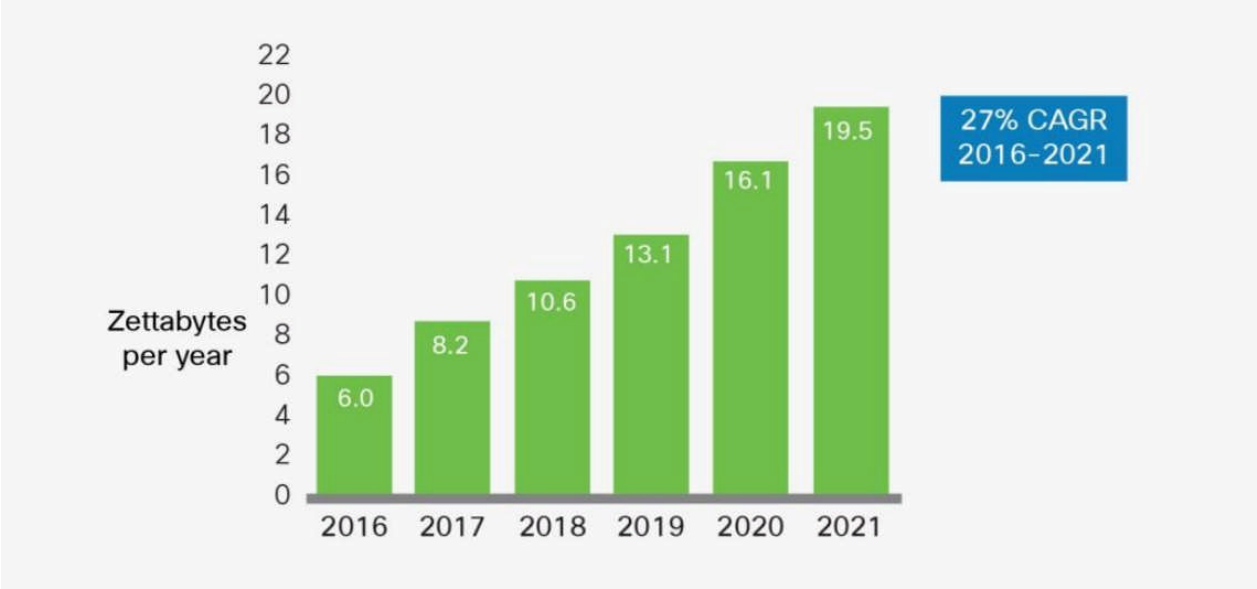

Figure 1.1: Croissance de trafic du Cloud Computing 2016-2021 [4].

1.2. Définition:

Il est difficile d'attribuer une définition précise au Cloud, du fait de son évolution progressive avec l'avènement de plusieurs technologies et de concepts ainsi que de son utilisation très vaste (diversité et la richesse). Le Cloud Computing n'est pas une technologie à part entière mais un paradigme associé aux technologies de l'information, il consiste à la livraison à la demande de puissance de calcul, de stockage sur base de données, d'applications et d'autres ressources informatiques. Ces ressources sont fournies comme un service externe payant via Internet

La Figure 1.2 représente les différents services du Cloud :

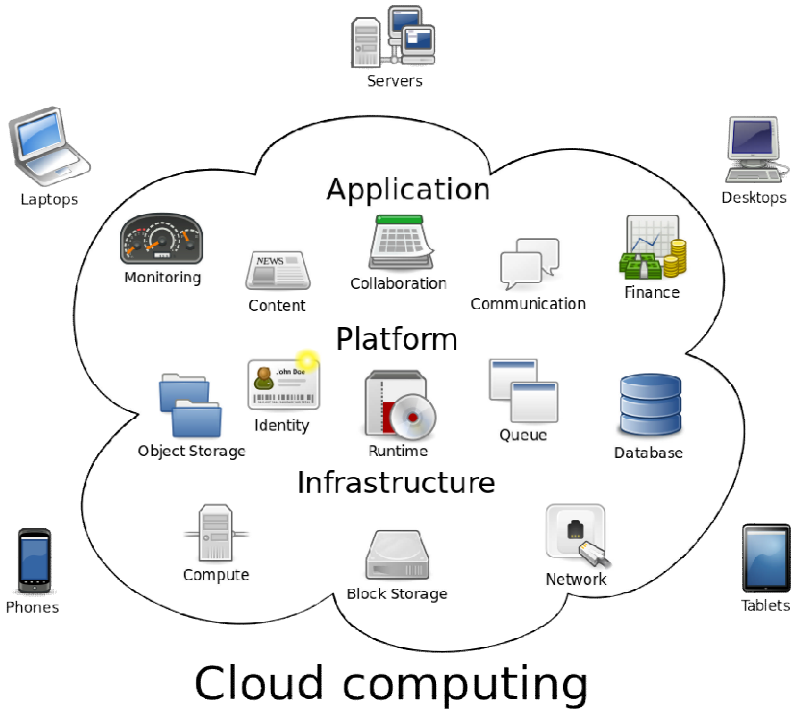

Figure 1.2: Services du Cloud.

Beaucoup de chercheurs sont confus au sujet de ce que le Cloud Computing est. Selon le laboratoire de **Barkley RAD** de l'université de Californie, Le Cloud Computing fait référence aux services fournis par les centres de données, à savoir les applications livrées en tant que services sur Internet, le matériel et le logiciel des systèmes, ces services en soi ont été appelés auparavant Software as a Service (SaaS). On qualifie un Cloud de public quand il est mis à la disposition du grand public. Le terme Cloud privé est utilisé pour les centres de données internes d'une entreprise ou d'une organisation qui ne sont pas mis à la disposition du grand public. Les personnes peuvent être des utilisateurs ou des fournisseurs de SaaS. [5]

Selon Buyya, Le Cloud Computing est un système parallèle et distribué d'un ensemble d'ordinateurs interconnectés et virtualisés, qui sont dynamiquement provisionnés et qui sont présentés comme une ressource ou plusieurs ressources informatique unifiées basées sur des accords SLA (Service Level Agreement) établis par voie de négociations entre les fournisseurs de services et les consommateurs. [6]

Une autre définition citée par les deux chercheurs Stanoevska-Slabeva et Wozniak [7] est:

- Le Cloud Computing est un nouveau paradigme informatique.

- Lorsque les services sont offerts par un fournisseur indépendant ou à des clients externes; les ressources de l'infrastructure (matériel, stockage et logiciel système) et les applications sont fournies de manière X-as-a-Service. Cloud Computing est basé sur des modèles d'affaires à usage payant.
- Les principales caractéristiques des Clouds sont la virtualisation et l'évolutivité dynamique à la demande.
- L'informatique utilitaire et le SaaS sont fournis de manière intégrée, même si l'informatique utilitaire peut être consommée séparément.
- Les services Cloud sont consommés via un navigateur Web ou via une API définie.

Cependant, le National Institute of Standards and Technology (NIST) a donné une brève définition qui reprend les principes de base du Cloud. Il le définie comme étant un modèle pratique et ubiquitaire et à la demande de joindre un réseau pour accéder à un ensemble partagé de ressources informatiques configurables (réseaux, serveurs, stockages, applications et services). Ces ressources peuvent être rapidement mobilisées et mises à disposition en minimisant les efforts de gestion et les contacts avec le fournisseur de services.

Le NIST précise que le Cloud Computing est composé de cinq caractéristiques essentielles, trois modèles de services et quatre modèles de déploiement [8], Figure 1.3.

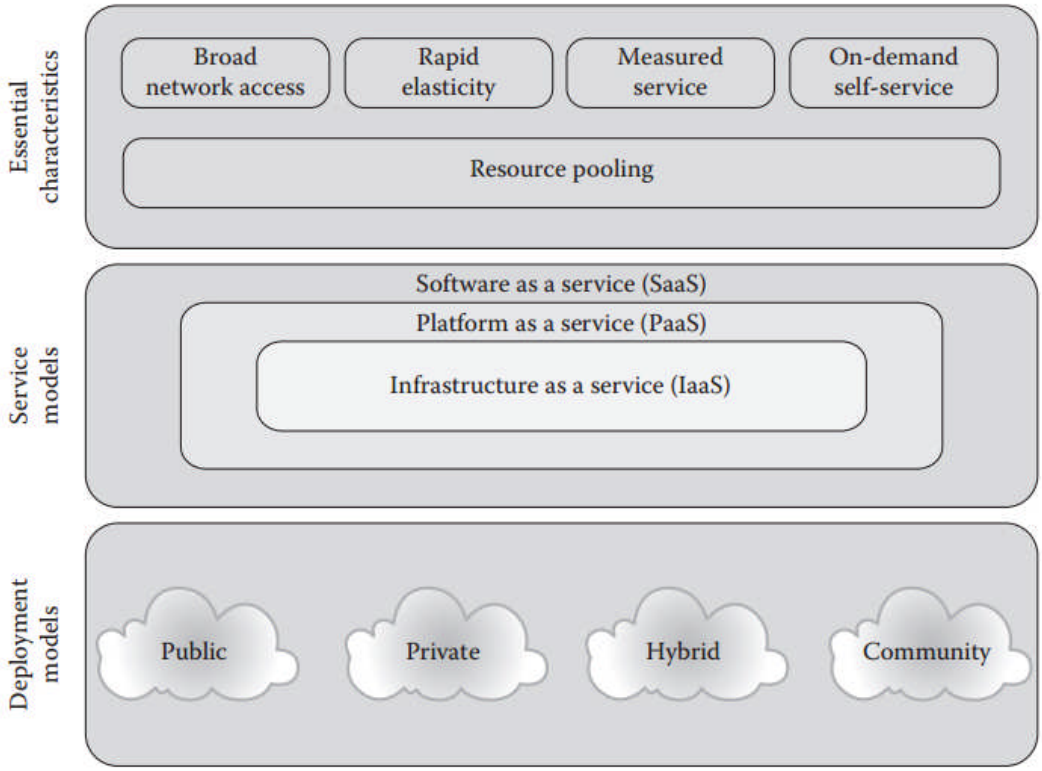

Figure 1.3: Éléments fondamentales du Cloud Computing.

1.3. Caractéristiques:

Le modèle Cloud Computing se différencie par les cinq caractéristiques essentielles suivantes :

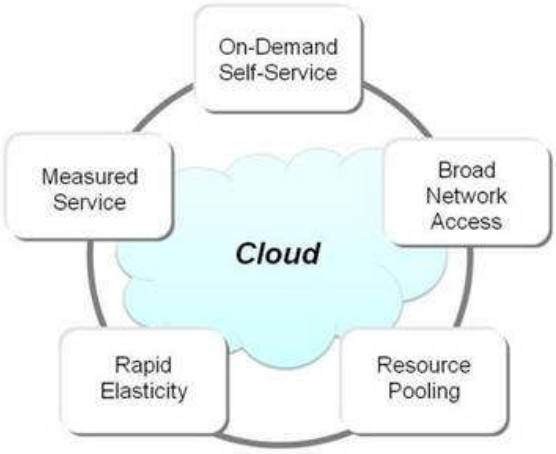

Figure 1.4:Caractéristiques du Cloud Computing.

- **Ressources à la demande:**

Les ressources informatiques (serveur, réseau, capacité de stockage, environnement d'exécution, application et performances de calcul) peuvent être allouées par un utilisateur de façon unilatérale, automatique, sans nécessité d'interaction humaine avec chaque fournisseur de services et au besoin.

- **Large accès réseau:**

Les ressources du Cloud Computing sont disponibles à travers le réseau et accessibles via des mécanismes standards qui favorisent leurs utilisations à partir des appareils clients hétérogènes (ordinateurs portables, téléphones, tablettes).

- **Mutualisation des ressources:**

Les ressources informatiques du fournisseur Cloud Computing sont mutualisées pour servir plusieurs clients en utilisant un modèle multi-tenant. Ces ressources, physiques ou virtuelles, ne sont pas dédiées à un client spécifique, elles sont allouées et libérées dynamiquement selon la demande du consommateur. Généralement, l'utilisateur n'a ni le contrôle ni la connaissance de l'emplacement exact des ressources allouées, dans certains cas, il peut choisir l'emplacement géographique à un haut niveau (par pays, continent ou Datacenter).

- **Élasticité rapide:**

Les ressources sont allouées et libérées d'une façon idéalement élastique et automatique, pour s'adapter rapidement à la demande qu'elle soit croissante ou décroissante. Pour le

consommateur, les ressources disponibles à l'allocation apparaissent illimitées et peuvent s'allouer à tout moment.

- **Services mesurés:**

Toutes les ressources allouées peuvent être surveillées et contrôlées afin de mesurer leurs consommations avec un niveau d'abstraction approprié selon le type de service (stockage, temps de calcul, bande passante etc.). L'usage des ressources peut être monitoré, contrôlé et reporté, ce qui assure une transparence pour le fournisseur et le consommateur de service.

1.4. Limites des approches traditionnelles:

Au cours de ces dernières décennies, Les technologies de l'information (TI) ont changé le monde entier. Peu de temps avant, les entreprises faisaient leurs affaires simplement à la main, avec le téléphone ou le fax. Peu à peu, les systèmes informatiques ont intrusionné des processus manuels et ont commencé à les automatiser. Le stylo et le papier ont été remplacés par des communications et des données numériques, et même les services de téléphone et de télécopie ont commencé à être gérés par des ordinateurs.

À l'heure actuelle, les entreprises utilisent ces systèmes pour réaliser presque toutes les tâches liées à leur fonctionnement. Les individus pareillement, dépendent fortement de ces systèmes informatiques pour leurs activités quotidiennes.

L'informatique à des facteurs très importants et la vie serait inimaginable sans accès facile et permanent à ces systèmes informatiques, Par ailleurs, les entreprises et les utilisateurs rencontrent beaucoup de difficultés en termes de simplicité d'utilisation, de la gestion et de la budgétisation.

1.4.1. Perspectives d'entreprise :

Les entreprises ont toujours été des utilisateurs précieux de l'informatique depuis sa création, mais leur réussite dépend de la gestion des contraintes liées à cet usage, à commencer par les investissements pour la mise en place de l'infrastructure initiale, l'installation des systèmes d'exploitation (SE), les pilotes de périphériques, la gestion des routeurs, les pare-feux, etc. en plus du budget dédié aux équipes de maintenance et de configuration.

Les équipements installés deviennent obsolètes suite au développement rapides du matériel, d'autre demeurent insuffisants par rapport aux besoins et doivent être remplacés (exemple : la capacité des disques dures). Ces équipements révolus représentent des pertes pour les entreprises.

1.4.2. Perspectives individuelles :

Les utilisateurs sont également confrontés à de nombreuses difficultés avec l'informatique traditionnelle, et se trouvent continuellement dans la nécessité de mettre en place un système physique est chère surtout avec le développement continu de nouveaux matériels toujours plus puissants et plus performants. Par méconnaissance ou manque d'expérience, ils sont menés à

dépenser plus d'argent sur des dispositifs dépassant leurs besoins, le cas pour certaines installations (SE, logiciels, etc.) et pour la maintenance. Un débutant utilisant un simple logiciel doit avoir des connaissances sur la Platform sur laquelle il travaille.

Les ordinateurs ne sont pas exploités parfaitement. Selon une étude présentée par le chercheur Thomas BEAUVISAGE (Oranges Labs, France) sur l'usage des ordinateurs dans la vie, La moyenne quotidienne d'utilisation réelle se situe plutôt autour de 2h51, ainsi que 63% de cette utilisation est consacré au Web [9], Figure 1.5.

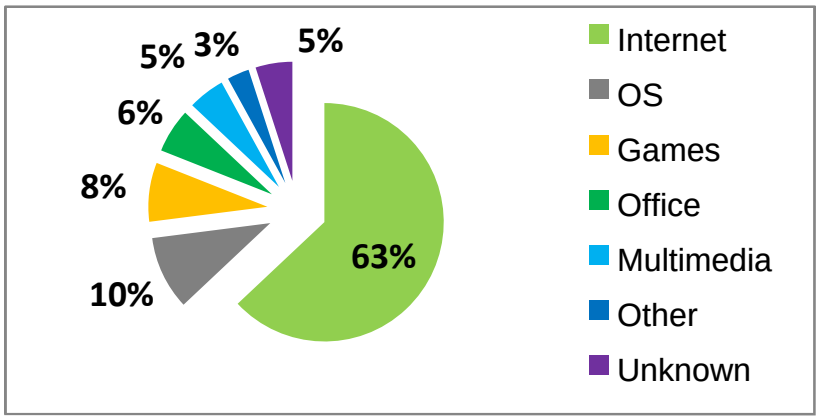

Figure 1.5: A quoi les utilisateurs passent-ils leur temps? [9].

Le Cloud Computing permet aux utilisateurs d'obtenir facilement des équipements informatiques, sans la mise en place d'une nouvelle infrastructure ou l'achat de nouveaux équipements et de logiciels sous licence, au contraire ils peuvent posséder le volume de stockage ou bien d'autres dispositifs suffisants pour réaliser leurs tâches (besoins).

Ce nouveau modèle est devenu possible grâce aux modèles informatiques avancés et aux technologies de communication et Web sophistiquées (en particulier l'internet à haute vitesse).

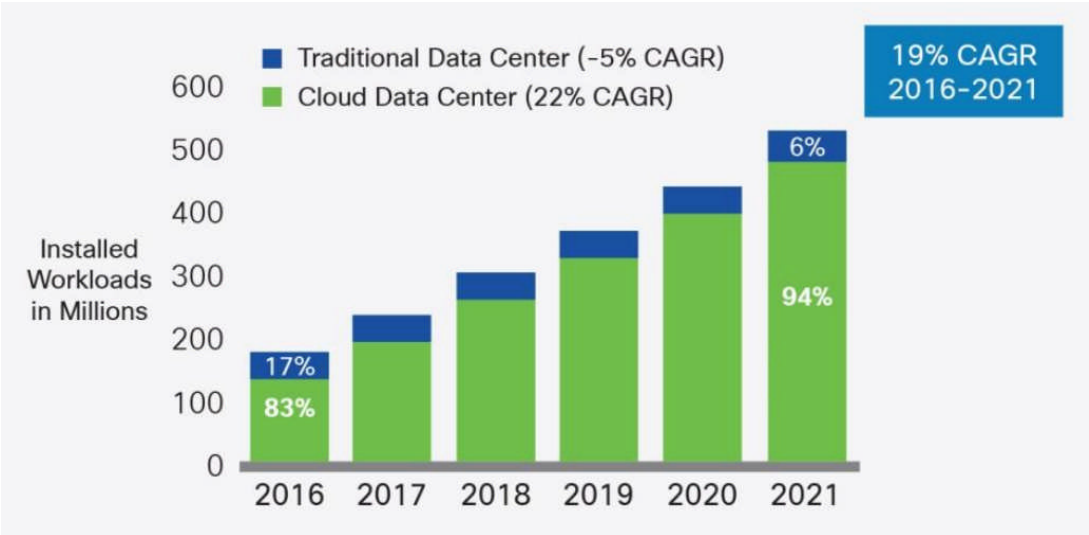

Figure 1.6: Répartition des charges de travail: 2016-2021 (4).

1.5. Evolution du Cloud Computing:

Le Cloud Computing n'est pas une innovation à part, ni une technologie unique, mais c'est une combinaison de plusieurs technologies.

Le Cloud Computing est apparu à la suite des recherches et des développements dans différents domaines informatiques, à savoir : l'évolution des systèmes distribués, l'émergence des techniques de virtualisation des ressources, le développement dans le domaine des services web et les technologies de communication, l'arrivée du paradigme SOA (Service Oriented Architecture) dans le développement des nouvelles applications, L'avancement dans le domaine de l'IA (Intelegence Artificielle) pour gérer automatiquement les infrastructures informatiques. Les progrès réalisés dans l'ensemble de ces domaines ont contribué à concrétiser le rêve du Cloud Computing, Figure 1.7.

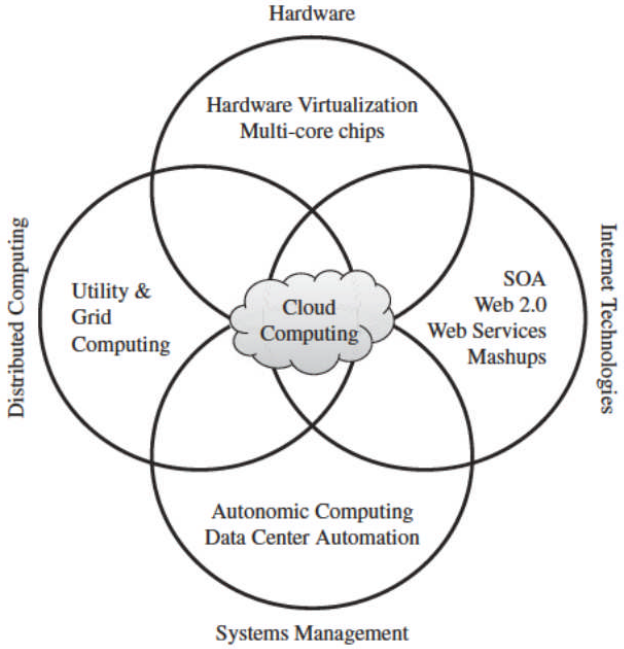

Figure 1.7: les technologies derrière l'évolution du Cloud Computing.

Les points suivants décrivent les technologies les plus importantes qui ont contribué à l'initiation de ce nouveau concept:

- **Informatique autonome (Autonomic Computing):**

L'informatique autonome fait référence aux systèmes informatiques intelligents qui peuvent se gérer sans aucune intervention humaine. Ils peuvent se reconfigurer automatiquement avec des conditions variables et se protéger de toutes défaillances techniques.

Ces systèmes peuvent prendre des décisions grâce à l'application des concepts de l'Intelligence Artificielle. Ils sont contrôlés par des politiques et des règles prédéfinies pour éviter toute intervention extérieure.

Le concept d'informatique autonome a été présenté par IBM en 2001. Quatre domaines sont définis, Figure 1.8:

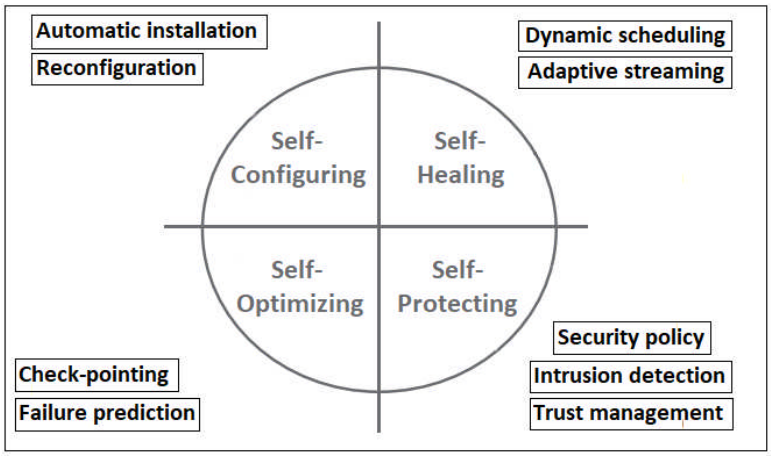

Figure 1.8: Propriétés d'un système autonome.

- L'auto-configuration est la capacité d'un système à être configuré en se basant sur des exigences sans aucune intervention extérieure.
- L'auto- récupération est la capacité de découvrir, de diagnostiquer et de corriger les erreurs.
- L'auto-optimisation est la capacité d'un système à contrôler automatiquement les ressources pour une utilisation optimale.
- L'autoprotection signifie la capacité d'identifier toute occurrence de comportement dangereux (comme: une infection virale, une attaque par déni de service, un accès non autorisé, etc.) et de prendre des mesures correctives pour rendre le système moins vulnérable [10].

- **SOA, Web services, Web 2.0, Mashup:**

L'architecture orientée services (Service Oriented Architecture, SOA) est un modèle d'interaction applicatif qui met en œuvre des Web services, Les Web services sont des programmes autonomes, auto-descriptifs et indépendants de la plate-forme, ils peuvent être invoqués sur Internet pour résoudre une activité particulière.

Le paradigme SOA a été créé en utilisant des technologies Web telles que XML (Extensible Markup Language), WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration). L'ensemble des services est

défini avec des interfaces connues et bien décrites qui peut communiquer entre elles, Ces interfaces sont spécifiées en XML et les services sont exprimés en WSDL.

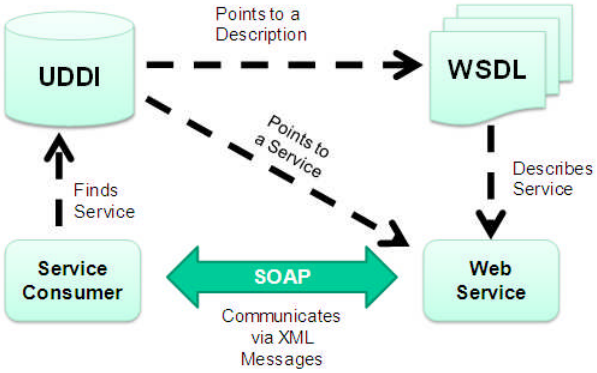

Figure 1.9: Web service architecture [11].

Web 2.0 est la deuxième génération du World Wide Web, il est apparu dès l'année 2002. Il consiste à regrouper les services Web à l'aide d'une nouvelle technique de programmation tel qu'AJAX (JavaScript asynchrone et XML) et RSS (Really Simple Syndication) ou REST (Transfert d'état REpresentational). AJAX est une technique qui permet la mise à jour des pages Web sans recharger la page complète. RSS distribue les nouvelles informations sur les pages Web d'une manière dynamique. Les données exposées par REST et RSS sont extraites par une technique appelée Mashup.

Le Mashup est une application Web qui peut combiner des données provenant de plusieurs sources Web dans une seule interface. Une application ou un service peut servir plusieurs fonctionnalités. L'idée derrière Mashup était de fusionner les meilleures applications Web disponibles ensemble sans effort dans une seule application pour la commodité des utilisateurs. Cela est fait en appelant ces applications ouvertes en utilisant leurs interfaces de programme d'application (API).

- **La virtualisation:**

La virtualisation est une technique permettant de créer une version virtuelle du système d'exploitation, du réseau, du processeur, du serveur ou des périphériques de stockage, etc,. C'est une solution intégrée pour augmenter l'utilisation des ressources dans un centre de données.

Actuellement, la virtualisation continue à porter essentiellement sur les serveurs. On observe toutefois l'émergence d'une stratégie globale de virtualisation des ressources de stockage et de gestion du réseau.

La grande majorité des environnements Cloud performants sont basés sur une infrastructure virtualisée. Cela fait déjà plusieurs années que la virtualisation est utilisée avec succès dans les centres de données en tant que stratégie de consolidation des serveurs. Elle utilisée de façon plus large

pour mettre en commun les ressources d'infrastructures, la virtualisation peut également fournir les éléments constitutifs élémentaires requis pour améliorer l'agilité et la flexibilité d'un environnement Cloud.

La virtualisation et le Cloud Computing sont utilisés de manière interchangeable, elle dématérialise les ressources informatiques (généralement sous forme de machines virtuelles), ainsi que les infrastructures de stockage et de connectivité réseau associées. Le Cloud détermine ensuite les modalités selon lesquelles ces ressources virtualisées sont allouées, distribuées et présentées. La virtualisation n'est pas indispensable pour créer un environnement Cloud, mais elle permet une extension des ressources dans des délais difficiles à atteindre avec des environnements non virtualisés [11].

- **Peer To Peer, Cluster Computing ET Grid Computing:**

Les systèmes P2P (peer to peer ou pair-à-pair) permettent à plusieurs ordinateurs de communiquer via un réseau en partageant simplement des objets (fichiers, flux multimédia continus (streaming), calcul réparti et téléphonie sur IP). Ces données peuvent être transférées directement entre deux postes connectés au réseau, sans transiter par un serveur central. Il permet ainsi à tous les ordinateurs (nœuds) de jouer directement le rôle de client et de serveur [13].

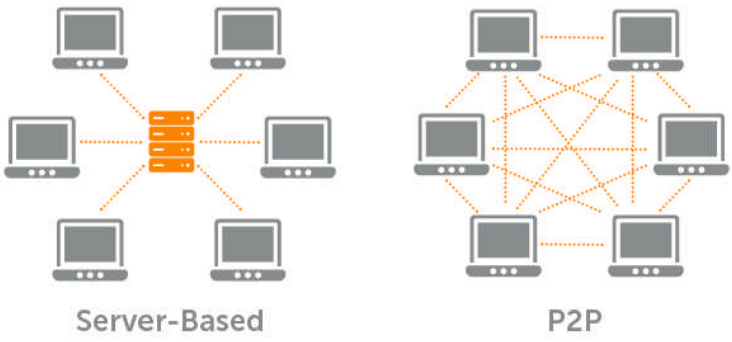

Figure 1.10: Architecture d'un réseau server-based et P2P.

Les Clusters (Les grappes de calcul) sont constitués de plusieurs nœuds (ordinateurs) connectés via un réseau et ils exécutent des tâches similaires. L'exécution d'une tâche peut être plus rapide car elle peut être distribuée et exécutée en parallèle sur plusieurs machines à l'intérieur du Cluster. L'ensemble des nœuds dans un cluster constitue un système unique.

Dans chaque Cluster, un ordinateur est affecté à contrôler les autres nœuds. Cet ordinateur particulier (ou nœud) est nommé Cluster head. A l'apparition des tâches informatiques correspondantes son rôle est de diviser et de répartir les tâches entre les différents nœuds du même cluster.

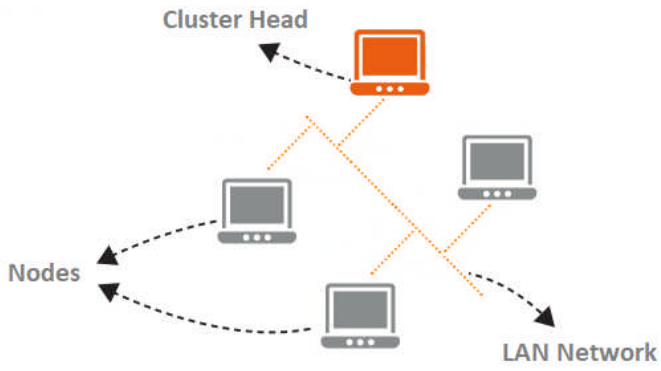

Figure 1.91: Modèle de Cluster Computing.

Dans les systèmes actuels, plusieurs Clusters (construits pour effectuer différents types de fonctionnalités) sont reliés entre eux sous un réseau LAN. Dans un tel environnement, lorsqu'un travail particulier apparaît, le Cluster head le divise sur les autres Clusters correspondants (désignés pour ces travaux) pour une exécution plus rapide. La distribution et l'attribution du travail dépendent de sa nature.

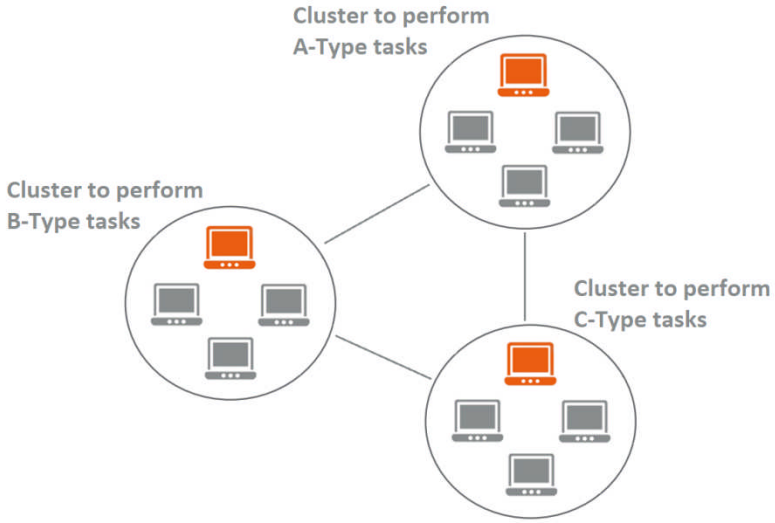

Figure 1.102: le modèle des nouveaux Clusters.

Le Cluster Computing conduit à des systèmes informatiques plus puissants et plus fiables, mais il soulève des inquiétudes concernant l'homogénéité des systèmes et la dépendance à la gestion de Cluster head. La performance d'un tel système dépend largement de l'efficacité et de l'accessibilité du Cluster head. Cette dépendance augmente la possibilité d'une défaillance, ce qui implique la nécessité de concevoir un nouveau modèle pour éliminer ces problèmes.

Le Grid Computing (Grille informatique) est une nouvelle architecture introduite en 1990, il consiste à donner la même priorité à tous les nœuds, qui peuvent passer et exécuter des tâches sans qu'un ordinateur particulier ne possède le rôle du Head. Cette architecture peut être construite avec des systèmes informatiques hétérogènes (Systèmes avec diverses configurations matérielles).

Le Grid Computing est très avantageux, d'une part de l'hétérogénéité des systèmes qui augmente le nombre des ressources, et d'autre part de la répartition des nœuds sur différents domaines géographiques en utilisant des interconnexions pour les interactions; à l'opposé des Clusters où les groupes d'ordinateurs se situent dans un seul endroit et ne sont connectés qu'avec un réseau LAN.

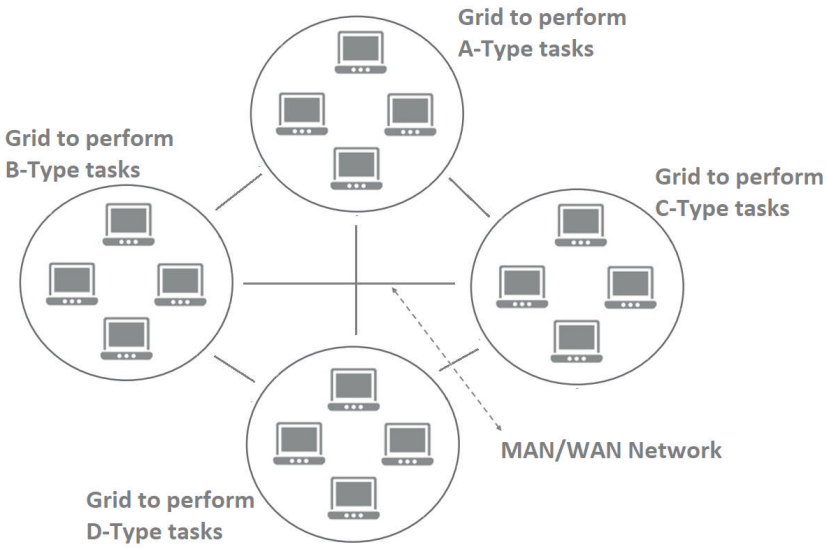

Figure 1.113 : Modèle de Grid Computing .

Dans une grille, la puissance de calcul et la capacité de stockage sont pratiquement illimitées puisque toutes les ressources de la grille peuvent être mobilisées en cas de besoin. Il permet, sans effort, de mettre en production intensive une application développée localement et de mieux partager les ressources disponibles.

Caractéristiques	Cluster Computing	Grid Computing
Gestion des ressources	Centralisé	Distribué
Hétérogénéité	Homogène	Hétérogène
Localisation	Centralisé	Centralisé/Distribué
Architecture des systèmes	Homogène	Hétérogène
Type de connexion réseau	LAN	LAN/MAN/WAN
Sécurité	Haute	Moyenne
Coût	Trés cher	Cher

Tableau 1.1 : Comparaison entre le Cluster et Grid computing.

- **Informatique utilitaire:**

Avec la popularité et l'utilisation croissante de Grid Computing, les grandes installations de grilles ont rencontré de nouveaux problèmes, tel que les demandes excessives de ressources. Initialement, la gestion des ressources n'assure pas un accès équitable aux ressources dans de nombreux systèmes. Les paramètres traditionnels (débit, temps d'attente, latence) ne permettaient pas de garantir les exigences les plus subtiles des utilisateurs, sans réelle flexibilité et souplesse en matière de ressources, de même les utilisateurs avec des travaux urgents ne sont pas servis immédiatement.

L'utility Computing est simplement une délocalisation d'un système de calcul ou de stockage. Il présente un modèle d'affectation des ressources à la demande et facturation des utilisateurs en fonction de leur usage. Le Cloud Computing peut être perçu comme une réalisation de l'informatique utilitaire. Il adopte un système de tarification basé sur l'utilité, entièrement pour des raisons économiques. Avec l'approvisionnement des ressources à la demande et le paiement à l'usage, les fournisseurs de services peuvent maximiser l'utilisation des ressources et minimiser leurs coûts d'exploitation [3].

- **L'apparition du Cloud Computing :**

Le Cloud Computing implémente beaucoup de technologies tel que l'architecture orientée services, les services web et la virtualisation. Le Cloud est souvent confondu avec plusieurs paradigmes informatiques tels que l'Autonomic Computing, l'Utility Computing et le Grid Computing, compte tenu des aspects qu'ils partagent.

Avec l'apparition du Cloud Computing Beaucoup de bénéfices du Cloud accablé l'ancien concept de Grid Computing :

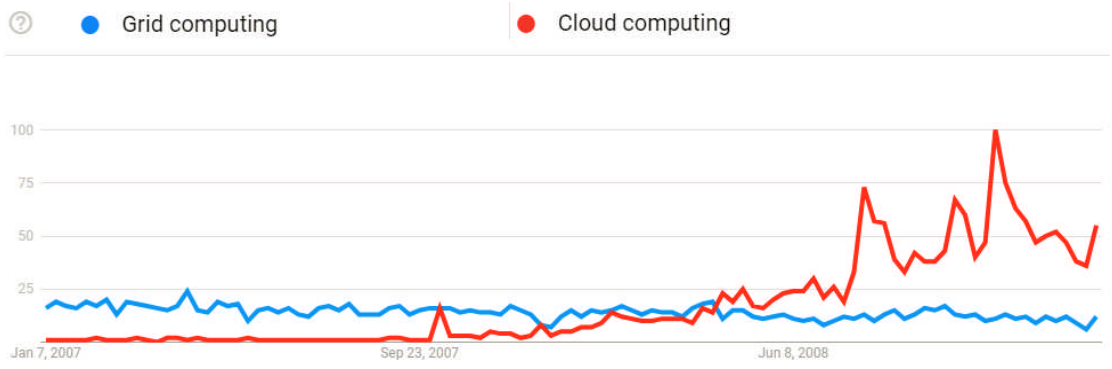

Figure 1.14: les recherches du "Cloud Computing" par rapport "Grid Computing" sur google.com, source : Google Trends.

L'avantage majeur du Cloud est la scalabilité, contrairement au Grid Computing, les ressources informatiques dans le Cloud peuvent être ajoutées en temps réel pour répondre à la demande, cela est devenu possible grâce à la virtualisation des ressources. Le tableau suivant représente la comparaison entre le Grid Computing et le Cloud Computing :

Paramètres	Grid Computing	Cloud Computing
But principale	Partage de resources	Utilisation du service
Possession/ Gestion	Distribué	Centralisé/ plusieurs administrateur
Architectures du matérielle	Hétérogène	Hétérogène
Systèmes d'exploitation	Tous les systèmes d'exploitation standards	Un hyperviseur (VM) sur lequel plusieurs systèmes d'exploitation s'exécutent.
Scalabilité	Normal	Haute
Virtualisation	Virtualisation des données Et des ressources informatiques	Virtualisation de plates-formes matérielles et logicielles
Type de service	CPU, réseau, mémoire, périphérique, stockage, ...	IaaS, PaaS, SaaS, ou Xaas

Tableau 1.2: Différences entre le Cloud et le Grid Computing.

1.6. Modèles de déploiement :

Selon la définition du Cloud Computing donnée par le NIST, il existe quatre modèles de déploiement des services du Cloud, à savoir : Cloud privé, Cloud communautaire, Cloud public et Cloud hybride, Figure 1.15 représente les différents modèles du Cloud :

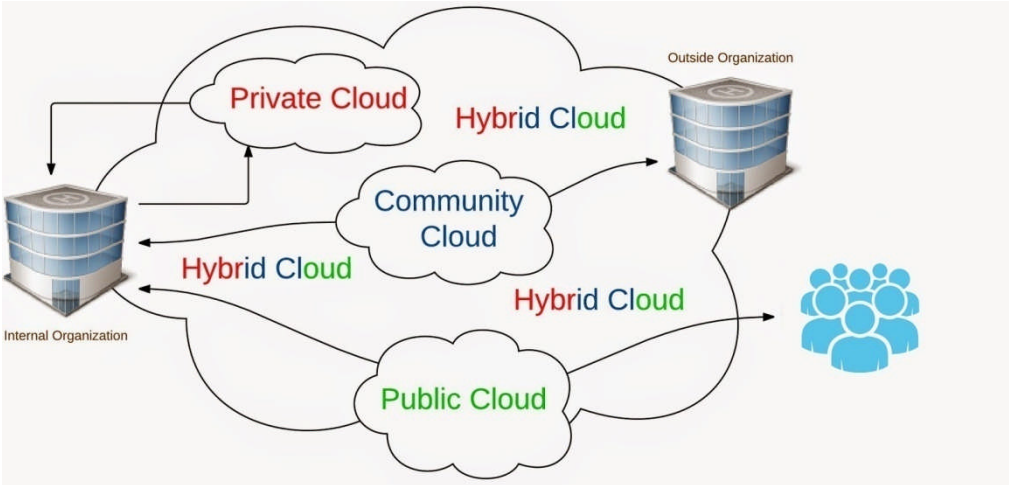

Figure 1.15: les modèles de déploiement du Cloud.

1.6.1. Cloud public :

Un Cloud public permet aux mêmes ressources d'être accessibles à tout le publique. Une caractéristique fondamentale du Cloud public est qu'il est destiné à servir une multitude d'utilisateurs. Le consommateur peut être un utilisateur individuel ou un groupe de personnes représentant une organisation ou une entreprise.

Les consommateurs dans ce modèle sont totalement libres de toute responsabilité liée à l'administration de l'infrastructure et aux problèmes liés à la gestion du système, leur contrôle sur le Cloud est infime.

Pour fournir des services aux consommateurs dans un Cloud public, les centres de données sont gérés par certains fournisseurs de services informatiques (exemple: Google, Amazon Ec2, IBM Smart Cloud et SalesForce.com etc.).

1.6.2. Cloud privé :

L'ensemble des ressources d'un Cloud privé est exclusivement mis à disposition d'une entreprise ou une organisation unique. Le Cloud privé peut être géré par l'entreprise elle-même (Cloud privé interne) ou par une tierce partie (Cloud privé externe). Les ressources d'un Cloud privé se trouvent généralement dans les locaux de l'entreprise ou bien chez un fournisseur de services. Dans ce dernier cas, l'infrastructure est entièrement dédiée à l'entreprise et y est accessible via un réseau sécurisé (de type VPN) [3].

Dans un Cloud privé, les consommateurs ont la plupart des avantages du Cloud Computing et peuvent toujours garder le contrôle de l'environnement, contrairement au Cloud public où ils n'en ont aucun.

1.6.3. Cloud communautaires:

L'infrastructure d'un Cloud communautaire est partagée par plusieurs consommateurs ou organisations indépendantes appartenant à la même communauté, et qui ont les mêmes besoins, ce qui réduit le coût.

Les ressources du Cloud peuvent résider dans les locaux d'un membre de la communauté ou être situées dans un endroit externe. Comme le Cloud privé, ce modèle peut également être géré par une ou plusieurs organisations participantes (de la communauté) ou auprès d'un fournisseur informatique externe.

Ce type de déploiement peut être considéré comme une forme généralisée du Cloud privé; Un Cloud privé est accessible uniquement par un consommateur, un Cloud communautaire est utilisé par plusieurs consommateurs de la communauté. Le but du Cloud communautaire est d'offrir et les avantages du Cloud privé comme le niveau de sécurité et ceux du Cloud public comme le coût.

1.6.4. Cloud hybride :

L'infrastructure d'un Cloud hybride est une composition de deux ou trois Clouds (privé, communautaire ou public) qui restent des entités uniques mais sont liés ensemble par une technologie standardisée ou exclusive.

Ce modèle de déploiement aide les entreprises à stocker les applications et les données critiques sur le Cloud privé ou communautaire pour les sécuriser, tout en permettant la réduction des coûts en conservant les données et les applications partagées sur le Cloud public. La Figure 1.16 représente une comparaison entre les différants modèles du Cloud.

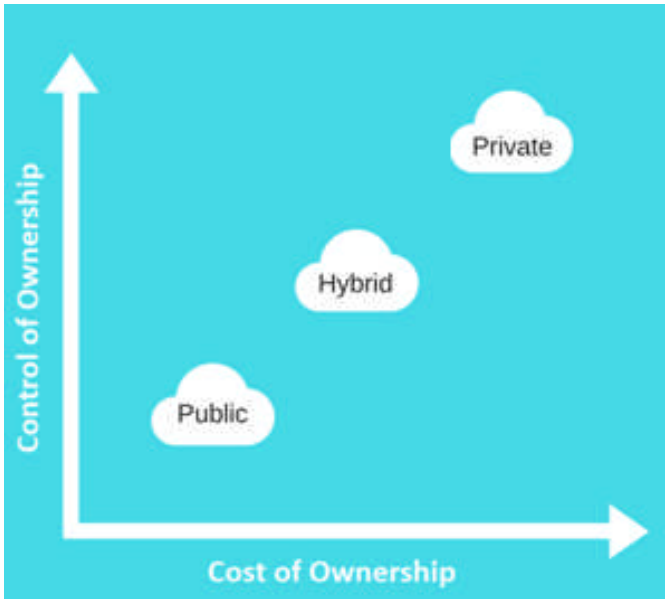

Figure 1.126: Comparaison entre les différants modèles.

1.7. Type de services du Cloud Computing:

Le NIST a classé les services du Cloud Computing selon le type de service offert à l'utilisateur en trois catégories principales : l'infrastructure en tant que service (IaaS), la plate-forme en tant que service (PaaS) et le logiciel en tant que service (SaaS).

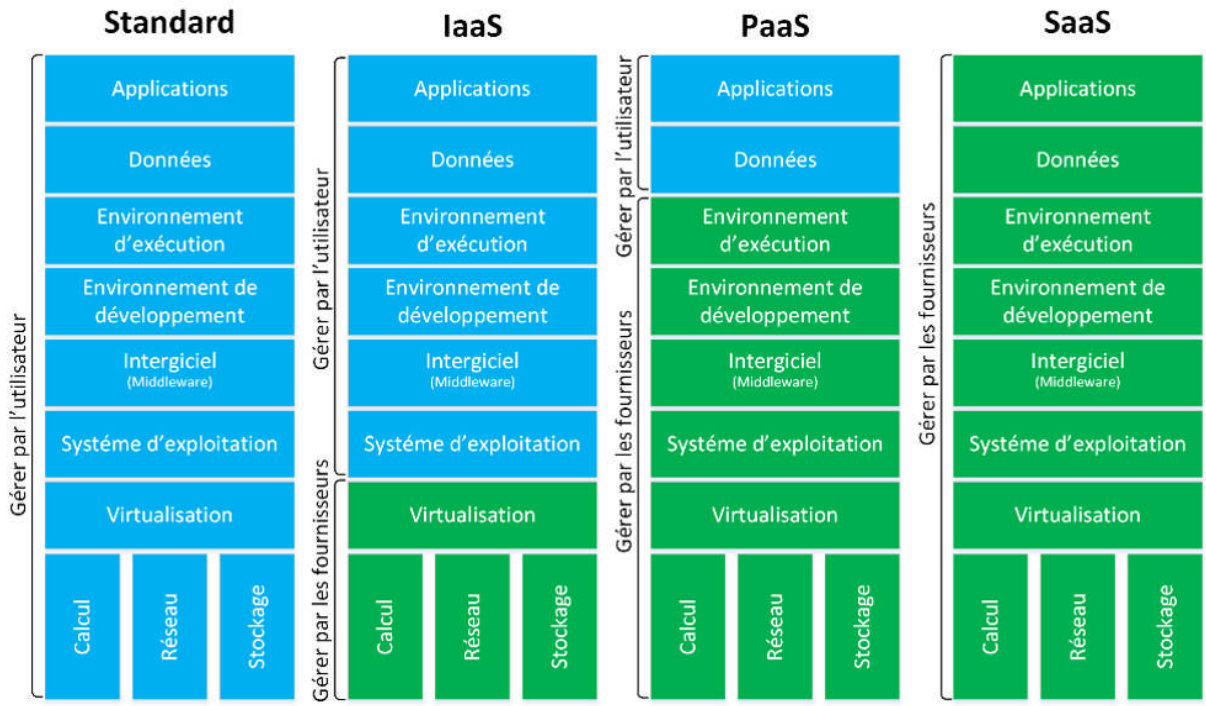

Figure 1.137: Types de services Cloud Computing.

1.7.1. Software as a Service (SaaS):

Ce modèle de service est caractérisé par l'utilisation d'une application partagée qui fonctionne sur une infrastructure Cloud, l'utilisateur accède à ces applications à travers différentes interfaces (Navigateur, client léger, etc.), il ne gère pas l'infrastructure sous-jacente incluant le réseau, les serveurs, les systèmes d'exploitation et le stockage. L'utilisateur n'a aucun contrôle sur les fonctions de l'application à l'exception d'un paramétrage de quelques fonctions utilisateurs limitées. Ce modèle facilite la maintenance des logiciels pour les clients et leur développement pour les fournisseurs. Les logiciels de messagerie accessibles à travers un navigateur tel que Gmail ou Yahoo mail est le meilleur exemple de ce type de services.

1.7.2. Platform as a Service (PaaS):

Dans ce modèle, l'utilisateur a la possibilité de créer et de déployer ses propres applications sur une infrastructure Cloud PaaS, et ce en utilisant les langages et les outils du fournisseur. L'utilisateur ne gère pas l'infrastructure Cloud sous-jacente (réseaux, serveurs et stockage) mais il a le contrôle sur les applications déployées et il peut aussi configurer l'environnement d'hébergement. Exemples des fournisseurs PaaS : Google App Engine, Windows Azure, Engine Yard, Force.com, Heroku et MTurk.

1.7.3. Infrastructure as a Service (IaaS):

Les services Cloud Computing de type IaaS correspondent aux ressources d'une infrastructure offertes à la demande. Ces ressources peuvent être des ressources de calculs, de stockage ou de réseau et d'autres ressources indispensables (partage de charge, pare-feu, cache).

L'utilisateur a la possibilité de déployer n'importe quel type de logiciel incluant les systèmes d'exploitation ne peut ni gérer ni contrôler l'infrastructure Cloud sous-jacente mais il a le contrôle sur les systèmes d'exploitation, le stockage et les applications. Il peut aussi choisir les caractéristiques principales des équipements réseau comme le partage de charge, les pare-feu, etc.

Exemples des fournisseurs IaaS : Amazon EC2, VPC, IBM Blue Cloud, Eucalyptus, FlexiScale, Joyent, Rackspace Cloud, etc.

La nomenclature « x as a Service » est utilisée pour caractériser les services et les ressources Cloud Computing et il existe plusieurs sous-ensembles d'un ou de plusieurs des trois types de base (IaaS, PaaS et SaaS) dont on peut citer quelques abréviations:

- **Security Management-as-a-Service (SECaaS) :** Les consommateurs de services Cloud peuvent déléguer les responsabilités de tous les problèmes de sécurité de leurs environnements informatiques à un fournisseur Cloud en particulier. Exemple : Cisco, McAfee, Symantec, etc.
- **Identity Management-as-a-Service (IDaaS) :** La gestion d'identité fait partie des services Cloud, mais certain nombre de fournisseurs la proposent en tant que service à part.

- **Storage-as-a-Service :** Dans le type IaaS, les fournisseurs offrent le stockage en tant que partie importante de leurs services. Nombreux fournisseurs de Cloud comme Amazon et Rackspace proposent le stockage indépendamment des autres services.
- **Database-as-a-Service (DbaaS) :** les services de base de données sont offerts sous la couche PaaS. Mais, plusieurs fournisseurs ont mis au point une solution de Cloud Computing exclusive pour la gestion des bases de données, appelée Database-as-a-Service (DBaaS). Elle propose une plate-forme unique et dotée d'une fonctionnalité à la demande et en libre-service, où même les non-administrateurs de base de données peuvent facilement parvenir à leurs demandes.
- **Backup-as-a-Service (BaaS) :** la Sauvegarde et la récupération des données est un service spécialisé que certains fournisseurs offrent aux utilisateurs en cas de fuite ou perte de données.
- **Desktop-as-a-Service (DaaS) :** un utilisateur peut accéder aux services de Cloud via plusieurs dispositifs (PC, tablette, etc.), mais l'environnement de son bureau n'est pas le même. Les fournisseurs (DaaS) délivrent des environnements de bureau personnalisés aux utilisateurs en tant que service.
- **Monitoring-as-a-Service (MaaS):** Ce type de service est utilisé pour superviser (surveiller) l'état d'une application, de stockage, de réseau, etc.

1.8. Avantages du Cloud Computing:

L'utilisation du Cloud Computing présente de nombreux avantages :

- **Réduction des coûts d'infrastructure:** Avec le Cloud Computing, il n'est plus requis de mettre en place une infrastructure entière ou d'augmenter le nombre de serveurs physiques proportionnellement aux charges. Toutes les ressources nécessaires peuvent être allouées facilement et avec un coût à l'utilisation afin de répondre à des montées de besoins et en temps réel.
- **Réduction de la responsabilité de gestion des systèmes:** La gestion et la maintenance des serveurs sont prises en charge par le fournisseur de service Cloud. Ce qui permet donc d'alléger considérablement les tâches des utilisateurs.
- **Mises à jour logicielles automatiques :** La mise à jour des applications est systématique, et le fournisseur décharge les clients de toute responsabilité de maintenance. Ainsi les utilisateurs de service n'ont pas besoin d'acheter aucun logiciel ou licence.
- **La flexibilité des ressources :** L'utilisateur peut augmenter ou réduire la capacité de son infrastructure sans investissement majeur et dans un délai réduit.
- **Puissance de calcul et stockage illimités:** contrairement aux grilles, les ressources dans un Cloud peut être ajouté à tout moment ce qui met à dispositions des ressources de grandes capacités.
- **Fiabilité :** le Cloud Computing garantie aux utilisateurs la qualité de service (QoS), l'équilibrage de la charge, la sauvegarde et la récupération des données.

- **Commodité de l'accès :** l'utilisateur peut à tout moment et à partir de n'importe quel appareil se connecter à ses applications et ces données.
- **La sécurité des données :** La sécurité des données est le principal frein d'adoption du Cloud Computing. Dans ce contexte, les fournisseurs garantissent aux utilisateurs un très haut degré de sécurité des données avec le chiffrement des données, la surveillance logicielle et la sécurisation des lieux de stockage (Centres de données).
- **Partage de données :** Les données peuvent être partagées, puisque tout utilisateur du Cloud Computing peut aisément rendre disponibles ses données à un ou plusieurs autres utilisateurs du Cloud Computing. Il est donc possible de créer une plateforme virtuelle collaborative en un temps record.
- **informatique verte (Green Computing) :** Les infrastructures gérées en interne sont souvent sous-utilisées, alors que l'infrastructure d'un Cloud mutualise l'ensemble de ressources pour un grand nombre de consommateurs. Elle permet alors de minimiser le nombre des équipements et d'augmenter le taux son utilisation.

1.9. Obstacles du Cloud Computing:

Le Cloud Computing n'a pas que des avantages, plusieurs obstacles et désavantages sont abordés dans [3] .Parmi ces obstacles, il y a :

- Disponibilité du service.
- Confidentialité et auditabilité des données.
- Chiffrement des données.
- Imprévisibilité des performances.
- Scalabilité de stockage.
- Nécessité d'un accès réseau constant.
- Mauvais fonctionnement avec les connexions à basse vitesse.
- Risque d'engorgements lors des transferts de données.
- Problème d'interopérabilité.
- Problème de portabilité.
- Des contrats de service SLAs non normalisés.

1.10. Challenges de recherche dans l'environnement Cloud:

Même si certaines des caractéristiques essentielles du Cloud Computing ont été réalisées par des efforts commerciaux et universitaires, de nombreux problèmes existants n'ont pas été pleinement pris en compte, et d'autres nouveaux défis continuent d'émerger [14].

- **Implémentation des politiques de sécurité :**

L'implémentation des politiques de sécurité est la tâche la plus difficile pour les fournisseurs, ils comprennent des politiques de gestion, des politiques réglementaires qui sont liées à la conformité aux normes, des politiques informatives qui éduquent les parties prenantes internes et externes de l'entreprise, etc. [15].

- **Gestion de Virtualisation :**

La virtualisation décrit une technologie dans laquelle les applications, les systèmes d'exploitation ou le stockage de données est séparé du matériel. Le logiciel émule le matériel à partir du système d'exploitation (Linux, Windows, etc.) ou un système d'exploitation dédié tel que VMware.

La virtualisation est désormais un élément essentiel dans le Cloud, elle constitue la colonne vertébrale de l'infrastructure en tant que service (IaaS), ce qui soulève la question sur les risques de sécurité liés à la virtualisation. [15].

- **Sécurité de données et confidentialité :**

Dans le Cloud Computing, les données doivent être transférées entre les dispositifs de l'utilisateur et les Datacenter des fournisseurs de services de Cloud Computing, ce qui les rendra cible facile pour les pirates. 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 [3].

1.11. Conclusion :

Dans ce chapitre nous avons présenté les notions fondamentales du Cloud Computing, ses enjeux, ses évolutions et son utilité ainsi que la technologie qui la constitue. Nous avons étudié les trois services principaux, sur lesquels le Cloud Computing repose: applicatif, plateforme, infrastructure, et qui ont donné naissance aux fameux SaaS/PaaS/IaaS. Enfin nous avons présente les différents avantages et inconvénient du Cloud Computing, et les challenges de recherche dans ce domaine.

Chapitre II

Sécurité dans le Cloud Computing.

2.1. Introduction:

Cloud Computing est une révolution économique et technologique, dans lequel les ressources informatiques sont fournies en tant que service via internet. En particulier, ces ressources peuvent être provisionnées de façon dynamique et libérées en fonction de la demande de service et avec un effort minimal de gestion. Il présente une meilleure solution pour gérer les données, les infrastructures, etc. Cependant, la sécurité des données en transit dans un Cloud public reste un challenge pour les fournisseurs de Cloud.

Les données stockées sur le Cloud sont exposées à des risques pouvant être la cible de plusieurs attaques réseau, victimes de pannes aléatoires, elles peuvent être perdues totalement ou partiellement, modifiées ou corrompues, rendues publiques alors qu'on voulait qu'elles restent confidentielles. Par conséquent, il est essentiel de faire face à ces attaques en vue d'améliorer l'utilisation et l'adoption de Cloud.

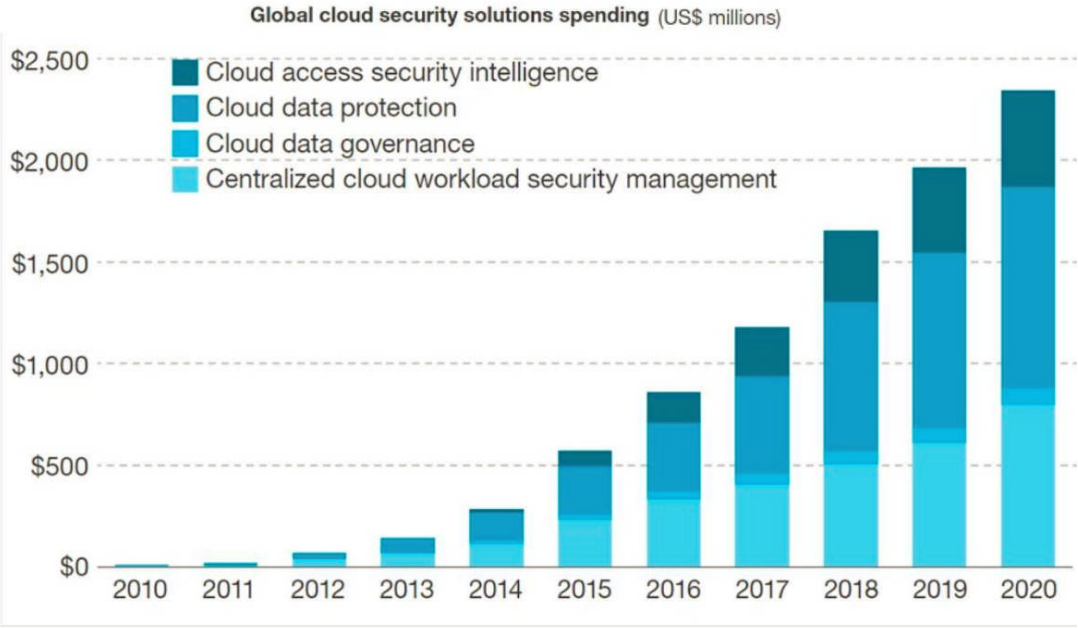

Figure 2.14: Dépenses mondiales de sécurité du Cloud [16].

Les solutions pour sécuriser les données sont basées sur la cryptographie et le codage : chiffrement des données stockées sur des supports externes pour garantir leur confidentialité et restreindre les accès ; signature et redondance pour leur intégrité.

2.2. Problèmes de sécurités dans le Cloud :

Les avantages du Cloud Computing sont aujourd'hui une évidence. Cependant, devant toutes les possibilités offertes par ce nouveau concept, il demeure des problèmes dans son adoption, les plus importants sont :

- **Multi-locataires :**

Les architectures multi-locataires facilite à faire en sorte qu'un logiciel soit capable de gérer un certain nombre de clients en une seule installation. Au lieu d'installer le logiciel une fois pour chaque client, ce dernier est capable de créer des environnements virtuels distincts pour chaque client de sorte à ce que de l'extérieur les autres environnements ne soient pas du tout visibles [17].

L'architecture multi-locataires a de nombreuses qualités quand il s'agit d'exploiter le logiciel. Mais cela implique des contraintes auxquelles on ne pense pas forcément au départ.

Lors d'une mise à jour, si une régression est introduite, elle affectera immédiatement l'ensemble des clients. C'est un risque qu'il faut accepter de prendre, mais il peut être largement réduit en travaillant sur la qualité du logiciel. De la même façon, il est préférable de mettre à jour un logiciel à un moment où il est peu utilisé. Un logiciel multi-tenant implique de mettre tout le système à jour en même temps. Si les clients sont localisés un peu partout dans le monde, il est difficile de trouver un moment où tout le monde dort.

- **Elasticité :**

L'élasticité correspond à la capacité d'adapter des ressources informatiques (calcul, stockage, etc.) à la volée en fonction des besoins applicatifs. En d'autres termes, le Cloud élastique adapte de manière autonome et très précise les ressources disponibles dans le système par un approvisionnement /dé-provisionnement automatique pour gérer les variations de charges et offrir le coût le plus optimal.

Concrètement, les méthodes des fournisseurs de Cloud actuels permettent une grande évolutivité, soit l'une des grandes promesses du Cloud. Cependant, cette évolutivité recèle souvent des faces cachées ou imprévues [18].

- **Perte de contrôle:**

Le Cloud utilise un modèle de localisation transparent par lequel il permet aux associations de ne pas connaître la zone de leurs services et données. Par la suite, le fournisseur peut avoir ses administrations de n'importe où dans le nuage. Pour cette situation, l'association peut perdre ses informations et potentiellement ne pas connaître la politique de sécurité mis en place par le fournisseur.

- **Fuites de données :**

La sécurité absolue n'existe pas. Des fuites de données peuvent survenir au sein des machines virtuelles, et les informations personnelles des clients ou les données confidentielles de l'entreprise peuvent ainsi être dérobées.

Malgré tout, cette perspective dissuade de nombreuses entreprises d'adopter le Cloud Computing. Malheureusement, les mesures pour empêcher les fuites ou le vol de données peuvent exacerber ces menaces [19].

- **Perte de données :**

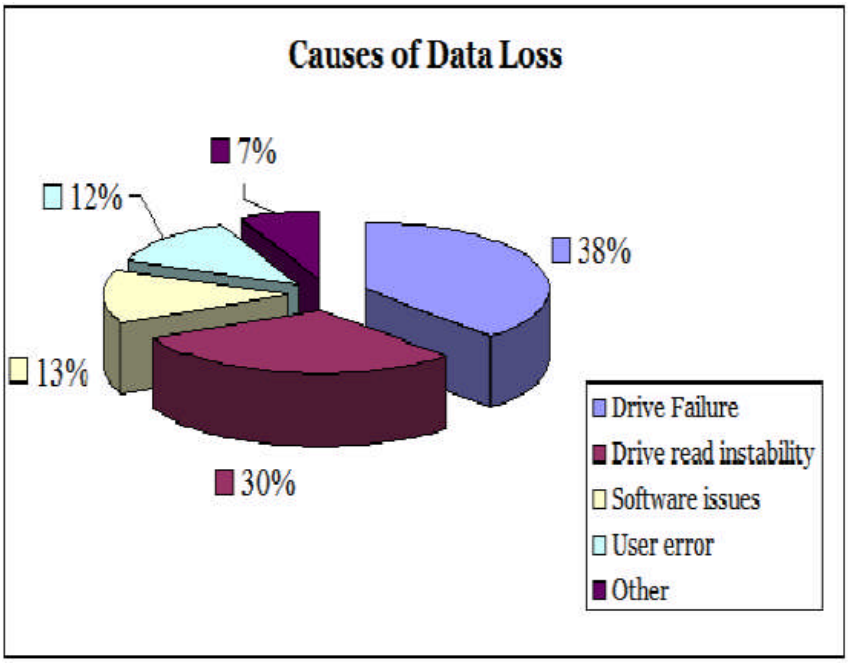

Figure 2.215:Causes de la perte de données [23].

Parfois, les données perdues à partir des serveurs Cloud ne sont pas dues à une cyberattaque. Les causes non-malveillantes de la perte de données incluent les catastrophes naturelles comme les inondations et les tremblements de terre, et les erreurs humaines simples, comme lorsqu'un administrateur du Cloud supprime accidentellement des fichiers.

Il est facile de sous-estimer le risque que quelque chose de mauvais arrive aux données en raison d'une erreur innocente. L'une des clés permettant d'atténuer la menace de perte de données non malveillante consiste à gérer de nombreuses sauvegardes sur des sites physiques situés dans différents emplacements géographiques [20].

- **Mauvaise utilisation des services du Cloud :**

Cracker une clé de chiffrement à l'aide d'un hardware limité peut prendre des années. Toutefois, les hackers ont eux aussi accès aux services du Cloud, et peuvent utiliser ces services pour cracker ces clés en quelques minutes seulement. Ils peuvent également utiliser ces serveurs pour lancer des malwares, des attaques DDoS, ou pour distribuer des logiciels piratés.

La capacité de stockage incomparable du Cloud a permis à la fois aux pirates informatiques et aux utilisateurs légaux d'héberger et de diffuser des logiciels malveillants, des logiciels illicites et d'autres ressources numériques. Ces dangers comprennent le partage d'émissions, de musique, d'enregistrements ou de livres volés. Les fournisseurs de services ont pour responsabilité d'éviter de tels abus, mais il est difficile de détecter des usages inappropriés.

- **Menaces internes:**

La plupart des employés sont dignes de confiance, mais un employé d'un service informatique ou d'un service métier peut disposer d'informations qui peuvent être faciles à acquérir par un cyber-attaquant externe qui sait manipuler l'ingénierie sociale comme la carotte financière.

Dans une étude citée dans [21], il avait été dit que 70% des attaques provenaient de l'intérieur. Dans le cadre du Cloud, cela veut donc dire que les utilisateurs doivent avoir les moyens de protéger leurs données contre les administrateurs système du service hébergé dans le Cloud. Cependant Le système du Cloud doit identifier les logiciels malveillants et les supprimer des serveurs virtualisés au sein du centre de données, ainsi que détecter et neutraliser les attaques émanant de comptes utilisateur dotés de privilèges.

- **Attaques externes :**

Les composants de sécurité telle que les pare feux ou les systèmes de détection d'intrusion, ne sont pas adaptés pour détecter les attaques distribuées. Ces attaques sont donc subdivisées en sous attaques afin d'être indétectable par un tel système de sécurité [22].

- **Injection de programmes malveillants :**

Les logiciels malveillants consistent généralement à voler des données confidentielles, telles que des noms d'utilisateur, des mots de passe, des informations et autres renseignements financiers utiles. Ces informations sensibles peuvent à servir ensuite de lancer d'autres attaques contre des personnes et des entreprises ou sont vendues à d'autres acteurs malveillants. Il existe plusieurs types de programmes malveillants tels que les chevaux de Troie, les keyloggers et les rootkits, etc.

- **Attaques par déni de service :**

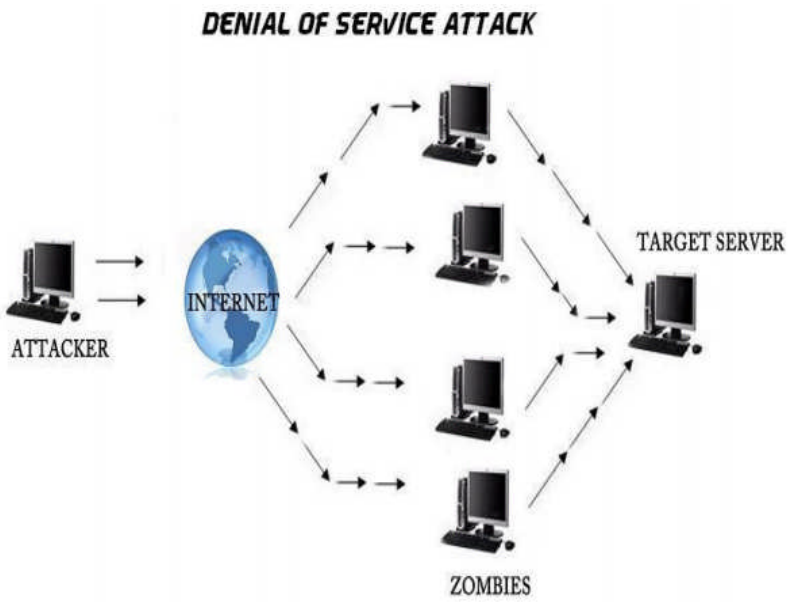

Figure 2.316: Attaque par déni de service [23].

Contrairement aux autres types de cyber-attaques, qui sont généralement lancées pour créer des informations sensibles à long terme et détourner des informations sensibles, les attaques par déni de service consistent à rendre les serveurs indisponibles par une consommation abusive des ressources telles que les processeurs, la mémoire ou le réseau. L'idée, pour le pirate, c'est d'envoyer des millions de requêtes automatisées à un service pour le saturer. Ces attaques sont de plus en plus sophistiquées et difficiles à détecter avant qu'il ne soit trop tard. De plus, pour une entreprise, elle peut recevoir une facture astronomique à cause des ressources utilisées pendant l'attaque.

2.3. Les risques de sécurité du Cloud :

Selon un rapport intitulé "Assessing the Security Risks of Cloud Computing.", présenté par le groupe d'analyse Gartner, le Cloud Computing est exposé à des risques de sécurité. Les clients intelligents poseront des questions difficiles et envisageront pour obtenir une évaluation sur la sécurité des services du Cloud Computing.

Les utilisateurs peuvent toujours exiger certaine transparence par les fournisseurs à la manière dont ils gèrent des incidents de sécurité et la confidentialité. Ils doivent aussi poser des questions sur l'équipe technique (les architectes, les codeurs et les opérateurs), les processus de contrôle des risques et les mécanismes techniques, le niveau de test effectué pour vérifier que les processus de service et de contrôle fonctionnent comme prévu et que les fournisseurs peuvent identifier les vulnérabilités imprévues.

Voici les sept risques de sécurité que Gartner recommande aux clients d'examiner avant de sélectionner un fournisseur de Cloud [24].

a. La qualité des superviseurs :

Sous-traiter ses données les plus sensibles ne peut s'envisager que si on a la certitude que les informaticiens du sous-traitant sont dignes de confiance et que leurs faits et gestes sont contrôlés. Gartner recommande un droit de regard et de contrôle sur les personnels du fournisseur.

b. Conformités légales :

En fin de compte, c'est le propriétaire des données qui est tenu responsable en cas d'infraction à la législation. Les fournisseurs de service doivent se plier à toutes les demandes d'audit externes et disposer de toutes les certifications de sécurité nécessaires pour que les clients aient la certitude d'être couverts.

c. Localisation des données :

L'utilisation de sites de stockage multiples fait partie des points forts de cette approche, mais aussi de ses points faibles. En effet, la dématérialisation touche à ses limites lorsqu'on s'intéresse au lieu où se trouve implanté un site de stockage. Les données qu'il contient relèvent alors du régime juridique local. Autant savoir sous lequel peuvent se trouver ses données.

d. Isolement des données :

Par définition, le Cloud Computing rime avec le partage des ressources. Cela engendre une menace sur la confidentialité des données. Il faut s'assurer de leur cryptage correct et qu'il est possible de les isoler. Ce point est crucial. Un cryptage qui ne respecte pas les règles de l'art peut déboucher sur une perte irréversible.

e. Récupération :

Ignorer où se trouvent ses données ne veut pas dire que l'on ne puisse pas avoir l'assurance des moyens mis en place pour leur sauvegarde en cas de problème majeur. La réplication sur plusieurs sites distants est un impératif. Une restauration complète dans des délais contractuels l'est aussi.

f. Collaboration avec la justice :

Une architecture en Cloud Computing ne doit pas empêcher de répondre aux injonctions de la justice, que ce soit pour des raisons fiscales ou d'autres d'ordre juridique. La traçabilité de l'accès aux données, en particulier, peut être une gageure pour le fournisseur. Un accord contractuel voire, dans l'idéal, la démonstration qu'il a été répondu facilement aux demandes lors d'une précédente enquête, s'imposent.

g. Viabilité à long terme :

Le fournisseur idéal ne défaille jamais et gagne suffisamment bien sa vie pour, d'une part ne pas déposer le bilan et d'autre part ne pas devenir une cible et être absorbé. Quoi qu'il en soit, les données de ses clients doivent traverser ces éventuels aléas sans en être affectées et surtout pouvoir d'être restituées. La description précise de cette restitution (conditions, délais, formats) doit figurer dans le contrat originel.

2.4. Normes et stratégie de sécurité du Cloud :

L'Organisation pour l'avancement des normes d'information structurées (OASIS) élabore des normes pour la sécurité, le commerce électronique et les services Web. OASIS a créé des standards liés au Cloud concernant le paradigme SOA, la sécurité, les données (import et export), la gestion des identités, etc. [25].

Le comité technique de normalisation ISO/IEC JTC1/SC 38, intitulé « Cloud Computing and Distributed Platforms », a été créé en novembre 2009. Il est constitué de trois groupes de travail :

- G1: Cloud Computing Service Level Agreements (CCSLA).
- G2: Cloud Computing Interoperability and Portability (CCIP).
- G3: Cloud Computing Data and its Flow (CCDF).

Ce comité a développé plusieurs normes autour du Cloud, notamment la norme ISO/IEC 17203 qui spécifie un format ouvert de machines virtuelles, les normes ISO/IEC 17788 et ISO/IEC

17789 qui spécifient la nomenclature et l'architecture de référence d'une plateforme de Cloud et bien d'autres encore.

Le comité technique de normalisation ISO/IEC JTC1/SC 27, intitulé « IT Security Techniques », a été créé en 1989. Plusieurs normes développées par ce comité répondent également à des besoins et problématiques liés au Cloud. La norme ISO 27017 sert de code de conduite à la gestion de la sécurité de l'information dans le Cloud. Cette norme fournit des conseils afin d'implémenter les contrôles de sécurité nécessaires dans une infrastructure de Cloud. La norme ISO 27018 se focalise sur les aspects de données personnelles (Personally Identifiable Information, PII) enregistrées dans le Cloud et propose un ensemble de conseils, de mécanismes de contrôle et de bonnes pratiques afin de gérer ces données de manière sécurisée et afin d'augmenter la confiance dans le Cloud.

Lorsqu'un fournisseur de services Cloud n'expose pas les détails de sa propre politique interne ou de la technologie, les locataires doivent approuver les revendications de sécurité du fournisseur.

Tous les aspects de la sécurité doivent être intégrés à la stratégie de sécurité d'un Cloud, qu'il est préférable de concevoir sous forme d'un document formel ayant reçu l'approbation et la bénédiction totale de la direction. Une stratégie de sécurité sert de référence à partir de laquelle sont déduites les exigences de sécurité. Elle ne doit pas détailler les approches techniques ou architecturales, car elles risquent de changer plus fréquemment que la stratégie, mais doit présenter les exigences sous-jacentes d'un point de vue organisationnel ou métier [3].

2.5. Standards de sécurité dans le Cloud :

Les standards de sécurité définissent les processus, les procédures et les pratiques nécessaires utilisées par un programme de sécurité. Ces standards s'appliquent également aux activités informatiques liées au Cloud, incluent des étapes spécifiques à suivre pour assurer un environnement sécurisé, et garanties la confidentialité des informations confidentielles dans le Cloud [27].

a. Security assertion markup language (SAML):

Security assertion markup language est un standard informatique définissant un protocole pour échanger des informations liées à la sécurité. Il est basé sur le langage XML. SAML définit trois rôles: un utilisateur, un fournisseur de services (SP) et un fournisseur d'identité.

b. Open Authentication (OAuth) :

OAuth est un protocole libre, permet aux utilisateurs de donner au site ou logiciel « consommateur », l'accès à des informations personnelles provenant du site « fournisseur » de service ou de données, ceci tout en protégeant le pseudonyme et le mot de passe des utilisateurs. Par exemple, un site de manipulation de vidéos pourra éditer les vidéos enregistrées sur Dailymotion d'un utilisateur des deux sites, à sa demande [28].

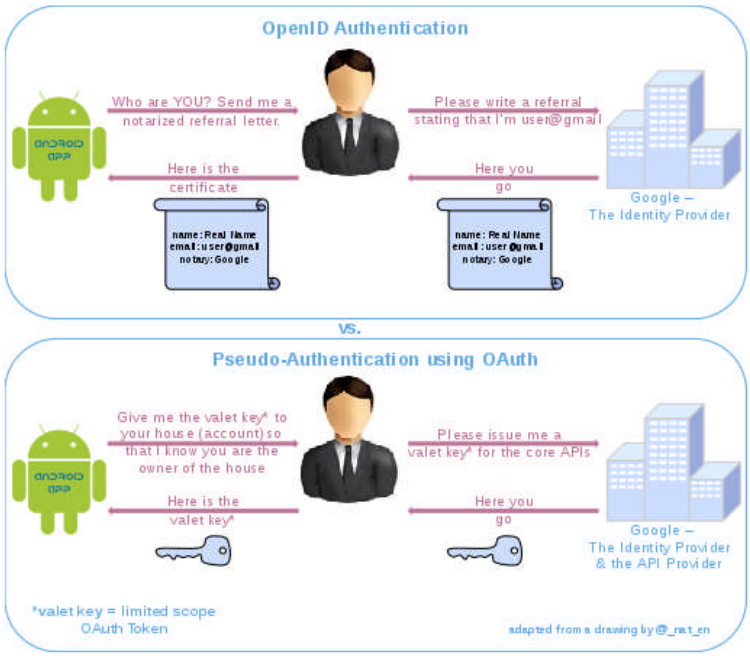

Figure 17.4: SAML ET OAuth [28].

c. OpenID:

OpenID est un système d'authentification décentralisé qui permet l'authentification unique, ainsi que le partage d'attributs. Il permet à un utilisateur de s'authentifier auprès de plusieurs sans avoir à retenir un identifiant pour chacun d'eux mais en utilisant à chaque fois un unique identifiant OpenID. Le modèle se base sur des liens de confiance préalablement établis entre les fournisseurs de services et les fournisseurs d'identité (OpenID providers). Il permet aussi d'éviter de remplir à chaque fois un nouveau formulaire en réutilisant les informations déjà disponibles. Ce système permet à un utilisateur d'utiliser un mécanisme d'authentification forte [29].

d. SSL/TLS:

TLS (Transport Layer Security) et son prédécesseur SSL (Secure Sockets Layer), sont des Protocoles de sécurité conçus pour assurer l'intégrité des données pour des communications sur TCP/IP.

TLS et SSL chiffrent les segments des connexions réseau sur la couche de transport. Le protocole TLS permet aux applications client-serveur de communiquer sans risque d'interception, d'altération ou de falsification des messages [27].

e. Secure Shell (SSH) :

SSH est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur [30].

2.6. Sécurité physique :

L'étendue des problèmes de sécurité physique est vaste et implique de nombreuses mesures pour éviter, empêcher, détecter et répondre aux accès non autorisés aux bâtiments, aux ressources ou aux informations présentes dans les locaux. La sécurité physique d'un bâtiment doit être vue comme un système de protection, avec les éléments de sécurité individuels se complétant les uns les autres pour mettre en place une défense multifacette à plusieurs niveaux. Ces éléments comprennent une conception environnementale, des contrôles d'accès (mécaniques, électroniques et procéduraux), une surveillance (capteurs vidéo, thermiques, de proximité), une identification du personnel avec un contrôle des accès, et une détection des intrusions associée à des systèmes de réponse (témoins, grilles, zones fermées).

La sécurité physique d'un bâtiment doit être constituée de couches dont chaque élément est associé à un contrôle général automatisé et à un centre de surveillance.

La planification d'une sécurité physique efficace implique une prise en compte approfondie des circonstances qui seront rencontrées, en incluant les activités normales et les situations imprévues. Les éléments de la sécurité physique doivent être soutenus par des procédures appropriées et mises en œuvre par un personnel professionnel expérimenté. Cette équipe de sécurité doit avoir pour mission exclusive la protection des biens et l'application des procédures de sécurité physique, même en cas de désastre. Étant donné l'étendue et la complexité d'une planification de la sécurité physique, une bonne solution consiste à la confier à des experts expérimentés et reconnus [31].

2.7.Sécurité de réseau:

Le recours au Cloud Computing exige un renforcement de la sécurité, notamment en matière d'accès et d'infrastructure réseau afin d'en protéger les données ou ressources partagées par nombre d'utilisateurs :

a. Contrôles d'accès au réseau :

L'accès au réseau est un point de sécurité fondamental garantit que les vecteurs d'attaque de base sont atténués par des contrôles efficaces. Les contrôles peuvent être implémentés dans des dispositifs physiques ou virtuels.

b. Contrôle de pare-feu :

Dans le Cloud les pare-feux fournissent un contrôle de protocole en temps réel pour détecter et bloquer les attaques connues, ceux-ci assure une sécurité distincte dans la couche de virtualisation du Cloud (les niveaux de réseau des machines virtuelles créés au sein du Cloud) et les attaques extérieurs.

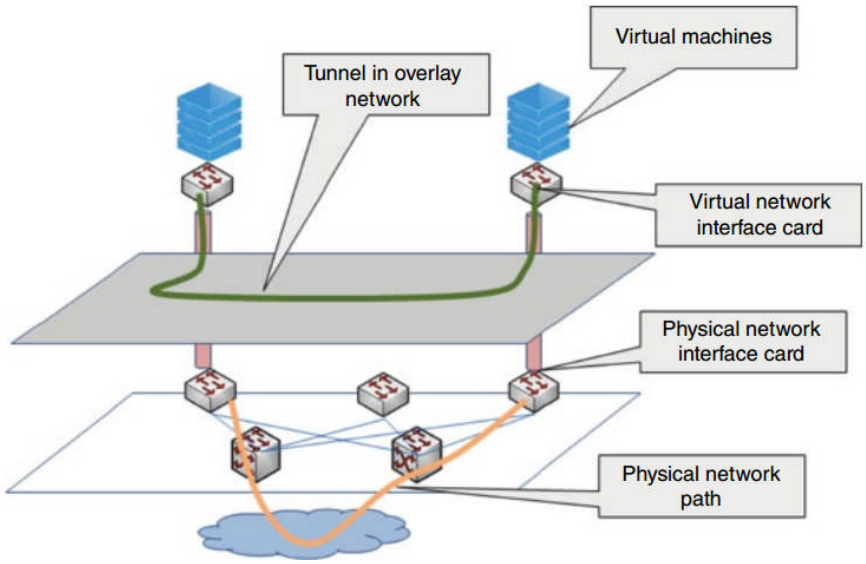

Figure 18.5:Architecture de réseau dans le Cloud [32].

c. Liste de contrôle d'accès (ACL) :

Les listes de contrôle d'accès fournissent une couche de contrôle de sécurité de base pour prendre en charge la sécurisation des machines virtuelles contre les menaces de sécurité de couche 2 standard, telles que l'inondation et l'analyse.

d. Contrôle de contenu :

Diverses technologies existent pour protéger le réseau, les systèmes d'entreprise et les données professionnelles contre les attaques externes et le vol de données internes. Ceux-ci incluent la détection d'intrusion, la prévention d'intrusion, la prévention de perte de données et les serveurs proxy.

e. Protection contre les DDOS (Distributed Denial of Service) :

L'attaque par déni de service distribué peut être atténuée une fois qu'une condition d'attaque est identifiée, les entités de surveillance déclenchent un réacheminement du trafic suspect via une instance de nettoyage.

2.8.Sécurité de données dans le Cloud:

La sécurité des données dans le Cloud reste un frein pour certaines entreprises désireuses d'adopter cette technologie pour leurs applications ou leurs infrastructures informatiques.

a. Authentification et identité:

L'authentification des utilisateurs et des systèmes communicants est effectuée de différentes stratégies, la plus connue est la cryptographie. La validation des clients se fait de plusieurs techniques comme les mots de passe, ou sous la forme d'une quantité mesurable comme l'empreinte digitale. Dans le Cloud, le problème qui se pose lors de l'utilisation de cette approche est la multitude de fournisseurs de services Cloud [33].

b. Intégrité et confidentialité des données :

Le Cloud Computing fournit des données et des ressources à des utilisateurs valides. Les ressources peuvent être accessibles via les navigateurs Web et peuvent également être consultées par des attaquants malveillants [34], Certains mécanismes sécurisé comme les certificats RSA et les tunnels SSH doivent être fournis pour éviter tels risques.

c. Disponibilité de l'information :

La non-accessibilité de l'information ou des données est un problème majeur concernant les services du Cloud. SLA (Service Level agreement) est utilisé pour indiquer si les ressources du système sont accessibles aux clients ou non. C'est un lien de confiance entre le client et le fournisseur [34].Une approche pour donner aux utilisateurs l'accessibilité aux informations est d'avoir un plan de sauvegarde pour les ressources locales et les données significatives.

d. Chiffrement de données :

Les données des utilisateurs en transit ou en repos sont exposées au vol, pour cela, des techniques devraient être utilisés afin d'assurer la vie privé et la confidentialité de données. Pour augmenter le niveau de sécurité des données dans le Cloud, il est important de les chiffrer en utilisant l'anonymisation avec des sauvegardes et des audits [35].

Les algorithmes de chiffrement utilisés dans le Cloud Computing pour sécuriser les données sont : AES, RSA, BLOWFISH, DES et IDEA. Le tableau 1.3 présente une comparaison entre les deux algorithmes les plus utilisés :

Caractéristiques:	**AES**	**RSA**
Platform	Cloud Computing.	Cloud Computing.
Évolutivité	Scalable.	Non-scalable.
Sécurité	Fournisseur et l'utilisateur.	Utilisateur.
Capacité de chiffrement des données	Enorme quantité de données.	Peu de données.
Type d'authentification	Meilleur Authentification pour le fournisseur.	Robuste.
Utilisation de mémoire (RAM)	Peu de besoins.	Mémoire plus élevée.
Temps d'exécution	Plus rapide que d'autres.	Nécessite maximum de temps.

Tableau 1.3 : Comparaison entre les algorithmes AES et RSA [36].

2.9. Fondamentales sur la Cryptographie :

La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné. La cryptographie utilise des concepts issus de nombreux domaines (Informatique, Mathématiques, Electronique). Toutefois, les techniques évoluent et trouvent aujourd'hui régulièrement racine dans d'autres branches (Biologie, Physique, etc.).

Depuis de nombreuses années, la cryptographie était le domaine exclusif des militaires, des services diplomatiques et gouvernementales secrètes, et a été utilisé pour fournir principalement des propriétés de sécurité, telles que la confidentialité des données, l'intégrité des données et l'authentification de l'origine des données.

Le chiffrement consiste à transformer une donnée (texte, message, etc.) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de déchiffrement.

Un exemple simple est le chiffrement de César où l'on décale simplement les lettres de l'alphabet, il suffit alors à l'interlocuteur de connaître le nombre de lettres à décaler pour retrouver le message originel.

Le problème de cette méthode est qu'une personne interceptant le message et connaissant la méthode utilisée n'a que 25 décalages non triviaux à tester pour retrouver le message. De nombreuses méthodes plus complexes ont vu le jour et à présent on considère qu'un attaquant doit faire face à au moins 2^{80} possibilités pour considérer le chiffrement comme sûr. Cette limite va certainement passer à 2^{100} à cause de l'augmentation des capacités calculatoires des ordinateurs.

2.9.1. Crypto-système :

Le but de la cryptographie, est de permettre à Bob d'envoyer des messages à Alice avec des garanties sur [26] :

- L'intégrité du message : le n'a pas été modifié entre son envoie et sa réception.
- La confidentialité : seuls Bob et Alice connaissent le contenu du message.
- La non-répudiation : il est bien possible que Bob il est bien l'expéditeur du message.

Un crypto-système est un ensemble d'algorithmes permettant de chiffrer et déchiffrer des messages. Éventuellement, un crypto-système pourra être accompagné d'un algorithme permettant la génération des clés de chiffrement et déchiffrement.

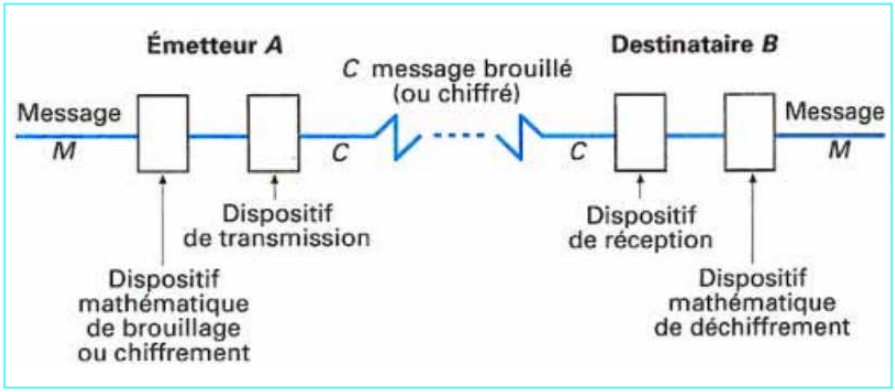

Figure 196:Schéma général de la communication chiffrée [37].

a. Crypto-système symétrique :

Dans un crypto-système symétrique, aussi appelé chiffrement à clef secrète, deux utilisateurs voulant communiquer vont tout d'abord convenir d'une clef K à utiliser qu'ils gareront secrète. Lorsque l'un d'entre eux souhaite communiquer le message M il lui appliquera tout d'abord la fonction de chiffrement E() en utilisant la clef K pour produire le chiffré C = E(M, K). En envoyant le chiffré C sur le réseau l'utilisateur sait que s'il est intercepté par un tiers ce dernier ne pourra pas en comprendre le sens, seul son interlocuteur connaissant K pourra effectuer la transformation inverse et obtenir M = D(C, K), où D() est la fonction de déchiffrement inverse de E().

Les principaux types de crypto-systèmes symétrique utilisés aujourd'hui se répartissent en deux grandes catégories : les crypto-systèmes par flots et les crypto-systèmes par blocs. Dans un crypto-système par flots, le cryptage des messages se fait caractère par caractère ou bit à bit, la taille de la clef est donc égale à la taille du message. Dans la deuxième classe le texte clair est fractionné en blocs de même longueur à l'aide d'une clef unique. Les algorithmes de chiffrement par blocs sont en général construits sur un modèle itératif.

La cryptographie symétrique est basée sur l'utilisation de deux algorithmes connexes pour le chiffrement des messages et le déchiffrement.

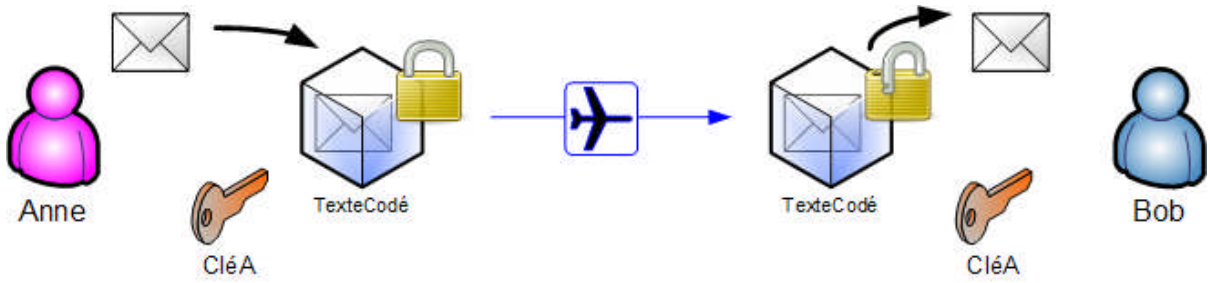

Figure 207:Schéma du chiffrement symétrique [38].

Les algorithmes de chiffrement symétrique ont des avantages. En général, ils sont très rapides. D'une part, ils ont une complexité moins élevée qu'un algorithme de chiffrement à clé publique tel que RSA et, d'autre part, ces algorithmes sont souvent très bien adaptés à l'architecture du processeur utilisé. Ils sont donc très bien appropriés au chiffrement de grande quantité de données. Les plus connus sont les algorithmes DES et AES [3].

i. Le Crypto-système AES :

L'AES (Advanced Encryption Standard) est le crypto-système le plus utilisé et le plus sûr disponible aujourd'hui. L'histoire de l'AES a débuté en 1997 lorsque le NIST décide de trouver un

successeur à un algorithme plus ancien, le DES (Data Encryption Standard). L'AES est un standard, donc libre d'utilisation, sans restriction d'usage ni brevet.

Ce crypto-système est fondé sur des entrées permutés selon une table définie au préalable, l'algorithme offre des tailles de blocs et de clés qui sont des multiples de 32 (compris entre 128 et 256 bits). Ces différentes opérations sont répétées plusieurs fois et définissent un «tour». A chaque tour, une clé unique est calculée à partir de la clé de cryptage et incorporée dans les calculs.

L'algorithme Rijndael est officiellement devenu la norme de cryptage AES après sa victoire sur ses concurrents lors d'une compétition internationale organisée en 2001. Ce nouvel algorithme se nomme Rijndael en l'honneur de ses créateurs, les chercheurs Belges Daemen et Rijmen.

ii. Algorithme de chiffrement AES (Rijndael) :

L'AES opère sur des blocs de 128 bits qu'il transforme en blocs cryptés de 128 bits par une séquence de N opérations ou « rounds », à partir d'une clé de 128, 192 ou 256 bits. Suivant la taille de celle-ci, le nombre de rounds diffère : respectivement 10, 12 et 14 rounds. L'opération d'encryptions se faite selon l'algorithme suivant :

Algorithme d'Encryptions AES :

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[0, Nb-1])

   for round = 1 step 1 to Nr-1
      SubBytes(state)
      ShiftRows(state)
      MixColumns(state)
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   end for

   SubBytes(state)
   ShiftRows(state)
   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   out = state
end
```

- **Chiffrement** :

La figure en dessus décrit succinctement le déroulement du chiffrement:

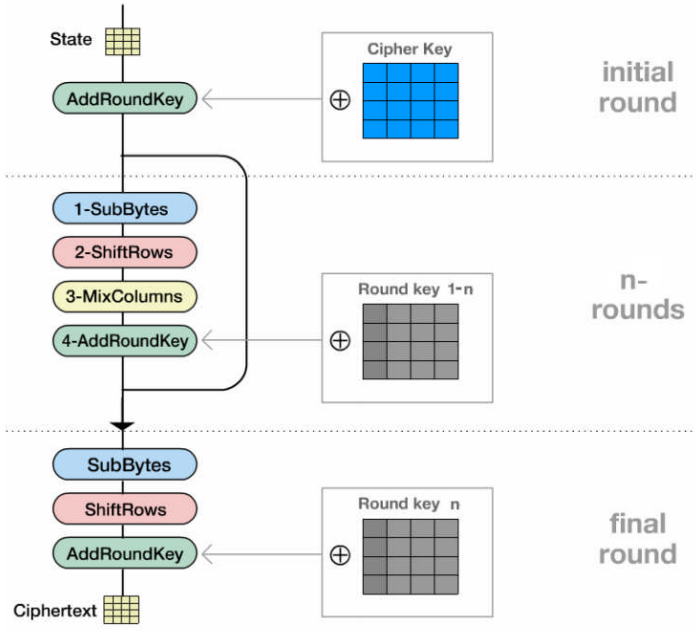

Figure 218: Structure générale de chiffrement.

Au début de chaque chiffrement, le block est transformé en état. Un état (state) est considéré comme un tableau de colonnes, chaque colonne comprenant 4 octets (32 bits).

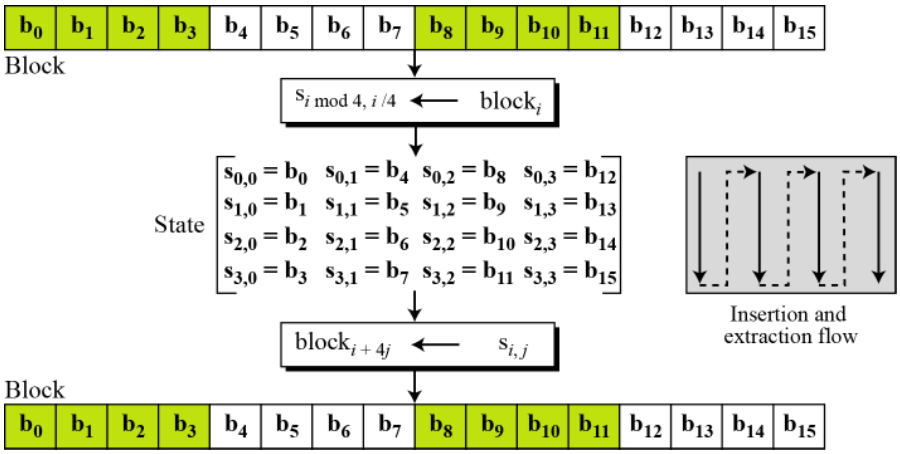

Figure 229: Transformation de block en état.

- **Transformation SubBytes () :**

SubBytes est une transformation non linéaire appliquée indépendamment à chacun des octets de l'état en utilisant une table de substitution (S-box). La table S-box contient une permutation de toutes les 256 valeurs de 8 bits. Chaque octet d'état est remplacé par un octet indexé par une ligne et une colonne dans la table.

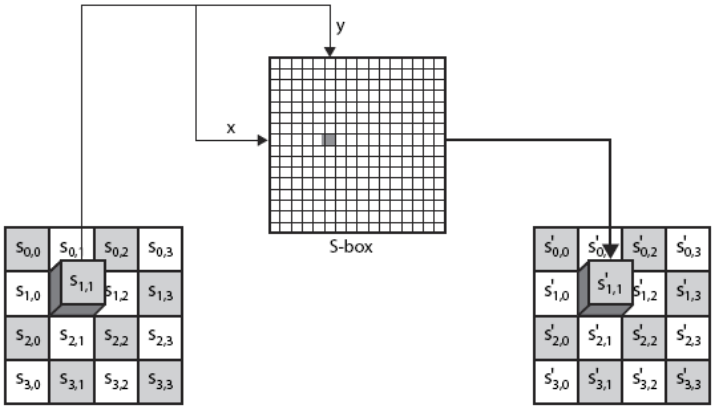

Figure 23: Transformation SubBytes.

- **Transformation ShiftRows () :**

Dans cette étape une permutation cyclique des octets sur les lignes de l'état est effectuée. Le décalage des octets correspond à l'indice de la ligne se fait comme suit :

- La 1ère rangée reste la même.
- La 2e rangée fait un décalage circulaire de 1 octet vers la gauche.
- La 3ème rangée fait un décalage circulaire de 2 octets vers la gauche.
- La 4ème rangée fait un décalage circulaire de 3 octets vers la gauche.

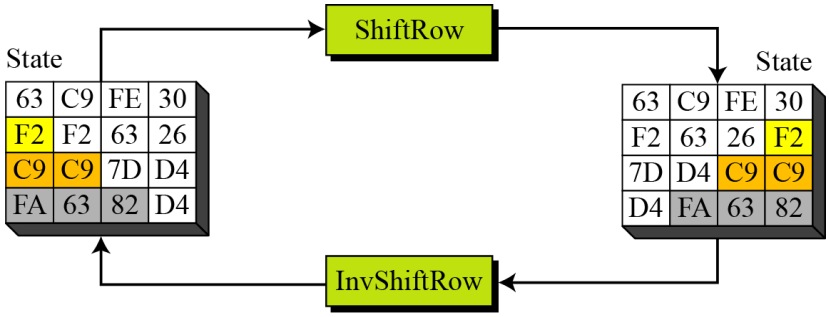

Figure 2411: Transformation ShiftRows/ InvShiftRows.

- **Transformation MixColumns () :**

Cette transformation linéaire est appliquée à un état colonne après colonne, on utilisant les 4 octets un produit matriciel est appliqué à chaque colonne. Les colonnes sont traitées comme des polynômes dans GF (28) et multipliées modulo x4 + 1 avec les polynômes fixes donnés:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

- **Transformation AddRoundKey () :**

Un XOR (au niveau des bits) est appliqué dans cette transformation entre chacun des octets de l'état et la clé de ronde.

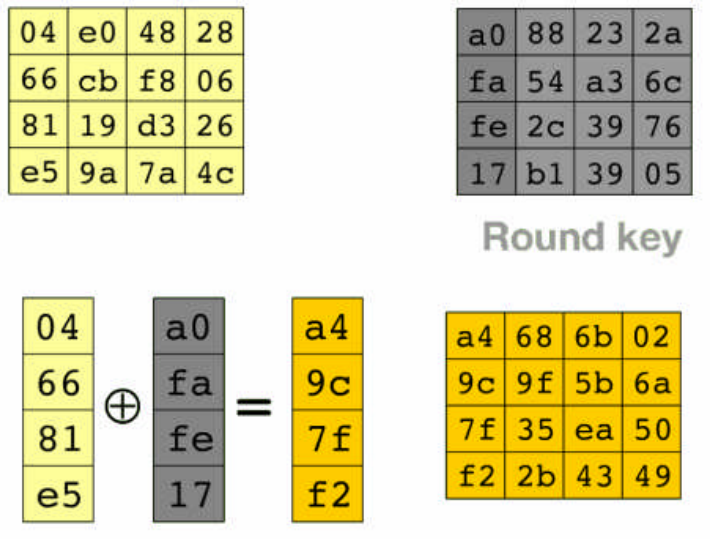

Figure 25.12: Transformation AddRoundKey.

- **Key Expansion (Diversification de la clé) :**

Avant le cryptage ou le décryptage, la clé doit être expansée pour être utilisée dans la transformation AddRoundKey.

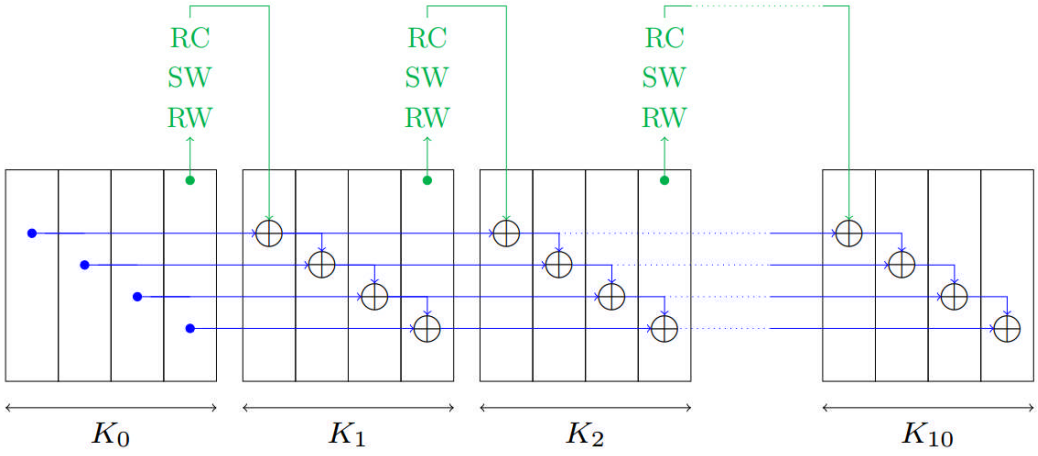

Figure 26.13: Diversification de la clé.

Chaque fois que la fonction AddRoundKey est appelée, une partie différente de la clé expansée est utilisé pour effectuer l'addition exclusive. Et pour que cela fonctionne, la clé expansée doit être suffisamment grande pour fournir des éléments clés à chaque ronde. La clé diversifiée est extraite selon l'algorithme suivant :

Algorithme Key Expansion :

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
   word  temp

   i = 0

   while (i < Nk)
      w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
   end while

   i = Nk

   while (i < Nb * (Nr+1)]
      temp = w[i-1]
      if (i mod Nk = 0)
         temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
      else if (Nk > 6 and i mod Nk = 4)
         temp = SubWord(temp)
      end if
      w[i] = w[i-Nk] xor temp
      i = i + 1
   end while
end
```

La fonction Rcon renvoie une valeur de 4 octets suivant la valeur d'entrée. Les résultats possibles sont :

Rcon(0) = 01000000
Rcon(1) = 02000000
Rcon(2) = 04000000
Rcon(3) = 08000000
Rcon(4) = 10000000
Rcon(5) = 20000000
Rcon(6) = 40000000
Rcon(7) = 80000000
Rcon(8) = 1B000000
Rcon(9) = 36000000
Rcon(10) = 6C000000
Rcon(11) = D8000000
Rcon(12) = AB000000
Rcon(13) = 4D000000
Rcon(14) = 9A000000

b. Crypto-système asymétrique (à clé publique) :

Dans un crypto-système asymétrique, aussi appelé chiffrement à clef publique, chaque utilisateur génère une paire de clefs, l'une appelée clef secrète qu'il est le seul à connaître, l'autre appelée clef publique qu'il diffuse sur un annuaire et qui peut servir aux autres pour le contacter. Lorsqu'un utilisateur veut envoyer un message M il va aller chercher la clef publique du destinataire Kpub,dest et l'utilise pour créer un chiffré C = E(M, Kpub,dest). Contrairement au chiffrement symétrique, il sera alors lui-même dans l'incapacité d'effectuer la transformation inverse ne possédant pas la clef privée de son interlocuteur. L'interlocuteur sera le seul à pouvoir récupérer le message originel M = D(C, Kpriv,dest), grâce à sa clef privée Kpriv,dest.

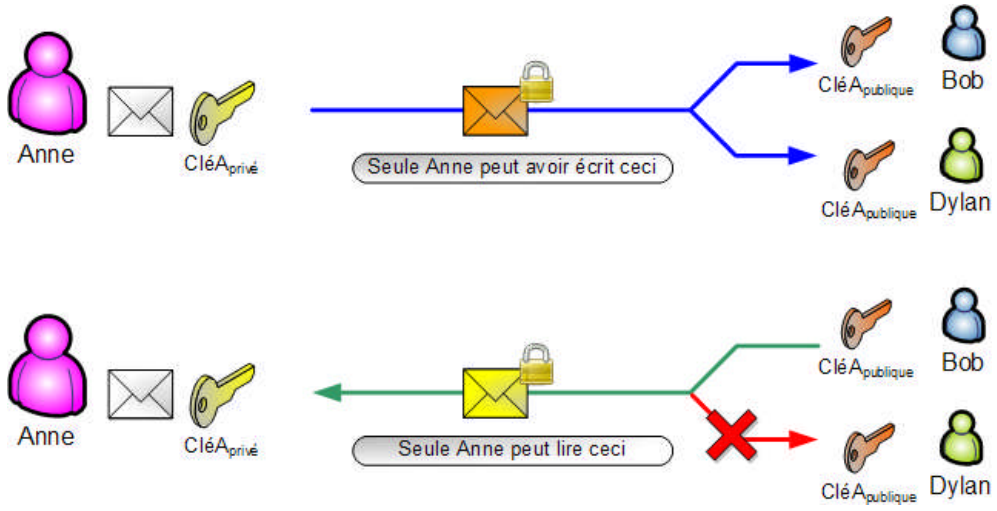

Figure 2.14:Schéma du chiffrement asymétrique [38].

i. Crypto-système RSA :

RSA est l'un des systèmes de chiffrement asymétrique ayant connu un grand succès d'utilisation. Découvert en 1973 par l'agence de renseignement britannique GCHQ, il reçoit la classification "secret défense". Les chercheurs Rivest, Shamir et Adleman sont crédités de le rendre public en 1977.

Contrairement aux systèmes de chiffrement symétriques traditionnels, RSA fonctionne avec deux clés différentes: une publique et une privée. Les deux se complètent, ce qui signifie qu'un message chiffré avec l'une ne peut être décodé sans l'autre. La clé privée ne peut pas être calculée à partir de la clé publique, cette dernière est donc généralement accessible par tous.

Ces propriétés permettent d'utiliser des crypto-systèmes asymétriques dans un large éventail de fonctions, telles que les signatures numériques. Par exemple, lors de la signature d'un document, une empreinte chiffrée RSA est jointe au fichier et permet au destinataire de vérifier à la fois l'expéditeur et l'intégrité du document. La sécurité des clés RSA est fondée sur la difficulté de factoriser des nombres entiers. Ainsi, un message sur le point d'être chiffré est traité comme un nombre entier. Par la suite, lors du chiffrement de ce message, il est élevé à la puissance de la clé utilisée, puis divisé par le reste du produit de deux nombres premiers. En répétant le processus avec l'autre clé, le texte en clair peut être récupéré. Actuellement, la meilleure méthode connue pour le craquer nécessite de factoriser le produit utilisé dans la division et il n'est pas possible de calculer ces facteurs pour des nombres supérieurs à 768 bits. C'est pourquoi les crypto-systèmes modernes utilisent des clés de 3072 bits au minimum.

ii. Algorithme de chiffrement RSA:

RSA travaille principalement avec des nombres premiers (donc entiers naturels). A priori, il est assez facile de dire si un petit nombre est premier ou non. Cependant, dès que ce nombre devient plus conséquent, il est très rapidement difficile de dire s'il s'agit d'un premier ou non.

- **Génération des clefs :**

Pour générer les clefs il faut choisir deux nombres premiers très grand ***p*** et ***q***, ensuite calcul n, qui est un constituant de la clé publique et de la clé privée en faisant : ***n = p*q.***

On choisit ensuite une clef de chiffrement ***e*** telle que ***φ (n) = (p-1) (q-1)*** soient premiers entre eux. Les nombres n et e forment la clef publique du crypto-système. Finalement, on utilise l'algorithme d'Euclide étendu pour calculer la clef de déchiffrement d telle que : ***ed =1mod (n).***

En d'autres termes ***d*** est l'inverse modulaire de ***e*** modulo ***(n)***. Le nombre ***d*** est la clef privée. Les deux nombres premiers ***p*** et ***q*** ne sont plus nécessaires. Ils peuvent être écartés mais jamais révélés.

- **Chiffrement** :

Pour chiffrer un message M, on le découpe en blocs numériques tels que chaque bloc ait une représentation unique modulo n (avec des données binaires, on choisit la plus grande puissance de 2 inférieur à n). Ainsi, si p et q sont tous deux des nombres premiers de 100 chiffres, alors *n* aura tout juste moins de 200 chiffres et chaque bloc de message doit avoir juste moins de 200 chiffres. Le message chiffré C sera constitué de manière similaire de blocs. La formule de chiffrement est simplement:C_i=m_i^e mod(n).

- **Déchiffrement :**

Pour déchiffrer un message, il faut effectuer l'opération inverse, c'est à dire pour chaque on calcule :

m_i=C^d mod (n).

iii. Sécurité du système RSA :

Supposons qu'un intrus intercepte le message c et cherche à le décrypter. Il connaît aussi la clé publique, à savoir les nombres n et e. En revanche, il ne connaît pas d. Pour découvrir ce nombre, il doit trouver p et q. En effet, la seule façon connue de découvrir d en connaissant n et e est de factoriser n pour connaître ***φ (n) = (p-1) (q-1)***, et de calculer ensuite la solution de l'équation : ***e d ≡ 1 mod φ (n)***. La difficulté vient de ce que la factorisation de n est une tâche impossible à effectuer en un temps raisonnable, dans l'état des connaissances actuelles, pour autant que n soit suffisamment grand. On prend aujourd'hui des premiers p et q tels que leur produit soit un nombre s'écrivant avec plus de 200 chiffres. Le système est donc sûr, du moins tant que l'on ne découvre pas un algorithme rapide pour factoriser les entiers.

2.9.2. Fonction de hachage :

Une fonction de hachage est une méthode permettant de caractériser une information, une donnée. En faisant subir une suite de traitements reproductibles à une entrée, elle génère une empreinte servant à identifier la donnée initiale. De telles fonctions datent de la fin des années 1980 (algorithme MD2) mais l'idée est plus ancienne, et a germé dès l'apparition des codes correcteurs d'erreurs (théorie de l'information).

Une fonction de hachage prend donc en entrée un message de taille quelconque, applique une série de transformations et réduit ces données. On obtient à la sortie une chaîne de caractères hexadécimaux, le condensée, qui résume en quelque sorte le fichier. Cette sortie a une taille fixe qui varie selon les algorithmes (128 bits pour MD5 et 160 bits pour SHA-1). Ces fonctions sont très utilisées en informatique et en cryptographie. On les rencontre en navigant sur le Web : les auteurs de logiciels proposent souvent des empreintes sur les pages dédiées aux téléchargements (des fichiers portant l'extension md5 ou sha1, qui contiennent la valeur hachée dudit programme). En comparant l'empreinte de la version téléchargée avec l'empreinte disponible sur le site, l'utilisateur peut s'assurer que sa version n'a pas été corrompue (erreurs de transmission, virus, etc).

2.9.3. Signatures numériques :

Un des avantages majeurs de la cryptographie à clé publique est qu'elle procure une méthode permettant d'utiliser des signatures numériques. Une signature est une primitive cryptographique introduite par Diffie et Hellman et qui permet d'authentifier une donnée. Les signatures numériques permettent à la personne qui reçoit une information de contrôler l'authenticité de son origine, et également de vérifier que l'information en question est intacte. Ainsi, les signatures numériques des systèmes à clé publique permettent l'authentification et le contrôle d'intégrité des données. Une signature numérique procure également la non-répudiation, ce qui signifie qu'elle empêche l'expéditeur de contester ultérieurement qu'il a bien émis cette information [3].

2.9.4. Certificat électronique :

Un certificat est l'équivalent d'une carte d'identité ou d'un passeport. Un passeport contient des informations concernant son propriétaire (nom, prénom, adresse, …), la signature manuscrite, la date de validité, ainsi qu'un tampon et une présentation (forme, couleur, papier) qui permettent de reconnaître que ce passeport n'est pas un faux, qu'il a été délivré par une autorité bien connue [40].

De manière très simplifiée, l'utilisateur désirant obtenir un certificat électronique fait une demande auprès de l'autorité d'enregistrement (AE). Après validation de l'identité du demandeur, l'AE génère un couple de clés (publique, privée), envoie la clé privée suivant des procédures sécurisées à l'utilisateur (chemin de confiance) et certifie la clé publique par l'autorité de certification en apposant sa signature électronique sur le certificat. Le certificat est alors installé sur un annuaire public accessible à tous.

À partir de ces informations, dont l'autorité de certification vérifie préalablement la validité, cette même autorité de certification génère une signature de certification en créant dans un premier temps une empreinte de ces informations grâce à un algorithme de hachage et en chiffrant cette empreinte par un algorithme de chiffrement asymétrique grâce à la clé privée de l'autorité de certification.

CHAPITRE III

Cryptanalyse d’AES

3.1.Introduction:

La cryptanalyse l'étude des mécanismes théoriques ou techniques visant à briser (casser) un algorithme de chiffre ment, c'est-à- dire le fait de retrouver le message M à partir de C, sans connaître la clé K a priori. Dans certains cas, il s'agira également de retrouver cette clé K [41].

La méthode basique pour retrouver une clef de chiffrement consiste à tester toutes les clefs jusqu'à trouver la clef correcte et s'appelle la méthode brute force. Toute méthode permettant de retrouver la clef en explorant un espace plus réduit que celui de la méthode brute force est une cryptanalyse [42].

3.2. Types des attaques cryptanalytiques :

Attaque sur le texte chiffré uniquement (cipher text-only) : le cryptanalyste possède des exemplaires chiffrés des messages, il peut faire des hypothèses sur les messages originaux qu'il ne possède pas.

Attaque à texte clair connu (known-plain text attack) : Etant donné un texte chiffré et un fragment de texte clair associé, on recherche le texte clair restant et/ou la clé.

Attaque sur un texte clair sélectionné (chosen-plaintext attack) : On peut faire crypter ce que l'on veut par la méthode de cryptage et voir ce qu'elle produit.

Attaque sur le texte chiffré uniquement (chosen-cicher text attack) : Etant donné la capacité de déchiffrer un fragment de texte chiffré choisi arbitrairement, on recherche la clé.

3.3. Attaques contre l'AES :

3.3.1. Attaques préexistantes :

a. La cryptanalyse différentielle:

Est l'une des premières attaques statistiques. Elle a été introduite en 1990 par Eli Biham et Adi Shamir dans le but de casser le DES.

La cryptanalyse différentielle s'effectue en général dans un contexte de texte clair choisi, ce qui signifie que l'attaquant est en mesure d'obtenir les résultats chiffrés de textes clairs de son choix. Il existe des variantes qui fonctionnent dans d'autres modes d'attaque : à texte clair connu ou à texte chiffré seulement. La cryptanalyse repose sur des paires de textes clairs qui ont une différence constante. L'opération de différence peut être définie de diverses manières, la Fonction OU exclusif est la plus courante. L'attaquant calcule ensuite les différences dans les textes chiffrés, afin d'en extraire des motifs pouvant indiquer un biais. Les différences en sortie du chiffrement sont nommées des différentielles.

b. La cryptanalyse linéaire:

Due à Mitsuru Matsui, Elle remonte à 1993 et s'avère être l'attaque la plus efficace contre DES. Ce type de cryptanalyse se base sur un concept antérieur à la découverte de Matsui : les expressions

linéaires probabilistes. Ces dernières ont été étudiées par Henri Gilbert et Anne Tardy-Corfdir dans le cadre d'une attaque sur FEAL (Fast Data Encipherment Algorithm).

La cryptanalyse linéaire est plus efficace que la cryptanalyse différentielle, mais moins pratique pour la simple et bonne raison que l'on part du principe que l'attaquant ne dispose pas de la boîte noire symbolisant l'algorithme de chiffrement, et qu'il ne peut pas soumettre ses propres textes. Dans le cas de DES, cette attaque nécessitait à l'origine 2^{47} couples (tous chiffrés avec la même clé) que l'attaquant a pu récupérer par un moyen ou un autre. Par la suite, Matsui améliore son algorithme en 1994 et propose une solution avec 2^{43} couples. La complexité avec une bonne implémentation est toutefois inférieure et de l'ordre de 2^{39} opérations DES.

La cryptanalyse linéaire consiste à faire une approximation linéaire de l'algorithme de chiffrement en le simplifiant. En augmentant le nombre de couples disponibles, on améliore la précision de l'approximation et on peut en extraire la clé. Les tables de substitution (S-Boxes) présentent en effet certaines propriétés linéaires, alors qu'elles étaient justement prévues pour résister devant ses attaques [43].

Les algorithmes plus récents comme AES (Rijndael), IDEA, et bien d'autres, sont insensibles à une attaque linéaire.

c. L'attaque boomerang:

Est une version améliorée de la cryptanalyse différentielle, cette méthode a été inventée par David Wagner en 1999. Elle consiste à attaquer les deux moitiés d'un algorithme de chiffrement par bloc et part du principe que certaines propriétés, après perturbations des entrées, ne se propagent pas à travers toute la structure.

L'attaque Boomerang fonctionne efficacement sur plusieurs chiffrements. Dans son papier, David Wagner montre comment l'utiliser dans le cadre d'une version simplifiée de Khufu de Ralph Merkle, (6 rondes de FEAL et 16 rondes de CAST-256). En 2004, elle a été mise en pratique sur 6 rondes d'AES par Alex Biryukov [43].

3.3.2. Attaques actuelles :

a. Les attaques algébriques :

Ce sont des attaques à clair connu qui exploitent des relations algébriques entre les bits du clair, ceux du chiffré et ceux de la clef secrète. La connaissance de plusieurs couples clairs-chiffrés fournit un système d'équations dont les inconnues sont les bits de la clef secrète. Ces derniers peuvent alors être retrouvés en résolvant le système, ce qui est possible s'il est de degré faible, de petite taille ou qu'il possède une structure particulière.

L'idée d'obtenir la clef par la solution des systèmes d'équations vient de C. Shannon, mais l'amélioration de l'efficacité de la méthode est récente. Les attaques algébriques ont été introduites en 2002.

b. L'attaque XSL :

Est une méthode heuristique de cryptanalyse contre les chiffrements par bloc. Elle a été publiée en 2002 par Nicolas Courtois et Josef Pieprzyk.

L'attaque XSL est basée sur la résolution d'un système d'équations quadratiques qui symbolisent la structure du chiffrement. Le système est typiquement très gros, plus de 8000 équations avec 1600 variables pour un AES de 128 bits. Plusieurs méthodes pour résoudre de tels systèmes sont connues (en particulier les bases de Gröbner) et une nouvelle technique a été proposée dans le papier sous le nom de eXtended Sparse Linearisation (XSL). Malgré ces découvertes, l'attaque reste purement théorique car elle demande une trop grande puissance de calcul.

c. L'attaque du Cube :

Est une méthode de cryptanalyse appliquer à une grande variété d'algorithmes à clé symétrique, publiée par Itai Dinur et Adi Shamir dans une pré-imprimer de septembre 2008, Une autre version révisée été mise en ligne en janvier 2009.

Cette attaque fait la somme des valeurs de bits en sortie pour toutes les valeurs possibles d'un sous-ensemble de bits d'entrée publics. Les bits en entrée sont choisis de sorte que la somme résultante soit une combinaison linéaire de bits secrets; l'application répétée de cette technique donne un ensemble de relations linéaires entre les bits secrets qui peuvent être résolus pour découvrir la clé. AES est immunisé contre cette attaque [46].

d. Les attaques par canal auxiliaire :

Ce sont des attaques basées sur des informations obtenues à partir du système informatique, plutôt que sur des faiblesses dans l'algorithme implémenté lui-même. Les informations de synchronisation, la consommation d'énergie, les fuites électromagnétiques ou même le son peuvent fournir une source d'information supplémentaire, qui peut être exploitée.

e. L'attaque biclique :

C'est une variante de la méthode de cryptanalyse meet-in-the-middle (MITM), elle utilise une structure biclique pour augmenter le nombre de rondes éventuellement attaqués par l'attaque MITM. Cette attaque est connues d'avoir cassé l'AES et l'IDEA, même si seulement avec un léger avantage par rapport brute force.

rounds	data	computations/succ.rate	memory	biclique length in rounds
AES-128 secret key recovery				
8	$2^{126.33}$	$2^{124.97}$	2^{102}	5
8	2^{127}	$2^{125.64}$	2^{32}	5
8	2^{88}	$2^{125.34}$	2^{8}	3
10	2^{88}	$2^{126.18}$	2^{8}	3

Table 1.4: Récupération de clé AES avec une attaque Biclique [39].

Il a été déterminé que la recherche en cryptanalyse progresse contre AES. En outre, il est recommandé de faire prudence vue au progrès réalisés dans le domaine public.

Les résultats montrent qu'AES est actuellement vulnérable à diverses attaques par canal auxiliaire. Cependant, des contre-mesures sont disponibles et peuvent éliminer ces vulnérabilités au niveau de des équipements si ils sont mises en œuvre correctement [47].

3.4. Contribution :

Le crypto-système AES utilisent la clé principale pour générer plusieurs clés de ronde. La diversification se faite à l'aide de la table Rcon, AES utilise la même clé pour encrypter les différents blocks de données. Notre méthode repose sur l'utilisation de ce crypto-système mais avec une clé dynamique (différente) pour l'encryptions de chaque block de données (128/192/256), toute en respectent les propriétés générales des crypto-systèmes :

- Réalisation simple et rapide du chiffrement et du déchiffrement (pour atteindre des débits élevés).
- Éviter un encombrement important des clés.
- Une méthode de cryptographie (fonctions E et D) doit être stable. On ne peut la changer que très rarement.
- Elle est le plus souvent publiée (largement connue).
- Un crypto système dépend de paramètres (clés) qui doivent pouvoir être modifiés aisément et fréquemment.
- On estime que la sécurité ne doit pas dépendre du secret des algorithmes E et D mais uniquement du secret des clés k et k' (exception pour le domaine militaire).

Voyons ces propriétés l'émetteur et le récepteur ne devrait pas échanger les nouvelles clés pour chaque block. Chaque clé (sauf la première) et extraite a partir l'ancien clé et le dernier block du texte claire traité (encrypter/décrypter).

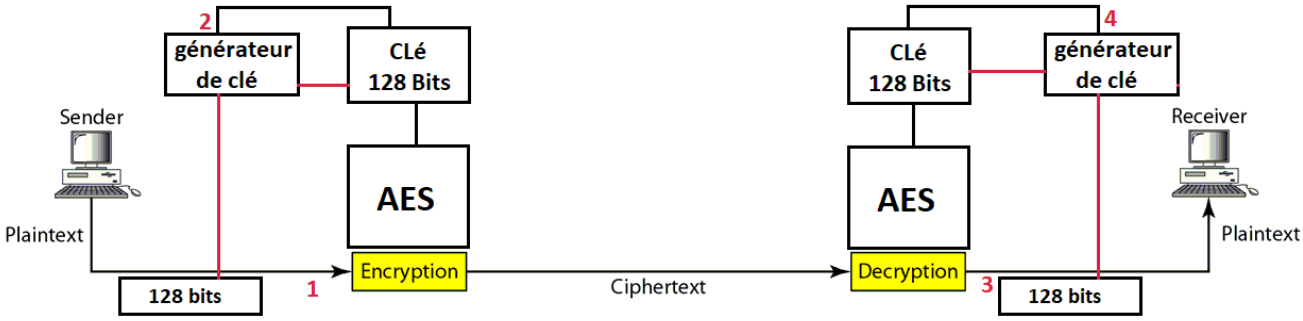

Figure 27: Modèle de la méthode proposé.

Algorithme d'encryptions AES-Dyn:

```
Cipher(state_t* state, uint8_t* RoundKey)
{  uint8_t round = 0;
   AddRoundKey(0, state, RoundKey);
  for (round = 1; round < Nr; ++round)
  {
    SubBytes(state);
    ShiftRows(state);
    MixColumns(state);
    AddRoundKey(round, state, RoundKey);
  }
  SubBytes(state);
  ShiftRows(state);
  AddRoundKey(Nr, state, RoundKey);
RoundKey ^= state;
}
```

a. Déroulement :

1. L'émetteur chiffre le premier block m1 avec la clé k1 qu'il possède ;
2. L'émetteur génère la clé k2 pour l'encryptions du block suivant avec : k2 = k1 ⊕ m1 ;
3. Le récepteur déchiffre le premier block chiffré c1 avec la même clé k1 pour qu'il avoir m1 ;
4. Le récepteur génère la clé k2 pour décrypter le block suivant avec : k2 = k1 ⊕ m1 ;

b. Implémentation:

Notre application est écrite en langage C. Le résultat ci-dessous (Tableau, Figure) est obtenu en utilisant un PC Lenovo ThinkPad T530 avec les spécifications suivantes: Intel (R) Core (TM) i7-3630QM CPU @ 2,33 GHz, avec 6820 Mo de RAM.

Taille de données (Kb)	8	16	32	64	128
AES encryption (second)	0,33	0,656	1,314	2,626	5,246
AES-Dyn encryption (second)	0,34	0,668	1,342	2,678	5,35

Table 1.5 : Temps d'exécution par taille de données avec AES et AES-Dyn.

La figure et le tableau 3.2 montrent le temps d'exécution requis pour AES et AES-Dyn pour encrypter différentes tailles de données, et avec une clé 128 bit.

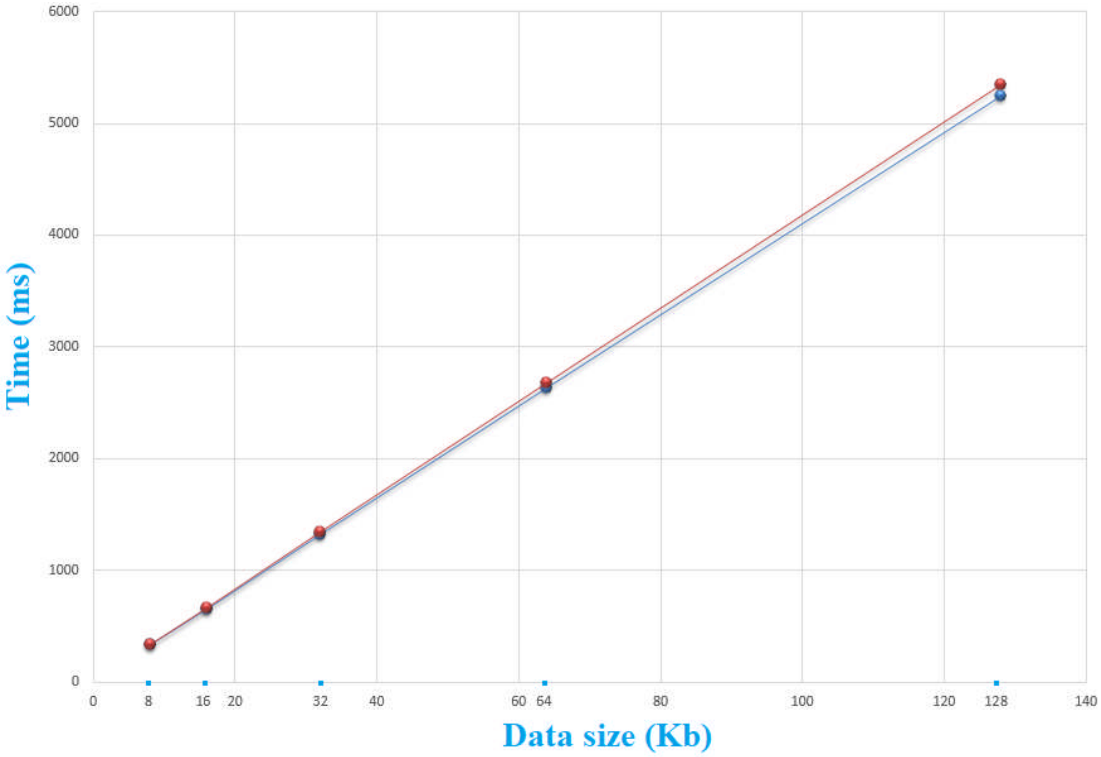

Figure 28:Graph de Temps d'exécution.

La figure suivante montre le résultat de chiffrement de 10 blocks de données (128bits) avec AES128 et AES-Dyn128.

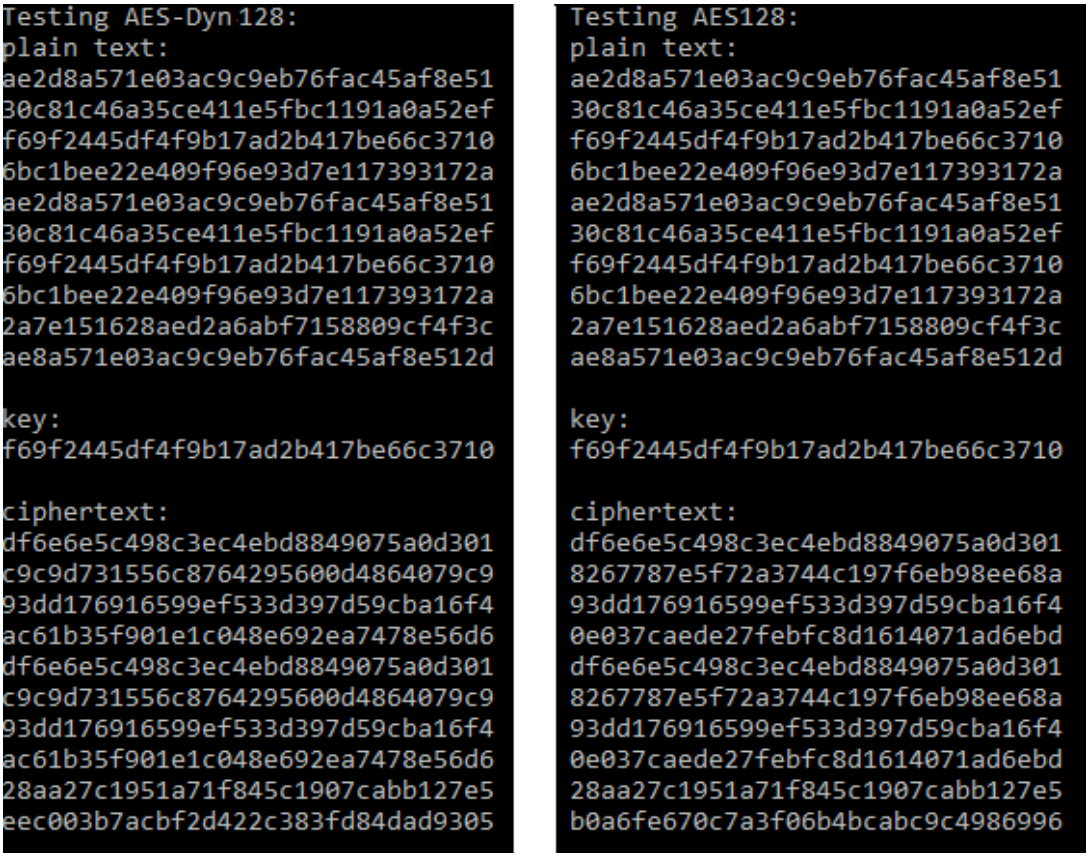

Figure 29: Résultat d'encryptions avec AES et AES-Dyn.

Supposons qu'une cryptanalyse sur AES qui nécessite 2^8 couple, AES-dyn met une condition de plus, l'attaquant devrait posséder les 2^8 couple consécutifs.

3.5. Conclusion :

Au cour de ce chapitre nous avons présenté les différents cryptanalyse sur AES, nous avons aussi proposer une approche basé sur le crypto-système AES pour mieux sécurisé les données.

Chapitre IV : mise en place d’une solution Cloud pour une boite de développement

4.1. Introduction :

Une boite de développement est une entreprise qui assure la conception, le développement et la commercialisation de produit logiciel. Elle peut confier la mise en œuvre, l'intégration, et la personnalisation à des entreprises de services du numérique.

L'édition logicielle peut être divisée en trois catégories fondamentales, celle des logiciels dits "horizontaux" (éditeurs et créateurs de logiciels proposant une offre générale à tous les secteurs d'activité), celle des logiciels dits "sectoriels" (éditeurs et créateurs de logiciels dédiés à un secteur d'activité particulier) et les particuliers et les jeux (éditeurs et créateurs de logiciels s'adressant aux particuliers et éditeurs de jeux).

4.1.1. Problématique :

Aujourd'hui, beaucoup de problèmes faire faces les développeurs qui veulent créer leur propre boite de développement, la nécessité de mettre en place une grande infrastructure (serveurs, terminaux, réseaux, etc.) est le premier obstacle.

Les principales exigences sont :

Matériels performants : certains logiciels font ralentir les ordinateurs, il requit beaucoup de ressource telle que les mémoires, vitesse de calcul, etc. dans le cas contraire, il est possible qu'un utilisateur est obligé d'obtenir un matériel spécifique pour pouvoir réaliser des taches peu se produisent

Mise à jour logiciels et licences: les outils utilisées doit être à jour avec les versions récentes pour bénéficier de nouveauté et des corrections. Certaines suites logicielles nécessitent une licence, faisant parfois l'acheter pour chaque poste de travail.

Synchronisation et transparence : parfois certaines applications devraient être développées simultanément à partir de plusieurs taches qui sont réalisées par différents développeurs. D'autres fois ces taches sont compléments, alors qu'il nécessite une transparence entre les développeurs.

4.1.2. Avantages de la solution:

a. Pour les administrateurs:

L'administrateur de projet peut gérer facilement les développeurs et contrôler leurs travaux à travers un seul poste. Pour une telle solution pas besoin de fournir plus de budget pour acheter des licences à chaque poste. Les mises à jour pour l'ensemble des postes ne sont nécessaires, il suffit de mettre à jour l'application sur le cluster et tous les utilisateurs bénéficient des nouveautés et des corrections. Il en résulte une plus grande cohérence de travail et des taches produites par l'ensemble de contributeurs.

b. Pour les développeurs :

Les utilisateurs bénéficiés des ressources très puissantes qui repondent à leurs nécessités. Ils peuvent à tout moment et à partir de n'importe quel appareil de travailler sur leurs taches. Ils peuvent accéder à partir de n'importe quel type d'appareil doté d'un navigateur.

4.2. Solution choisi « OpenStack » :

OpenStack est un logiciel permettant de déployer des infrastructures de cloud computing (IaaS) sous licence Apache. Il est devenu open source en juillet 2010 lorsque Rackspace Hosting et la NASA ont lancé conjointement un nouveau projet dans le domaine de Cloud Computing sous le nom «OpenStack ». L'objectif de ce projet est de permettre à toute organisation de créer et d'offrir des services de cloud en utilisant du matériel standard. La première version livrée par la communauté, dont le surnom est Austin, fut disponible dès octobre 2010, et la dernière version nommé Queens sortie en Février 2018 [44].

OpenStack s'installe sur un système d'exploitation libre comme Ubuntu et se configure entièrement en ligne de commande. C'est un système robuste et qui a fait ses preuves auprès des professionnels du domaine.

OpenStack est constituée de plusieurs services qui s'installent séparément. Ces services interagissent en fonction des besoins du cloud. Il s'agit notamment du service Compute, du service d'Identité, du service Réseau, du service Image, du service de Stockage Bloc, du service de Stockage Objet, du service de Télémétrie, du service d'Orchestration et du service de Base de Données. Ces services peuvent être installés indépendamment et configurés comme autonomes ou en tant qu'entités connectées.

a. Le service d'identité (Keystone):

- Fournit un service d'authentification et d'autorisation pour les autres services d'OpenStack.
- Fournit un catalogue de points d'extrémité pour tous les services d'OpenStack.

b. Le service Image (Glance) :

Il permet aux utilisateurs de découvrir, enregistrer et récupérer les images de machines virtuelles. Il offre une API REST qui permet d'interroger les metadata des images de machine virtuelle et de récupérer une image existante. Ce service se compose de quatre parties principales :

- glance-api : ce démon traite les appels à l'API pour la gestion des images. Il permet notamment de lister les images disponibles, récupérer une image ou en créer une nouvelle.
- glance-registry : stockage, traitement et récupération des méta-données associées aux images.
- Base de données : la base de données est utilisée pour stocker les méta-données.
- Un répertoire de stockage ou les fichiers d'image sont stockés.

c. Le service Compute (Nova) :

Gère le cycle de vie des instances dans un environnement OpenStack. Les tâches incluent la planification, la création et la mise hors service de machines virtuelles à la demande. Nova est divisée en composants :

- nova-api : traite les requêtes API venant des clients.
- nova-compute : daemon nova qui gère les VM.
- nova-scheduler : coordonne le déploiement des VM sur les machines physiques.
- nova-conductor : interlocuteur entre nova-compute et la base de données.

d. Le service Réseau (Neutron) :

- Permet le Network-Connectivity-as-a-Service pour d'autres services d'OpenStack, comme Compute.
- Fournit une API utilisateur pour définir les réseaux et les attachements à ces réseaux.
- Possède une architecture modulaire qui permet le support de la plupart des fournisseurs et des technologies réseau.

e. Le Dashboard (Horizon) :

Fournit un portail libre-service de type web permettant d'interagir avec les services sous-jacents d'OpenStack, comme le lancement d'une instance, l'attribution d'adresses IP ou la configuration des contrôles d'accès.

f. Le service de Stockage Bloc (Cinder) :

Fournit un stockage bloc persistant aux instances en cours d'exécution. Son architecture basée sur des drivers de type plugin facilite la création et la gestion des dispositifs de stockage bloc.

g. Le service Stockage Objet (Swift) :

Stocke et récupère des objets de données non structurées via une API RESTful basée sur HTTP. Le service est hautement tolérant aux pannes avec sa réplication de données et son architecture de type scale-out. Son implémentation diffère des serveurs de fichiers à répertoires montables. Le service écrit les objets et les fichiers vers plusieurs disques, en s'assurant que les données sont répliquées sur un cluster de serveurs.

h. Le service Télémétrie (Ceilometer) :

Surveille et mesure un cloud OpenStack dans un but de facturation, de mesure de performances, de scalabilité et de statistiques.

i. Le service Orchestration (Heat) :

Orchestre de nombreuses applications de cloud composites en utilisant soit le format de template natif HOT ou le format CloudFormation d'AWS, soit au travers d'une API REST native OpenStack, soit au travers d'une API compatible avec CloudFormation [45].

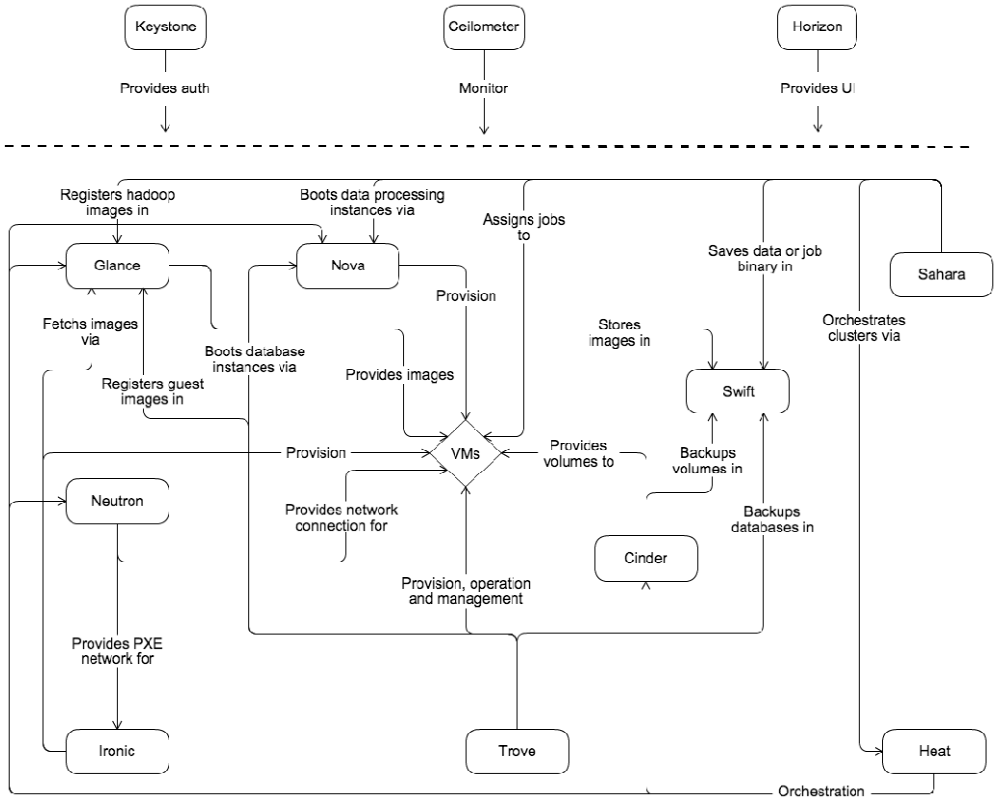

Figure 4.1: Relations entre les services OpenStack [45].

4.3. Mise en place de la solution :

Il existe de nombreuses façons pour déployer OpenStack, chacune d'entre elles ajoutant sa propre valeur au système d'exploitation du cloud. L'installation mononeud est particulièrement utile à des fins d'évaluation et pour se familiariser avec son fonctionnement. Pour une utilisation réelle en production, OpenStack doit être installé et configuré sur plusieurs systèmes ou nœuds.

L'architecture choisie comporte deux nœuds installés sur des machines virtuelles sous Virtual Box.

- Noeud de contrôleur : c'est le noeud où la plus grande partie des services partagés et d'autres outils sont exécutés. Le noeud de contrôleur contient le service d'identité, le magasin d'images et le tableau de bord. En outre, le service de gestion des calculs Nova ainsi que le serveur Neutron sont également configurés dans ce noeud.
- Noeud de calcul : c'est le noeud où les instances de machine virtuelle sont installées. Ce noeud exécute le démon de réseau et celle de calcul qui gère les instances de machine virtuelle concernées.

Chaque nœud inclus le système d'exploitation **Ubuntu**, version **16.04**, avec la configuration matériel suivante :

Nœuds	Processeurs	Mémoire	Stockage
Contrôleur	4 cœurs (2,40 GHz)	4 Go	20 Go
Calcul	2 cœurs (2,40 GHz)	2 Go	20 Go

Table 1.6: Configuration materiels.

4.3.1. Préparation de l'environnement :

Configuration réseau:

Les deux nœuds contrôleur et calcul nécessitent un accès internet pour des besoins administratifs comme l'installation des packages, de mises à jour de sécurité, le DNS et le NTP. En plus deux autres interfaces au minimum sont requises pour le réseau fournisseur (Provider) et le réseau de gestion (Management). Ces interfaces sont configurées comme la figure en dessus indique.

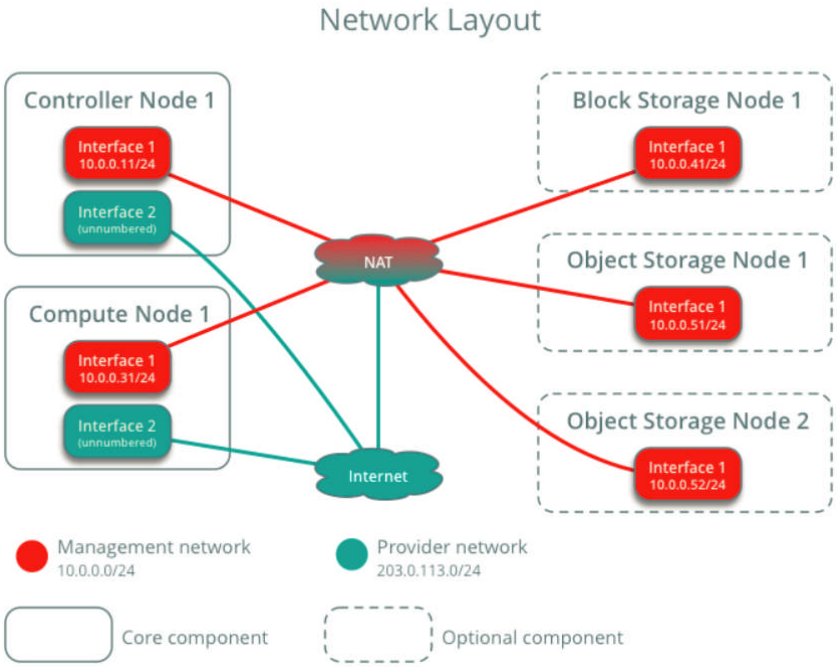

Figure 4.2: Configuration réseaux [45].

Configuration de NTP (Network Time Protocol):

Pour synchroniser les services entre les nœuds, on utilise l'outil Chrony, une implémentation de NTP. Il est recommandé de configurer le contrôleur pour pointer vers des serveurs plus précis (stratum inférieur) et les autres nœuds pour pointer vers le contrôleur.

 # apt install nova-compute

L'adresse ip du serveur (81.168.77.149 pour notre serveur) doit être précisée dans le fichier : **etc/chrony/chrony.conf.**

Pour permettre au nœud de calcul de se connecter au démon chrony du contrôleur, Il faut ajouter la ligne ci-dessus au même fichier dans le noued de calcul :

```
#pool 1.ubuntu.pool.ntp.org iburst maxsources 1
#pool 2.ubuntu.pool.ntp.org iburst maxsources 2
server controller iburst
# This directive specify the location of the file containing ID/key pairs for
# NTP authentication.
```

Installation de la base de données SQL :

La plupart des services OpenStack utilisent une base de données SQL pour stocker des informations. La base de données tourne généralement sur le contrôleur. Pour notre cas nous avons utilisé MariaDB:

 # apt install mariadb-server python-pymysql

Pour permettre aux autres nœuds d'accéder au contrôleur via le réseau de management, il faut définir le fichier : **/etc/mysql/mariadb.conf.d/99-openstack.cnf** comme suit :

```
 GNU nano 2.9.3        mysql/mariadb.conf.d/99-openstack.cnf        Modified

[mysqld]
bind-address = 10.0.0.11

default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

Installation de RabbitMQ:

OpenStack utilise une file de messages pour coordonner les opérations et les informations de statut entre les services. Le service de file de messages tourne en général sur le contrôleur. OpenStack supporte plusieurs services de file de messages dont RabbitMQ, Qpid, et ZeroMQ.

 # apt install rabbitmq-server

Creation de l'utilisateur stack:

 # rabbitmqctl add_user openstack root
 # rabbitmqctl set_permissions openstack ".*" ".*" ".*"

Installation de Memcached :

Le mécanisme d'authentification du service d'identité utilise Memcached pour mettre en cache les jetons. Ce service Memcached s'exécute généralement sur le nœud contrôleur.

 # apt install memcached python-memcache

Installations des services d'Openstack :

Le nœud contrôleur exécute le service d'identité (Keystone), le service d'image (Glance), les parties de gestion des services Compute (Nova) et réseau (Neutron), et le tableau de bord (Dashboard). Il inclut également des services de support tels qu'une base de données SQL ou la file d'attente de messages (message queue).

Le nœud calcul exécute la partie hyperviseur du service Compute qui fait fonctionner les instances. Par défaut, ce service utilise l'hyperviseur KVM. Le nœud de calcul héberge également un agent du service de réseau qui connecte les instances aux réseaux virtuels et fournit des services de firewall aux instances via les groupes de sécurité.

4.3.2. Installation de Keystone:

Avant d'installer les packages il faut créer la base de données de Keystone :

```
MariaDB [(none)]> CREATE DATABASE keystone;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' \
    -> IDENTIFIED BY 'kspass';
root@controller:/home/ar# GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' \
> IDENTIFIED BY 'kspass';
```

Installation des packages et population de la base de données:

 # apt install keystone

 # su -s /bin/sh -c "keystone-manage db_sync" keystone

Initialiser l'archive de clés (Fernet) et effectuer le Bootstrap de service :

```
root@controller:/home/ar# keystone-manage fernet_setup --keystone-user keystone
--keystone-group keystone
root@controller:/home/ar# keystone-manage bootstrap --bootstrap-password ADMIN_P
ASS \
>   --bootstrap-admin-url http://controller:5000/v3/ \
>   --bootstrap-internal-url http://controller:5000/v3/ \
>   --bootstrap-public-url http://controller:5000/v3/ \
>   --bootstrap-region-id RegionOne
```

4.3.3. Installation de Glance:

Création de la base de données glance :

```
MariaDB [(none)]> CREATE DATABASE glance;
Query OK, 1 row affected (0.13 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' \
    ->   IDENTIFIED BY 'glpass';
Query OK, 0 rows affected (0.36 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' \
    ->   IDENTIFIED BY 'glpass';
Query OK, 0 rows affected (0.00 sec)
```

```
ay@Controller:/etc$ openstack user create --domain default --password-prompt gla
nce
User Password:
Repeat User Password:
+---------------------+----------------------------------+
| Field               | Value                            |
+---------------------+----------------------------------+
| domain_id           | default                          |
| enabled             | True                             |
| id                  | b4968e792f32475ebb3dfa83b3c4a0bc |
| name                | glance                           |
| options             | {}                               |
| password_expires_at | None                             |
+---------------------+----------------------------------+
```

Création de l'utilisateur glance :

Création de service glance :

```
ay@Controller:/etc$ openstack service create --name glance \
>   --description "OpenStack Image" image
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Image                  |
| enabled     | True                             |
| id          | 3218938344a04ef1a12283382b7ec38d |
| name        | glance                           |
| type        | image                            |
+-------------+----------------------------------+
```

Installation des packages :

 # apt install glance

Définir la connexion avec la base de données dans le fichier : **/etc/glance/glance-api.conf**

```
[database]
connection = mysql+pymysql://glance:glpass@controller/glance

# connection = sqlite:////var/lib/glance/glance.sqlite
```

Population de la base de donnees :

 # su -s /bin/sh -c "glance-manage db_sync" glance

4.3.4. Installation de Nova :

Nœud de contrôleur :

Création de bases de données de Nova:

```
MariaDB [(none)]> CREATE DATABASE nova_api;
Query OK, 1 row affected (0.26 sec)

MariaDB [(none)]> CREATE DATABASE nova;
Query OK, 1 row affected (0.13 sec)

MariaDB [(none)]> CREATE DATABASE nova_cell0;
Query OK, 1 row affected (0.00 sec)
```

Création de l'utilisateur nova :

```
ay@controller:/etc/openrc$ openstack user create --domain default --password-pro
mpt nova
User Password:
Repeat User Password:
+---------------------+----------------------------------+
| Field               | Value                            |
+---------------------+----------------------------------+
| domain_id           | default                          |
| enabled             | True                             |
| id                  | ca26cc5f6f29499ea26203b24053cf1d |
| name                | nova                             |
| options             | {}                               |
| password_expires_at | None                             |
+---------------------+----------------------------------+
```

Création de service nova :

```
ay@controller:/etc/openrc$ openstack service create --name nova \
>   --description "OpenStack Compute" compute
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Compute                |
| enabled     | True                             |
| id          | d4291fbc6df4449897e1d64ea9f61981 |
| name        | nova                             |
| type        | compute                          |
+-------------+----------------------------------+
```

Installation des packages :

**# apt install nova-api nova-conductor nova-consoleauth **
nova-novncproxy nova-scheduler nova-placement-api

Configuration de fichier: **/etc/nova/nova.conf** :

```
[DEFAULT]
transport_url = rabbit://openstack:root@controller
my_ip = 10.0.0.11
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver

[api]
auth_strategy = keystone

[api_database]
connection = mysql+pymysql://nova:root@controller/nova_api

[database]
connection = mysql+pymysql://nova:root@controller/nova

[keystone_authtoken]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = root
```

Population de la base de données :

 # su -s /bin/sh -c "nova-manage api_db sync" nova
 # su -s /bin/sh -c "nova-manage db sync" nova

Nœud de calcul :

Installation des packages :

 # apt install nova-compute

Configuration de fichier : **/etc/nova/nova.conf :**

```
[DEFAULT]
transport_url = rabbit://openstack:root@controller
my_ip = 10.0.0.11
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver

[api]
auth_strategy = keystone

[glance]
api_servers = http://controller:9292

[keystone_authtoken]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = root
```

4.3.5. Installation de Neutron :

Nœud de contrôleur :

Création de la base de données :

```
MariaDB [(none)] CREATE DATABASE neutron;

MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' \
  IDENTIFIED BY 'root';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' \
  IDENTIFIED BY 'root';
```

Création de l'utilisateur neutron :

```
$ openstack user create --domain default --password-prompt neutron

User Password:
Repeat User Password:
+---------------------+----------------------------------+
| Field               | Value                            |
+---------------------+----------------------------------+
| domain_id           | default                          |
| enabled             | True                             |
| id                  | fdb0f541e28141719b6a43c8944bf1fb |
| name                | neutron                          |
| options             | {}                               |
| password_expires_at | None                             |
+---------------------+----------------------------------+
$ openstack role add --project service --user neutron admin
```

Création de service neutron :

```
$ openstack service create --name neutron \
  --description "OpenStack Networking" network

+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Networking             |
| enabled     | True                             |
| id          | f71529314dab4a4d8eca427e701d209e |
| name        | neutron                          |
| type        | network                          |
+-------------+----------------------------------+
```

Installation des composants de réseau :

**# apt install neutron-server neutron-plugin-ml2 **

**neutron-linuxbridge-agent neutron-l3-agent neutron-dhcp-agent **

neutron-metadata-agent

Configurer le composant serveur dans le fichier: **/etc/neutron/neutron.conf** :

```
[DEFAULT]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = true
transport_url = rabbit://openstack:root@controller
auth_strategy = keystone
notify_nova_on_port_status_changes = true
notify_nova_on_port_data_changes = true

[database]
connection = mysql+pymysql://neutron:root@controller/neutron

[keystone_authtoken]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = root
```

Configurer le service Compute pour utiliser le service Réseau « **/etc/nova/nova.conf** » :

```
[neutron]
# ...
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = root
service_metadata_proxy = true
metadata_proxy_shared_secret = root
```

Autres configurations :

- Le plugin ML2 utilise le mécanisme Linux bridge pour construire l'infrastructure réseau virtuel de niveau-2 (bridging et switching) pour les instances.
- L'agent Linux bridge construit l'infrastructure de réseau virtuel layer-2 (bridging and switching) pour les instances et gère les groupes de sécurité.
- L'agent Layer-3 (L3) fournit le routage et les services NAT aux réseaux virtuels self-services.
- L'agent DHCP fournit les services DHCP aux réseaux virtuels.

Noeud de calcul:
Installation des composants:

 # apt install neutron-linuxbridge-agent
Configurer le composant general « **/etc/neutron/neutron.conf** »:

```
[DEFAULT]
transport_url = rabbit://openstack:root@controller
auth_strategy = keystone

[keystone_authtoken]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = root
```

Configurer l'agent Linux bridge « **/etc/neutron/plugins/ml2/linuxbridge_agent.ini** » :

```
[linux_bridge]
physical_interface_mappings = provider:PROVIDER_INTERFACE_NAME

[vxlan]
enable_vxlan = true
local_ip = 10.0.0.11
l2_population = true

[securitygroup]
enable_security_group = true
firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

Configurer le service Compute pour utiliser le service Réseau « **/etc/nova/nova.conf** » :

```
[neutron]

url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = root
```

4.3.6. Installation de Dashboard :

Installation des packages:

 # apt install openstack-dashboard

Configuration de fichier: **/etc/openstack-dashboard/local_settings.py**

```
OPENSTACK_HOST = "controller"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"

ALLOWED_HOSTS = ['one.example.com', 'two.example.com']

SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

CACHES = {
    'default': {
         'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
         'LOCATION': 'controller:11211',
    }
}
```

4.1. Création d'une instance:

Pour accéder au service Dashboard via le navigateur web on utilise l'adresse http://controller/horizon, avec les credentials de l'utilisateur admin.

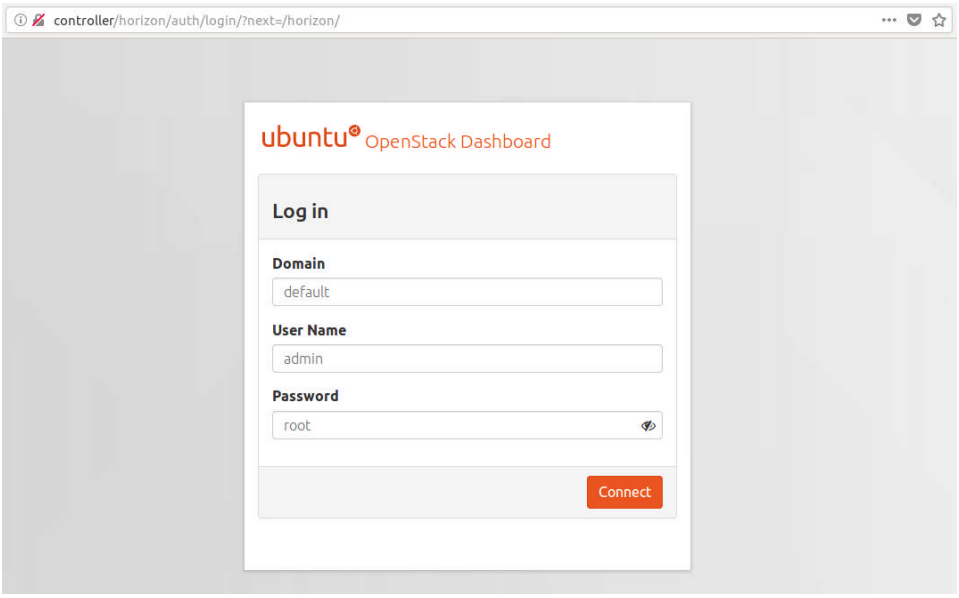

Figure 4.3: Page d'authentification de Dashboard.

Apres avoir authentifié, on peut maintenant créer un projet qui contient les ressources (CPU, RAM, réseau et espace de stockage). On va créer aussi un compte utilisateur, qu'ils vont utiliser pour accéder à leur espace Cloud.

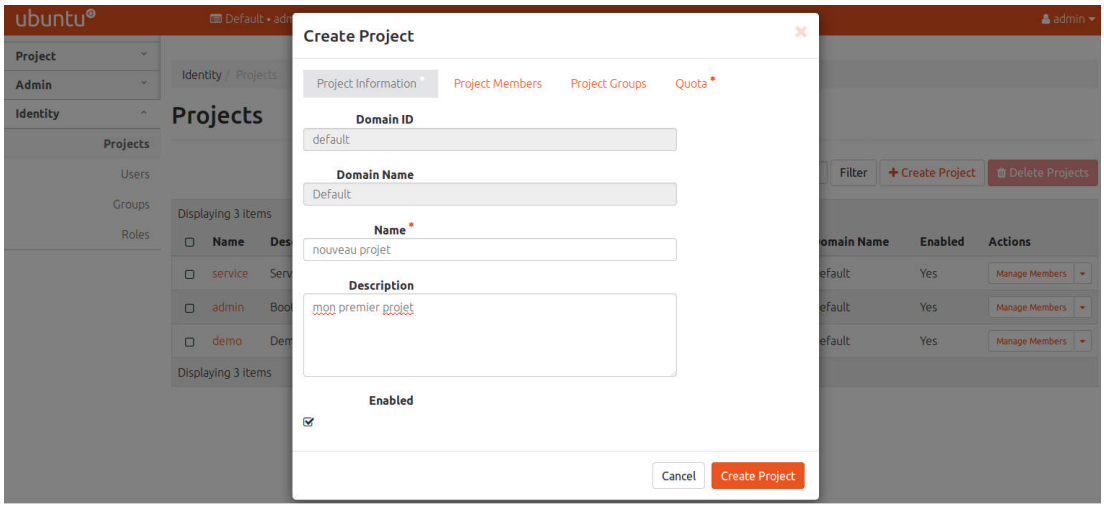

Figure 4.4: Création d'un projet.

L'onglet « Quota » permet de définir les ressources, comme montré dans la figure 40 :

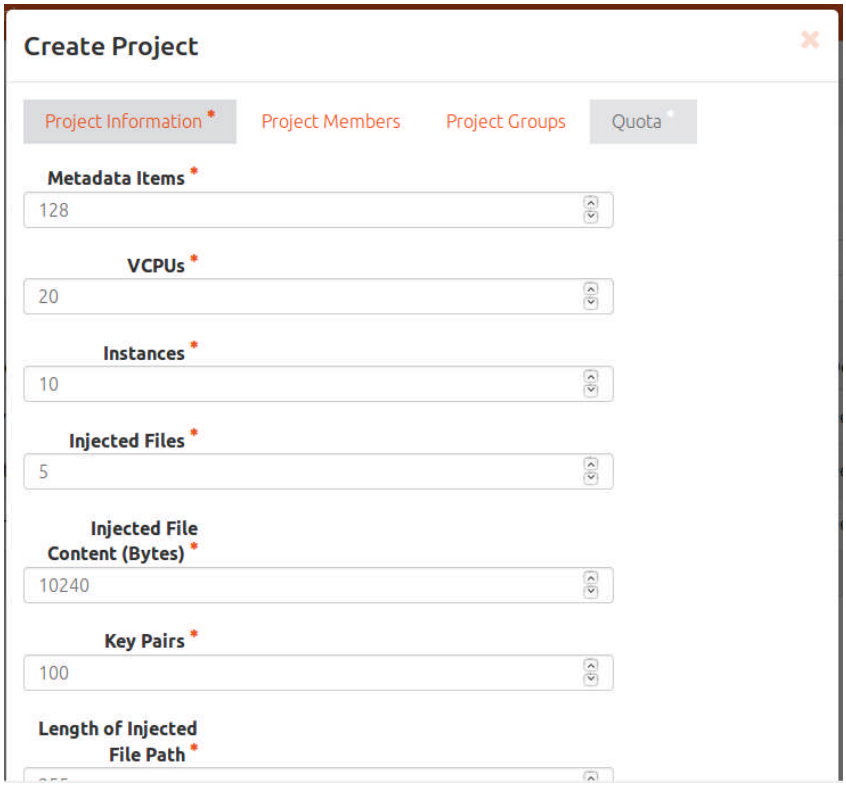

Figure 4.5: Fenêtre de définition des ressources.

Il est possible de créer un utilisateur qui sera membre de ce projet, il peut exploiter et manipuler les ressources qu'on lui a attribuées.

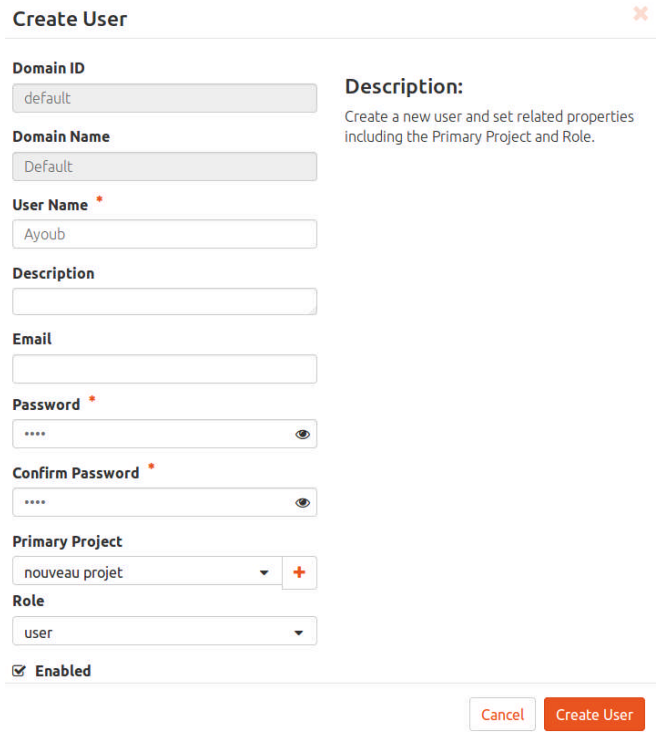

Figure 4.6: Création d'un utilisateur.

Pour accéder et exploiter les ressources affectées, il faut s'authentifié avec un utilisateur membre de ce projet. Dans la page d'accueil, on peut voir les ressources affectées, et leur taux d'utilisation.

La figure en dessus représente la page d'accueil pour les utilisateurs de projet.

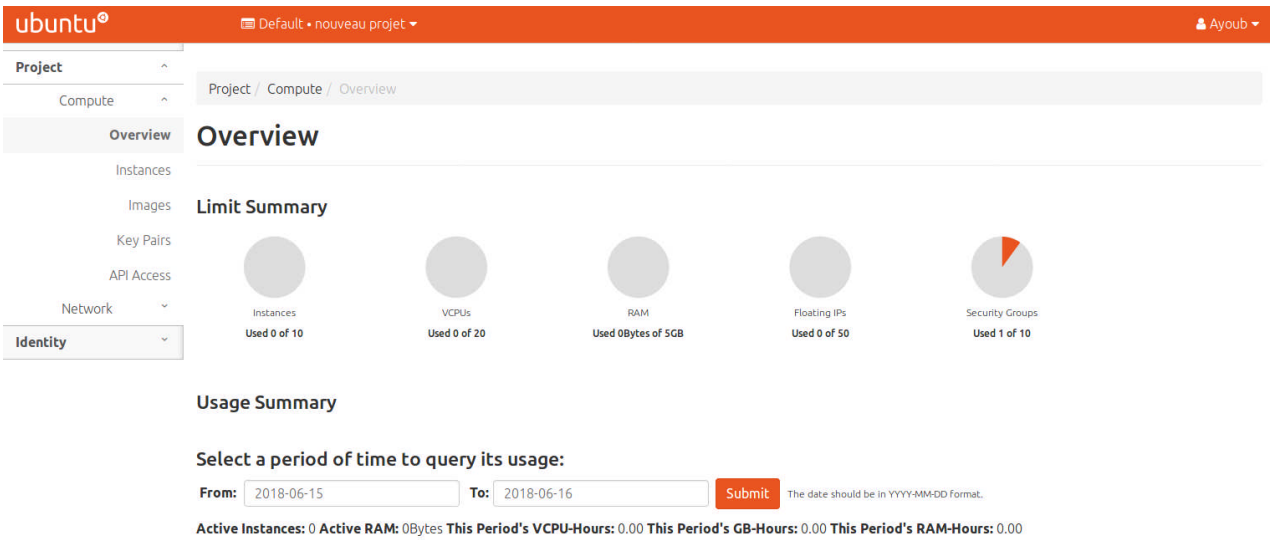

Figure 4.7: Page d'accueil de l'utilisateur.

Création d'un groupe de sécurité :

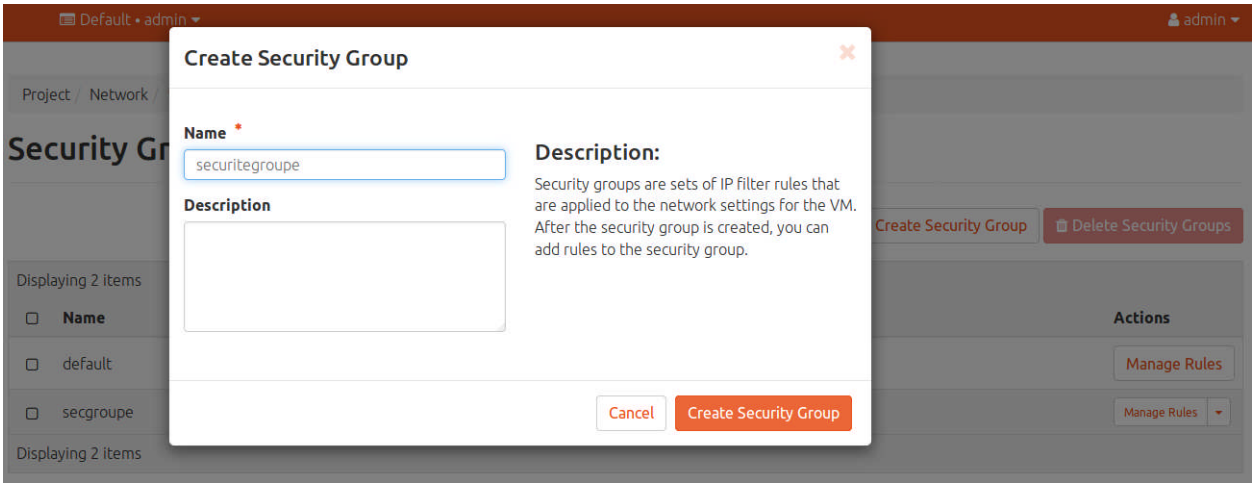

Figure 4.8: Création d'un groupe de sécurité.

Appuyez sur le bouton Modifier les règles à côté du groupe de sécurité que vous souhaitez ajouter des règles / modifier. Nous allons utiliser le TicCloudsec Groupe de sécurité, d'abord ajouter une règle pour permettre les connexions SSH entrantes sur le port 22.

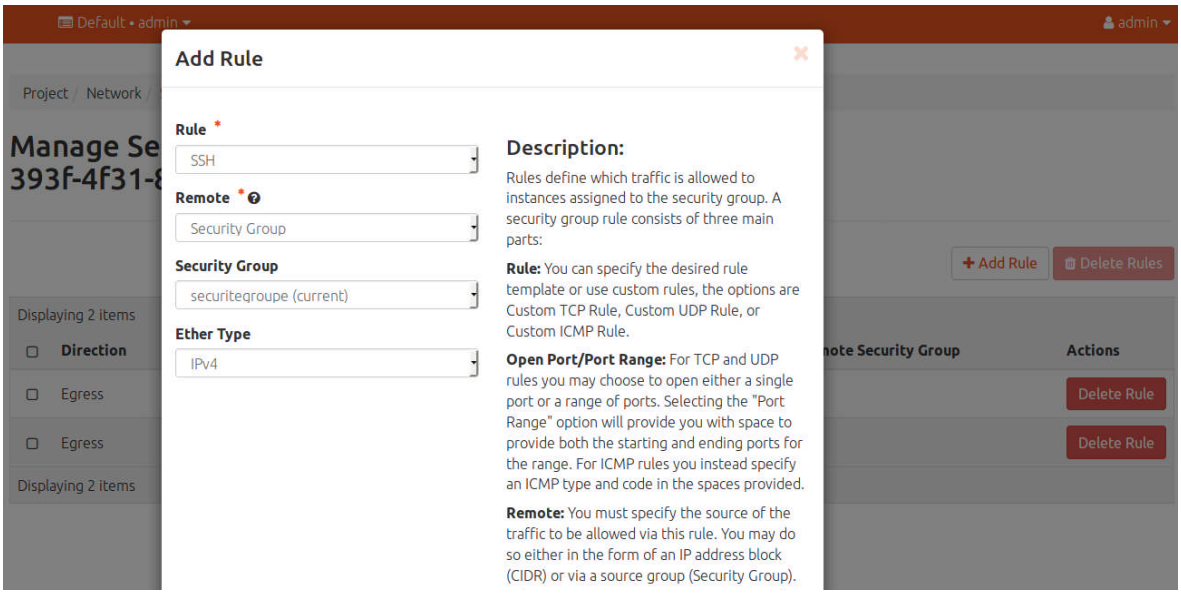

Figure 309: Définition des règles.

Ensuite, nous avons à générer une paire de clés qui seront utilisés pour authentifier les utilisateurs dans les machines virtuelles. Cliquez sur l'onglet "paires de clés" sur "l'accès et la sécurité" et cliquez sur "Créer une paire de clés".

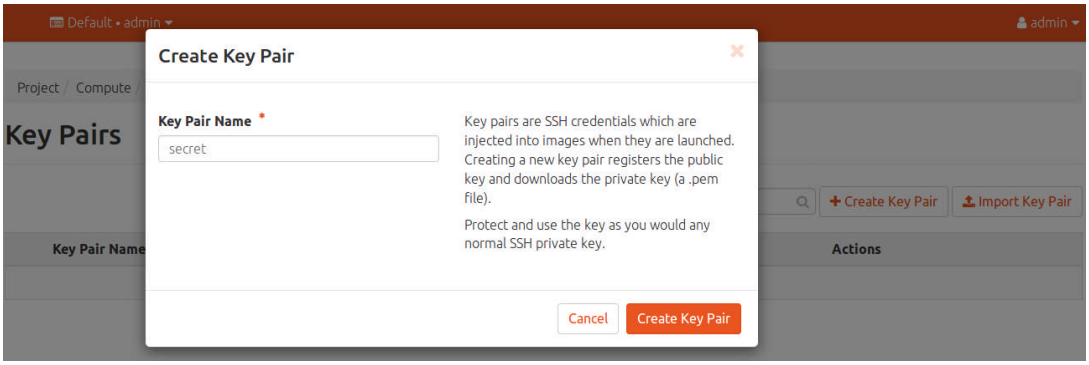

Figure 31.10: génération des clés.

Téléchargez et enregistrez le fichier de clé. Il sera utilisé pour se connecter à des machines virtuelles à partir de l'extérieur.

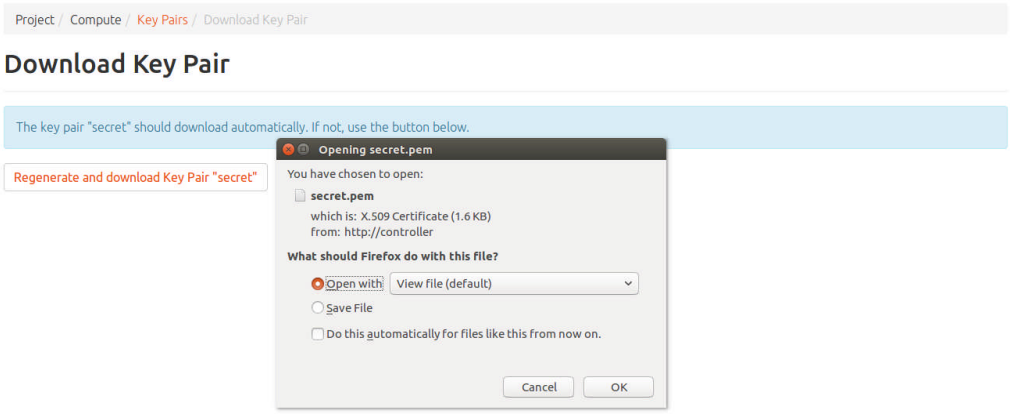

Figure 3211: téléchargement de la clé.

Maintenant, nous pouvons créer une instance en utilisant le groupe de sécurité et la paire de clés que nous avons créés. Cliquez sur le lien "Instances" sous l'onglet "Project" et cliquez sur "Lancer instance".

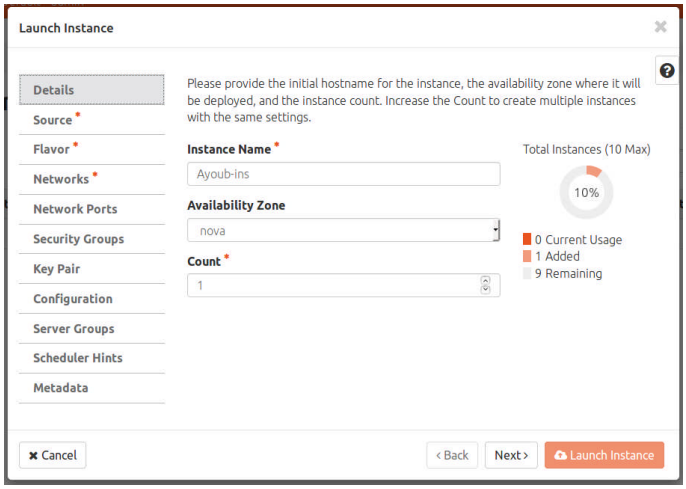

Figure 3312: Création de l'instance.

Dans l'interface utilisateur, vous pouvez configurer l'exemple en fournissant un nom, taille, etc. sous l'onglet "Détails". Sous l'onglet "Accès et Sécurité" nous pouvons choisir la paire de clés et le groupe de sécurité que nous créons ci-dessus.

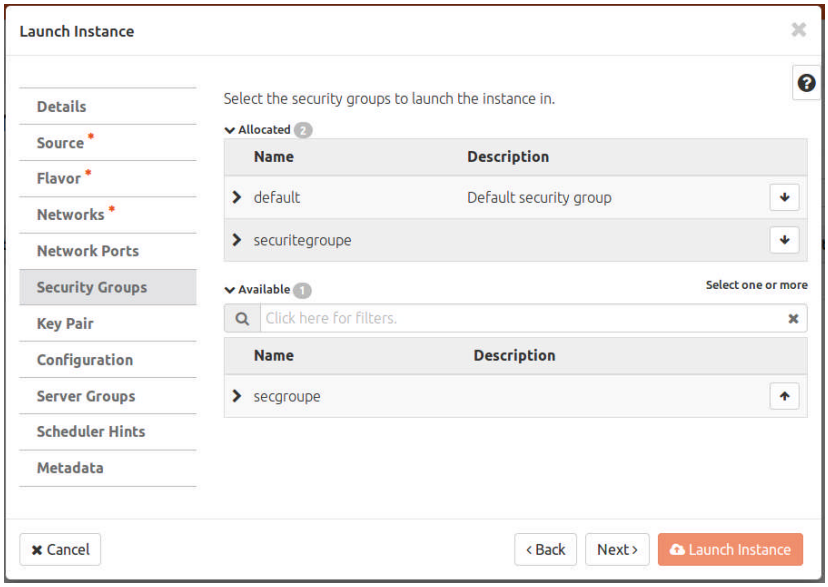

Figure 34.13 : sélection de groupe de sécurité.

4.2. Conclusion :

Le Cloud Computing représente un nouveau défi dans le monde informatique et plusieurs solutions sont proposées pour déployer un tel système. Dans ce chapitre, nous avons présenté OpenStack comme solutions open sources et d'une manière non exhaustive. On a présenté les outils logiciels et matériels ainsi que toutes les étapes et la démarche à suivre pour installer les différents composants d'OpenStack. Malgré que l'installation semble facile, mais on a rencontré beaucoup de problèmes avant d'arriver à finaliser le déploiement.

Conclusion générale

Ce travail s'est intéressé aux challenges liés au Cloud Computing, en particulier à la problématique de la sécurité des données. Nous avons commencé à traité les notions et les concepts de base de Cloud Computing, dans un même contexte nous avons parlé sur différents aspects de sécurité ainsi que la cryptographie. Nous avons aussi proposé solution pour la réalisation d'un échange de données sécurisé.

Nous avons fait par la suite l'installation et la configuration d'OpenStack qui a nécessité des prérequis matériels et logiciels. La configuration de notre solution a été réalisée sous le système d'exploitation Ubuntu 16.04 Desktop qui a été installé sur des machines virtuelle, le logiciel de virtualisation utilisé est VirtualBox.

Ce projet a été pour nous une chance et une formidable opportunité de découvrir un environnement informatique nouveau, complexe et vaste, ce qui nous a permis d'acquérir de l'expérience en administration systèmes et réseaux et d'approfondir nos connaissances dans le domaine de la virtualisation et du Cloud Computing. Mais et surtout d'acquérir les bons réflexes que doit avoir tout administrateur réseau.

Finalement, nous espérons qu'une grande partie des objectifs fixés au départ ont été réalisés.

Bibliographie :

[2]. Manyika, et al. Disruptive technologies : Advances that will transform life, business, and the global economy. 2013.

[3]. Contribution à la sécurité du Cloud Computing : Application des algorithmes de chiffrement pour sécuriser les données dans le Cloud Storage. KARTIT, Zaïd. 2016.

[4]. Cisco Global Cloud Index: Forecast and Methodology,2016–2021. 2018.

[5]. Above the Clouds: A Berkeley View of Cloud. Armbrust et al. 2009.

[6]. Future Generation Computer Systems. Buyya et al. 2009.

[7]. Grid and Cloud Computing.Katarina Stanoevska-Slabeva, Thomas Wozniak. 2009.

[8]. The NIST Definition of Cloud. Peter Mell, Timothy Grance. 2011.

[10]. Advantages of Autonomic Computing over Cloud Computing Comparative Analysis. Samah Mawia Ibrahim Ome, et al. 2014.

[12]. Virtualisation et cloud computing.center, intel IT. 2013.

[14]. Cloud Computing. Yumin Danny Z, et al. 2002.

[15]. Optimization of Security as an Enabler for Cloud Services and Applications.Deshpande, Varun M, et al. 2018.

[21]. Cloud computing Concept vaporeux ou réelle innovation ?.Loeckx, Johan et Ogonowski, Grégory. 2011.

[23]. Cyber Security on Cloud. Royston, et al. 2017.

[24]. Study on the security models and strategies of cloud computing. Chea, Jianhua, et al. 2011.

[25]. Optimization of Security as an Enabler for Cloud Services and Applications.Deshpande, Varun M., Nair, M ydhili K. et Ayush, Bihani. 2018.

[27]. Security Challenges in Cloud Computing. Ertaul, L., Singhal, S. et Saldamli, G. 2010.

[31]. La sécurité dans le Cloud. Winkler, Vic (J.R.). 2011.

[32]. Encyclopedia of Cloud Computing. rajkumar Buyya, et al. 2016.

[33]. Cloud Computing - Concepts, Architecture and Challenges. Jadeja, Yashpalsinh et Modi, Kirit. 2012.

[34]. An Analysis of Cloud. Behl, Akhil et ., Kanika Behl. 2012.

[35]. Enhancing Data Security during Transit in Public Cloud.Irudayasamy, Amalraj, L, Arockiam et N, Veeraragavan. 2013.

[36]. A Study of Securing Cloud Data Using Encryption Algorithms. Mohanaprakash, T. A., et al. 2018.

[37]. Cryptographie. TUAL, Jean-Pierre.

[39]. Biclique Cryptanalysis of the Full AES. Andrey Bogdanov, at al. *2012.*

[40]. Certificats (électroniques) : Pourquoi ? Comment ?. Archimbaud., Jean-Luc. 2000.

[41]. Cryptographie et Sécurité informatique. Dumont., Renaud. 2009-2010.

[42]. Etude de la sécurité d'algorithmes de cryptographie embarquée vis-`a-vis des attaques par analyse de la consommation de courant.Wurcker., Antoine. 2015.

Webographie :

[1]. cloudtweaks. [En ligne] https://cloudtweaks.com/2011/02/a-history-of-cloud-computing/.

[9]. internetactu. [En ligne] http://www.internetactu.net/2009/04/14/quel-est-lusage-de-nos-ordinateurs-au-quotidien/.

[11]. wst. [En ligne] http://www.wst.univie.ac.atworkgroupssem-nessiindex.phpt=semanticweb.

[13]. Pair à pair. [En ligne] https://fr.wikipedia.org/wiki/Pair_%C3%A0_pair.

[16]. Forrester Research World. [En ligne] 2016. https://www.forrester.com/report/Forrester+Research+World+Cloud+Security+Solutions+Forecast+2015+To+2020+Global.

[17]. fier de coder. [En ligne] https://www.fierdecoder.fr/2015/05/architecture-multi-tenant-une-fausse-bonne-idee/.

[18]. itpro. [En ligne] https://www.itpro.fr/lelasticite-suite-logique-cloud/.

[19]. Le Big Data. [En ligne] https://www.lebigdata.fr/securite-cloud-1101.

[20]. it social. [En ligne] https://itsocial.fr/enjeux/cloud-computing/cloud-public-prive-hybride/7-menaces-securite-cloud/.

[22]. wikipedia. [En ligne] https://fr.wikipedia.org/wiki/Sécurité_du_cloud#Les_normes_dans_le_cloud.

[26]. wikipedia. [En ligne] https://en.wikipedia.org/wiki/Cube_attack.

[28]. Wikipedia . [En ligne] https://fr.wikipedia.org/wiki/OAuth.

[29]. Wikipedia. [En ligne] https://fr.wikipedia.org/wiki/OpenID.

[30]. Wikipédia . [En ligne] https://fr.wikipedia.org/wiki/Secure_Shell.

[38]. Wikipédia . [En ligne] https://fr.wikipedia.org/wiki/Certificat_électronique.

[43]. Wékipedia. [En ligne] https://fr.wikipedia.org/wiki/Cryptanalyse_linéaire.

[44]. Wikipédia. [En ligne] https://fr.wikipedia.org/wiki/OpenStack.

[45]. OpenStack. [En ligne] https://docs.openstack.org/.