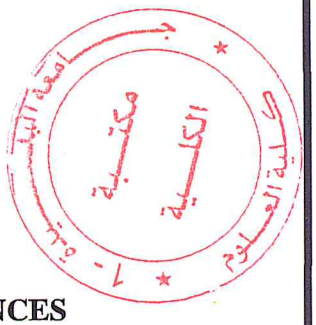
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITÉ SAAD DAHLAB BLIDA FACULTÉ DES SCIENCES

DÉPARTEMENT INFORMATIQUE

Mémoire de fin d'étude

Pour l'obtention d'un diplôme de Master en Informatique

Option : Systèmes Informatiques et Réseaux

Thème

Conception et déploiement d'un système de control d'accès ABE dans une architecture réseau basée SDN

*** Réalisé par :**

- Mohamedi Hadjer
- Ounnoughi Nesma

*** Encadré par :**

- Mr. Derki Mohamed Saddek

Organisme d'accueil : CEntre de Recherche sur l'Information Scientifique et Technique

Jury :

Président :	Mr. Ouled Aissa Ahmed	MAB à l'Université de Saad Dahleb Blida.
Examinateur :	Mme. Ykhlef Hadjer	MAA à l'Université de Saad Dahleb Blida.
Promoteur :	Mme. Boustia Narheman	Professeur à l'Université de Saad Dahleb Blida.

Date de Soutenance : 28/06/2018

Promotion : 2017/2018

Il ne faut pas penser à l'objectif à atteindre,

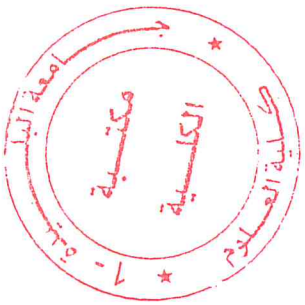
Il faut seulement penser à avancer.

C'est ainsi, à force d'avancer,

Qu'on atteint ou qu'on dépasse ses objectifs

Sans même s'en apercevoir

Bernard Werber



Dédicace

C'est avec profonde gratitude et sincères mots,

Que nous dédions ce modeste travail de fin d'étude à nous chers parents ;

Qui ont sacrifié leur vie pour notre réussite et nous ont éclairé le chemin par leurs conseils judicieux.

Nous espèrons qu'un jour, Nous pourrions leur redonner un peu de ce qu'ils ont fait pour nous,

Que dieu leur prête bonheur et longue vie.

Nous dédions aussi ce travail à.

Toutes nous famille ,

Nous amis,

Tous nouus professeurs qui nous'ont enseigné

Et à tous ceux qui nous sont chers.

Nesma et Hadjer.

Remerciements

Nous remercient tout d'abord le grand Dieu pour nous avoir donnée le courage et la santé pour accomplir ce travail. Ce travail n'aurait pas pu aboutir à des résultats sans l'aide et les encouragements de plusieurs personnes que nous remercient. Nous vifs remerciements accompagnés de toute nous gratitude vont ensuite à notre promotrice Mme Boustia Narimene maître de conférence à l'université de Blida et notre encadreur Derki Mohamed Sedkki attaché de recherche à CERIST, pour les conseils judicieux, l grande disponibilité et pour nous avoir suivie et orientée. Nous remercient gracieusement les membres de jury qui m'ont fait un grand honneur en acceptant la valorisation de ce modeste mémoire. Enfin, que tous ceux qui nous ont aidés et encouragés de près ou de loin dans la concrétisation de ce projet, trouvent ici ma gratitude et mes sincères remerciements.

Résumé

Le SDN -Software defined network- est certainement le sujet le plus recherché depuis ces dernières années qui agite le monde du réseau, la technologie SDN a déclenché un changement radical à long terme dans la conception des réseaux, et le marché s'est rapidement approprié le SDN comme ensemble de solutions/architectures permettant de supprime les frontières existantes entre les mondes des applications et du réseau. Dans un autre terme le SDN est reconnu comme une architecture permettant d'ouvrir le réseau aux applications. En utilisant cet aspect fourni par le SDN la sécurité du stockage des données, serveur, Cloud et IOT peut être fait au niveau de ces applications installées dans le contrôleur SDN. L'un des aspects récents de la sécurité utilisée comme utilitaire de cryptage/décryptage et/ou une politique de contrôle d'accès est l'ABE, le concept de chiffrement par attributs est un mécanisme relativement récent qui combine le chiffrement asymétrique avec le contrôle d'accès, dans l'ABE les clés et les messages chiffrées d'un utilisateur sont étiquetées avec un ensemble d'attributs et une clé particulière qui peut donner l'accès à un données ou décrypté un données chiffré, s'il y a une correspondance entre les attributs d'utilisateur et la politique d'accès . Avec la politique de contrôle d'accès caché permet au propriétaire de données de partager leur donnée cryptée utilisant par exemple le stockage en Cloud avec des utilisateurs autorisés en gardant les politiques de contrôle d'accès aveugle.

Mots clés : SDN (Software Defined Networking), ABE (Attribute Based Encryption), CP-ABE, controller, onos, OpenFlow, OVS (Open vSwitch).

Abstract

Software Defined Network -SDN- is certainly the most searched subject in recent years that is shaking the network world. In network design , the SDN technology has triggered a radical change, and the market has quickly appropriate this technology as set of solutions/architectures to remove existing boundaries between application and network worlds. In another term the SDN is recognized as an architecture for opening the network to applications. Using this aspect provided by the SDN the security of data storage, servers, Cloud and IOT can be done at the level of these applications installed in the SDN controller. One of the recent aspects of security used as an encryption /decryption utility and /or an access control policy is ABE, the concept of attribute based encryption is a relatively recent mechanism that combines asymmetric encryption with access control, in ABE the keys and encrypted messages of a user are tagged with a set of attributes and a particular key that can give access to a data or decrypt encrypted data, if there is a match between the user attributes and the access policy. With the access control policy hidden the data owners are able to share their encrypted data using for example cloud storage with authorized users by keeping the access control policies blind.

Keywords : SDN (Software Defined Networking), ABE (Attribute Based Encryption), CP-ABE, controller, onos, OpenFlow, OVS (Open vSwitch).

Table des matières

Table des figures

Table des matières

Liste des tableaux

Introduction générale

Le nombre des dispositifs et capteurs médicaux augmente de façon considérable, et par conséquence le taux des données générées. Ce qui rend la manipulation et le transport de ces données plus difficile dans des réseaux qui ne sont pas conçu pour ce scenario. De ce fait, la nécessité d'introduire des nouvelles politiques et architectures réseau qui peuvent améliorer la gestion des données est nécessaire .

C'est dans ce contexte que la technologie SDN a vu le jours. Visant à consolider la programmation et réseau, la technologie SDN est un nouveau paradigme d'architecture réseau qui consiste en une séparation entre la partie responsable de l'acheminement des données dans un équipement réseau (routeur, Switch, ...) i .e le plan de contrôle et la partie responsable de l'envoi des données i.e. le plan de données selon la décision du plan de contrôle.

Cette architecture répond aux besoins actuels car elle rend le réseau programmable et permet aux administrateurs de définir leurs politiques de transfert et de gestion des informations. La nature des données échangées dans ce réseau doit être prise en considération, toutes ces informations sont très sensible, ce sont des informations des patients (nom, genre, adresse), médicaments administrés, antécédents familiaux de maladies, hôpitaux visités pour le traitement etc. L'accès à ces informations doit être garanti seulement aux parties autorisées et interdit aux autres, ce dernier nécessite le déploiement d'un contrôle d'accès.

Le contrôle d'accès basé sur l'attributs intervient ici pour résoudre le problème , c'est une technique qui combine le chiffrement asymétrique avec le contrôle d'accès , dans lequel les utilisateurs reçoivent des attributs, et les données sont associées avec des politiques d 'accès, seuls les utilisateurs disposant d'un ensemble d'attributs valide, satisfaisant à la politique d'accès, peuvent accéder aux données .Par exemple, certains dossiers médicaux pourraient être accessible par un neurologue ou par infirmières ayant 10ans d'expérience.

Dans le cadre de notre projet de fin d'études ,notre but est la Conception et déploiement d 'un contrôle d 'accès Attribute based encryption (ABE) dans un réseau basé Software Defined Network (SDN).L'objectif de ce projet et de proposer une architecture réseau basé SDN qui contient au moins un contrôleur qui doit être capable de gérer le réseau pour assurer le transfert des données. En plus, une politique de contrôle d'accès doit être installée au niveau du contrôleur en utilisant la méthode ABE. Le contrôleur sera capable de donner le droit d'accès aux utilisateurs concernés, il peut aussi révoquer ce droit si l'utilisateur n'a pas d'autorisation d'accès.

Notre mémoire retrace les différentes étapes qui ont donné lieu à cette solution. Des recherches bibliographiques ont été menées sur l'approche SDN et le contrôle d'accès ABE. Les chapitres 1et 2 serviront à présenter les résultats de ces recherches et définir les notions théoriques à connaître. Dans le chapitre 3, étude et présentation du matériel. Dans le 4éme chapitre, nous allons présenter la conception et l'architecture de la solution proposée et un guide d'implémentation visant à détailler le déroulement de la mise en place de la solution. Nous conclurons ce document par une conclusion générale et des perspectives.

Chapitre 1

Software Defined Network -SDN-

1.1 Introduction :

Par rapport aux technologies de télécommunication et autres technologies, les technologies réseau ont atteint un certain degré d'évolution à une cadence faible, C'est-à-dire Les réseaux informatiques sont dynamiques et complexes, en conséquence, la configuration et la gestion continuent d'être difficiles.

Ces réseaux comprennent généralement un grand nombre de commutateurs, routeurs, pare-feu et de nombreux types de boîtes moyennes avec de nombreux types d'événements se produisant simultanément. Les opérateurs de réseaux sont responsables de la configuration du réseau pour appliquer diverses stratégies de haut niveau et répondre à la vaste gamme d'événements réseau (p. ex., déplacements de trafic, intrusions) qui peuvent se produire.

La configuration du réseau reste incroyablement difficile car la mise en œuvre de ces politiques de haut niveau nécessite de les spécifier en termes de configuration de bas niveau distribuée.

Les réseaux d'aujourd'hui fournissent peu ou pas de mécanismes pour répondre automatiquement à la vaste gamme d'événements qui peuvent survenir [1].

De ce fait, il est nécessaire d'introduire des nouvelles politiques et architectures réseau qui rendrent le réseau plus agile ,capable d'embrasser le changement ,et qui permet de consolider plusieurs services sur une infrastructure commune, de surveiller de manière intelligente les conditions du réseau et adapter automatiquement la configuration du réseau au besoin et éviter le verrouillage du fournisseur et centraliser la configuration complète du réseau.

Le paradigme SDN (Software Defined Networking) est une des plus récentes initiatives qui vise à répondre à cette problématique de rigidité architecturale [2]. Pour ce faire, SDN découple

le plan de contrôle du plan de données et le centralise dans un point logique programmable appelé contrôleur SDN [3]. Ainsi, en utilisant ce point de contrôle centralisé (logiquement), les administrateurs réseaux sont en mesure de définir et de mettre à jour rapidement le comportement de leur réseau, et cela, "simplement" en (re)programmant le contrôleur SDN via son interface de programmation applicative (API), communément appelée Northbound API.
Dans ce qui suit, nous présentons plus en détails ce nouveau paradigme ainsi que les différents concepts qui le définissent.

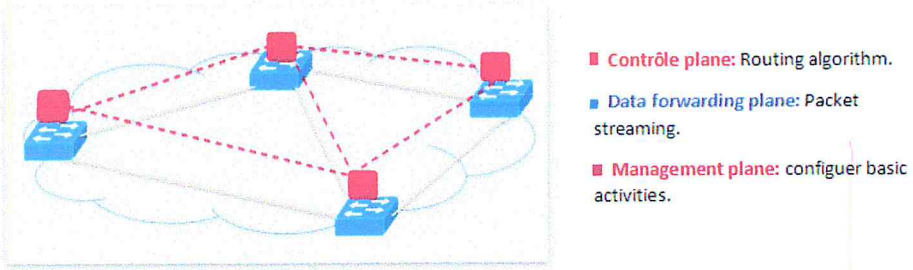


FIGURE 1.1 – Figure 1 : Réseau non programmable [4]

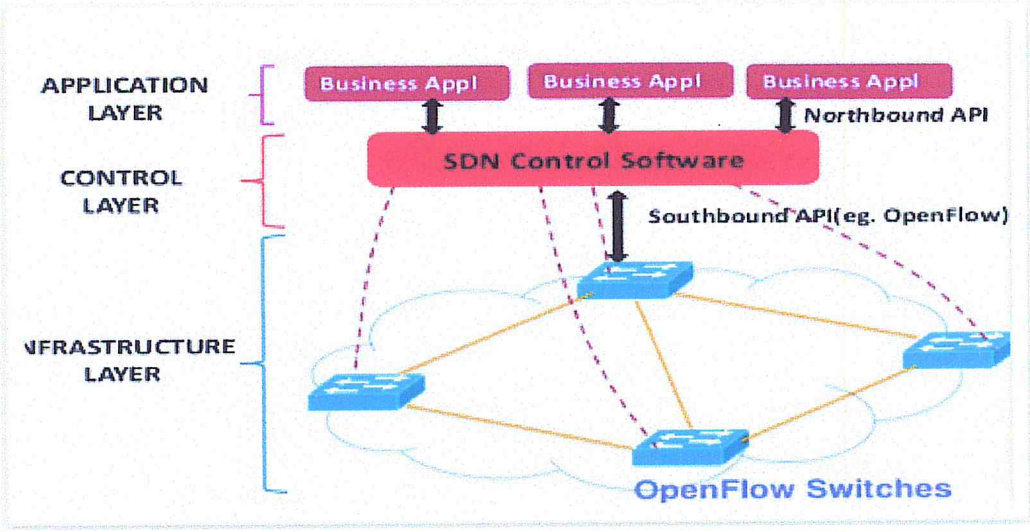


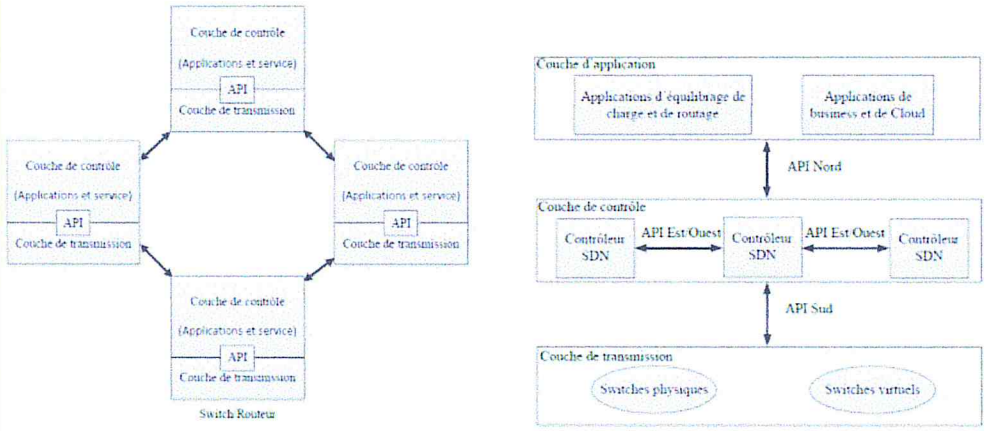
FIGURE 1.2 – Figure 2 : Software defined network -SDN- [5]

1.2 Comparaison entre non programmable et SDN :

les figures (1.1,1.2, 1.3) précédentes décrivent la différence d'architecture entre les réseaux traditionnels et les réseaux SDN :

— **Les Réseaux non programmable :** sont des réseaux statiques et inflexibles. Ils ne sont pas utiles pour les nouvelles entreprises.Ils possèdent peu d'agilité et de flexibilité.Ce sont des appareils Hardware.distribué,plan de contrôle distribué.utilisent des ASIC et des FPGA personnalisés.travaillent en utilisant des protocoles.

— **Software Defined Network :** Ce sont des réseaux programmables pendant le temps de déploiement ainsi qu'à une étape ultérieure en fonction de la modification des exigences. Ils aident les nouvelles entreprises grâce à la flexibilité, l'agilité et la virtualisation.Ils sont configurés en utilisant un logiciel ouvert (open software). plan de contrôle logiquement centralisé.utilisent du silicium marchand.utilisent des API pour configurer selon les besoins.



(a) Réseau traditionnel (b) Réseau SDN

FIGURE 1.3 – Figure 3 : Comparaison entre l'architecture des réseaux traditionnels(a) et celle des réseaux SDN (b) [6]

1.3 Historique des réseaux SDN :

Le début des réseaux SDN a commencé avec le projet Ethane, lancé en 2006 à l'université de Stanford. En effet, le projet Ethane définit une nouvelle architecture pour les réseaux d'entreprises. L'objectif d'Ethane était d'avoir un contrôleur centralisé pour gérer les règles (policies) et la sécurité dans le réseau. Ethane utilise deux composantes : un contrôleur pour décider si un paquet doit être transmis, et un Switch Ethane composé de table et d'une chaîne de communication entre les deux. Ethane était une source d'inspiration pour un système d'exploitation consacré aux réseaux dénommé Nox, et pour un nouveau concept appelé aujourd'hui « réseaux programmés par logiciel » (SDN, Software-Defined Networking)[7],[8].

1.4 Définition :

1.4.1 Définition 1 :

Software Defined networks est une architecture émergente dynamique, gérable, rentable et adaptable, ce qui en fait la solution idéale pour la large bande passante et la nature dynamique des applications d'aujourd'hui. Cette architecture découple les fonctions de contrôle et de transfert de réseau permettant au contrôle du réseau de devenir directement programmable et l'infrastructure sous-jacente à abstraire pour les applications et les services réseau.

1.4.2 Définition 2 :

SDN est une approche d'architecture réseau qui permet au réseau d'être intelligemment et centralement contrôlé, ou ''programmé '', à l'aide d'application logicielle, Cela aide les opérateurs à gérer l'ensemble du réseau de manière cohérente et globale, indépendamment de la technologie de réseau sous-jacente [9].

1.4.3 Définition 3 :

SDN se réfère à la capacité des applications logicielles à programmer des périphériques réseau individuels dynamiquement et donc à contrôler le comportement du réseau dans son ensemble. Boucadair et Jacquenet , soulignent que SDN est un ensemble de techniques utilisées pour faciliter la conception, la livraison et l'exploitation des services de réseau de façon déterministe, dynamique et évolutive [10].

1.4.4 Définition 4 :

SDN est un paradigme qui décrit une architecture réseau dont le plan de données est contrôlé à distance par une entité logique centrale. Plus concrètement, on peut dire qu'une architecture réseau suit le paradigme SDN si, et seulement si, elle vérifie ces quatre points fondamentaux :

- Le plan de contrôle est complètement découplé du plan de données
- Toute l'intelligence du réseau est externalisée dans un point logiquement centralisé appelé contrôleur SDN
- Le contrôleur SDN est un composant programmable.
- La transmission des paquets ne se fait plus suivant la destination, mais par flux, sachant qu'un flux est défini par un ensemble de champs d'en-têtes [11].

1.5 Architecteur et couches SDN :

Dans cette section, nous nous attarderons sur l'architecture conceptuelle de SDN (schématisée par la Figure 3). Nous y détaillons cette architecture et les différentes couches qui caractérisent cette technologie et les interfaces qui permettent à ces couches d 'interagir.

1.6 Architecteur SDN :

SDN définira une architecture réseau où le transfert de données est géré par un plan de contrôle à distance découplée des équipements. Cet aspect de découplement imposera une architecture et le respect de quatre piliers à savoir :

1. **Le plan de commande et de données est découplé :** Les fonctionnalités de contrôle sont supprimées au niveau des dispositifs de réseau qui sont devenus de simples éléments de transfert de paquets.
2. **Les décisions d'expédition sont en fonction du flux au lieu de destination :**Dans le contexte SDN/OpenFlow, un flux est une séquence de paquets entre une source et une destination. Tous les paquets d'un flux reçoivent les mêmes politiques du dispositif de transmission [12],[13]. L'abstraction de flux permet d'unifier le comportement des différents types de périphériques réseau tel que les routeurs, les commutateurs, les pare-feu[?].
3. **Le système de logique (plan de contrôle) est déplacé du matériel vers une entité externe :** Appelé contrôleur SDN ou Network Operating System (NOS), il est une plate-forme logicielle fonctionnant en client/serveur avec le NOS comme serveur. Il fournit l'essentiel des ressources et des abstractions afin de faciliter la programmation des dispositifs.
4. **Le réseau est programmable :** à travers des applications logicielles qui s'exécutent au-dessus du NOS et qui interagissent avec des équipements en dessous du NOS. Ceci est la caractéristique fondamentale du SDN.

Ces piliers impérativement suivis dans le SDN nous permettent d'entrevoir une architecture du SDN comme introduit dans la Figure 4 :

1.7 Les couches SDN :

En faisant abstraction des détails contenus dans chaque plan de la figure 4 et en retenant uniquement les trois (03) plans, on peut dire que, les différents plans/couches sont le plan de données, le plan de contrôle et le plan applicatif, ces plans communiquent entre eux grâce à des interfaces logiciels, dites NorthBound (NBI) et SouthBound (SBI).

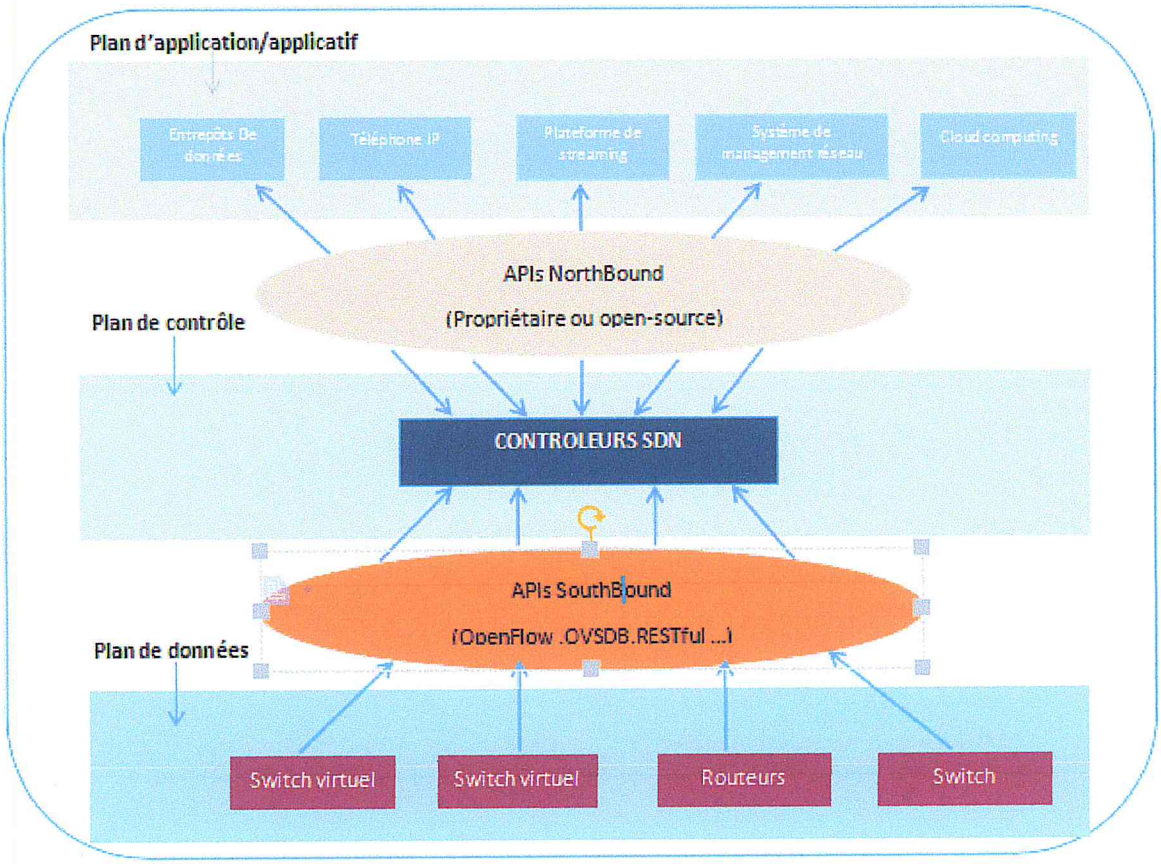


FIGURE 1.4 – Figure 4 :Architecture SDN

1.7.1 Couche Application :

La couche application réside au-dessus de la couche de contrôle, il se compose de services de réseau, d'applications et d'outils d'orchestration qui sont utilisé pour interagir avec la couche de contrôle. Grâce à la couche de contrôle, les applications SDN peuvent accéder facilement à une vue de réseau globale avec un statut instantané via une interface de contrôleurs en direction nord (NorthBound API), les applications SDN peuvent programmer des stratégies pour manipuler les réseaux physiques sous-jacents en utilisant un langage de haut niveau fournie pour la couche de contrôle.

1.7.2 Couche de Contrôle :

La couche de contrôle relie la couche application et la couche d'infrastructure, par, respectivement, les interfaces NBI et SBI. Cette couche consiste en un plan de contrôle centralisé qui est découplé de l'infrastructure physique pour fournir une vue globale centralisée à l'ensemble du réseau. La couche utilisera le protocole OpenFlow pour communiquer avec la couche inférieur i.e. couche d'infrastructure.

— **contrôleur SDN :**Contrôleur est le noyau d'un réseau SDN, il agit comme point de contrôle stratégique sur le réseau SDN, gère le contrôle de flux vers les commutateurs/ routeurs 'ci-dessous' via l'interface SouthBound et les applications et la logique métier au-dessus via l'interface NorthBound pour déployer des réseaux intelligents[?].Le contrôleur SDN utilise des protocoles tels que OpenFlow pour configurer les périphériques réseau, il gère le contrôle de flux pour permettre la mise en œuvre de réseaux intelligents.

Il existe plusieurs type de contrôleurs, on peut énumérer :

— **Beacon** qui est un contrôleur OpenFlow rapide, multiplateforme, modulaire basé sur Java.

— **Floodlight** qui est un contrôleur supporté par BigSwitch ; il est sous licence Apache et écrit en Java.

— **NOX et POX** qui sont les deux premiers contrôleurs OpenFlow. NOX sur C++ et POX sur Python.

— **Ryu** il est un contrôleur SDN capable de configurer les équipements réseaux en utilisant OpenFlow, NetConf et OF-config.

— **Open Daylight** Le projet OpenDaylight est un projet open source pour le Software

Defined Network qui a été annoncé publiquement le 8 Avril 2013. C'est un projet de collaboration hébergé par la Fondation Linux. Il contient un contrôleur modulaire et souple.

1.7.3 Couche d'Infrastructure :

Il s'agit de la couche de base constituée à la fois de périphériques réseau physiques et virtuels, en d'autres terme cette couche consiste en des dispositifs de commutation tel que des commutateurs et des routeurs qui sont interconnectés pour formuler un seul réseau. Tous les périphériques réseau implémentent le protocole OpenFlow pour implémenter des règles d'acheminement du trafic.

— **Commutateur SDN :** Il existe plusieurs types de commutateurs logiciels et matériels SDN disponibles. Les commutateurs logiciels peuvent être utilisés pour exécuter des simulations de test SDN ainsi que pour développer des protocoles et des services. Open vSwitch, par exemple, fait maintenant partie du noyau Linux (à partir de la version 3.3) et facilite à la fois la passerelle virtuelle entre les services physiques et virtuels ainsi qu'une plateforme de test à utiliser en tandem avec les outils de simulation de topologie SDN comme Mininet. En plus des commutateurs logiciels, des géants industriels tels que IBM, HP et NEC ont également mis sur le marché des commutateurs physiques de classe opérateur [16].

1.8 Les API (Application Programming Interface) :

Le « NorthBound » et le « SouthBound » Interfaces sont deux abstractions importantes du SDN, spécifie comment les composants logiciels doivent interagir les uns avec les autres. Les Api permettent de mettre en œuvre des fonctions réseau de base telles que le calcul de chemin, l'évitement de boucle, le routage, la sécurité et de nombreuses autres tâches.

1.8.1 Southbound API :

Cette interface, communément appelée SouthBound interface (SBI), relie le plan de données avec le plan de contrôle. Elle sert à communiquer les informations aux commutateurs et routeurs qui constituent le réseau. L'API la plus communément épandue est OpenFlow qui permet la communication entre nœuds de réseau, ce qu'est les composants de niveau inférieur, cela

facilitent le contrôle efficace du réseau et permettent au routeur d'identifier la topologie du réseau, de déterminer les flux du réseau et permettent aussi au contrôleur SDN d'effectuer des modifications dynamiques en fonction des demandes et des besoins en temps réel.

1.8.2 Northbound API :

Communément appelée NorthBound interface(NBI), elle représente l'interface entre le plan applicatif et le plan de contrôle. Elle dépend de l'application qui tourne au niveau du plan applicatif. Contrairement à la SBI (Standad OpenFlow), la NBI n'a pas été standardisée car les besoins coté NorthBound varient selon l'application et l'approche SDN adopté. Ceci dit, des projets ont vu le jour dans ce sens tel que le Intent NBI de HUAWEI qui vise à créer une seule interface qui interprète les demande des applications en faisant abstraction de leurs nature [17].Elle sert principalement de porte pour les applications au reste de l'architecture, ce qui leur permet de visualiser le réseau, d'interagir avec à un haut niveau d'abstraction et de collecter des statistiques permettant de rentabiliser, au maximum, l'infrastructure.

1.9 Protocole OpenFlow :

Protocole OpenFlow été développé par l'Open Networking Foundation (OPF), est la première interface SouthBound et probablement la plus connue . Il a été développé, en premiers temps, dans le souci de séparer des trafics de natures différentes au sein d'un même réseau en exploitant les tables de flux communes à tous les équipements de commutation. Il a, ensuite, été adopté par la communauté scientifique et les constructeurs comme interface de dialogue principale entre les contrôleurs et les commutateurs SDN. On le retrouve dans le plan de données au niveau des commutateurs OpenFlow (par exemple l'Open vSwitch); ceux-ci stockent des tables de flux propre à ce protocole. C'est grâce à ces tables, remplies et modifiées sur ordre du contrôleur, que les paquets sont acheminés dans le réseau. Ces tables de flux contiennent un ensemble d'entrées associant des flux et les actions que doit entreprendre le commutateur OpenFlow sur les paquets à leurs arrivées.

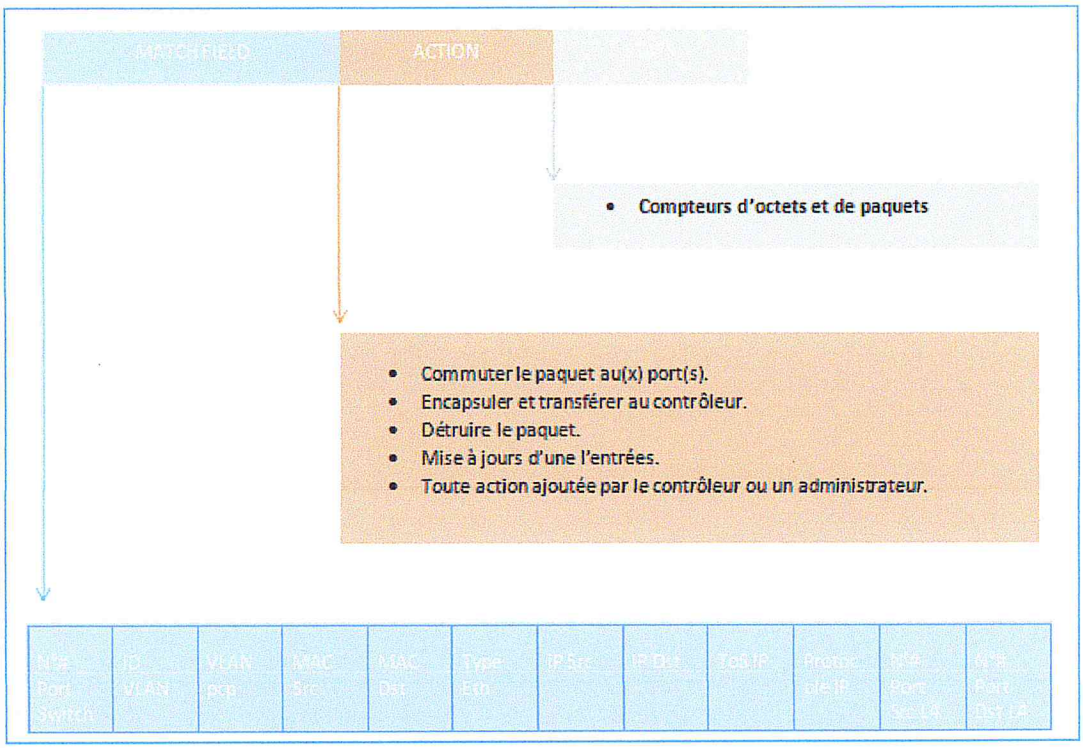


FIGURE 1.5 – Figure 5 :Structure d'une entrées OpenFlow [18]

1.10 les champs d'une entres OpenFlow :

La figure ci-dessous schématise la structure d'une entrée OpenFlow, les types des champs ainsi que les valeurs qu'ils peuvent contenir.

— **match-field (header) :**Sert à identifie le flux en vérifiant les informations contenues au sein des paquets pour l'associer à une action. L'identification peut se faire sur plusieurs niveaux de la couche OSI (Addresse Mac, Adresses IP, Numéros de ports, Vlans, Label MPLS... source ou destinataire soit il) et peut prendre en compte plusieurs données.

— **Action :**Champ contenant l'action à entreprendre après identification du paquet. C'est principalement le couple Header-Action qui définit une entrée de table de flux.

— **Stats :**Contient les valeurs des compteurs de taille et de nombre des paquets propres au flux concerné par l'entrée.

— **Priorities :**Sert à différentier les entrées, tel que dans le cas où un paquet serait concerné par plusieurs entrées de la table de flux, le commutateur applique l'action ayant la priorité la plus élevée.

1.10.1 Messages OpenFlow :

les commutateurs OpenFlow et le contrôleur communiquent entre eux grâce à OpenFlow. Pour cela trois types de messages sont définis pour ce protocole [18]

- **Messages asynchrones :** ces messages sont envoyés des commutateurs au contrôleur. Ils permettent au contrôleur de recevoir des informations à partir du réseau physique. Il existe deux types de messages asynchrones :
  - **Port statuts :**permet de notifier le contrôleur sur le changement d'état des ports (UP ou Down).
  - **Packet-in :** Permet de déléguer au contrôleur la décision sur un paquet qui ne répond à aucune règle de la table de flux ou qui répond à une règle qui a comme action la transmission vers le contrôleur.
- **Message symétriques :** Ces messages sont envoyés dans les deux sens (contrôleur -> commutateur ou l'inverse). Ils permettent d'entretenir le validité du canal de communication entre les commutateurs et le contrôleur. Il existe deux types de messages symétriques :
  - **Hello :** échange entre le contrôleur et un commutateur afin d'établir la connexion entre ceux-ci.
  - **Echo request/reply :** sert à entretenir et vérifier la connexion entre un commutateur et un contrôleur. Ils sont utilisés aussi pour mesurer la bande passante et le délai du canal.
- **Messages de commande :** Ils sont envoyés par le contrôleur afin de récolter des informations et configurer les commutateurs à travers leurs tables de flux. Les trois principaux types de messages de commande sont :
  - **Read-state :** Ils permettent au contrôleur de récolter des informations sur le plan de données telles que les statistiques des flux et la configuration du commutateur avec lequel il interagit.
  - **Modify-state :** Ce sont ces messages qui rendent possible au contrôleur la modification, la création et la suppression d'entrées (règles) dans les tables de flux des commutateurs OpenFlow.
  - **Packet-out :** Ces messages viennent en réponse aux messages Packet-in envoyés par le commutateur. Ils permettent de montrer au commutateur vers quel port sortie le paquet concerné par le message Packet-in doit être acheminé.

1.11 Les cas d'utilisations de SDN :

— **Automatisation** :un niveau d'automatisation accru réduisant les dépenses opérationnelles globales, facilitant le dépannage, réduisant les temps d'arrêt imprévus, facilitant l'application des règles et fournissant les ressources réseau et les charges de travail des applications correspondantes, selon les besoins. [19]

— **Orchestration** : orchestration du contrôle de la gamme complète d'appliances réseau par centaines ou même milliers, comme dans les centres de données ou les environnements de réseau de campus plus importants.[20]

— **Gestion dynamique des ressources** : modification dynamique de la taille du réseau et mise à jour de la topologie et des ressources réseau affectées, ce qui peut être facilité par la virtualisation du réseau.[20]

— **Ouvertes API** :les utilisateurs disposent d'un choix complet de modules d'extension modulaires, offrent l'abstraction, définissent des tâches par API et ne sont pas spécifiquement concernés par les détails d'implémentation. Communication entre deux nœuds à fournir sans spécification du protocole exact.les utilisateurs disposent d'un choix complet de modules d'extension modulaires, offrent l'abstraction, définissent des tâches par API et ne sont pas spécifiquement concernés par les détails d'implémentation. Communication entre deux nœuds à fournir sans spécification du protocole exact. [21]

— **Prise en charge de l'architecture mutualisée** :avec la prolifération croissante des services basés sur le Cloud, les locataires préfèrent un contrôle complet sur leurs adresses, leur topologie, leur routage et leur sécurité et, par conséquent, séparent l'infrastructure hébergée des services hébergés.[20]

— **Plus grande programmable** :une exigence fondamentale de l'approvisionnement réseau actuel est la capacité de changer le comportement et la configuration de l'appareil en temps réel en fonction des conditions de circulation prévalences.

— **Sécurité intégrée** : : capacité d'intégrer des dispositifs de sécurité au sein du réseau, ce qui permet une détection plus précise des incidents de sécurité et une gestion simplifiée. [19]

— **Gestion intégrée des ressources** :en plus des dispositifs de sécurité, intégration transparente de plusieurs services tels que les équilibreurs de charge et les moniteurs de ressources, qui peuvent être provisionnés à la demande et placés dans le réseau en fonction des besoins.[21]

— **Performances améliorées** :un cadre de contrôle offrant la possibilité d'intégrer des solutions d'ingénierie du trafic innovantes, le calcul de capacité, l'équilibrage de charge et un niveau d'utilisation plus élevé pour réduire l'empreinte carbone.[21]

— **Visibilité et surveillance en temps réel** :amélioration de la surveillance en temps réel et de la connectivité des appareils.

— **Virtualisation de réseau** :possibilité de provisionner des ressources réseau sans se soucier de l'emplacement de composants individuels tels que routeurs et commutateurs.

1.12 Étude critique :

1.12.1 Problématique :

— Complexité des réseaux non programmable : L'ajout ou la modification d'équipements et l'implémentation de politiques réseaux sont complexes, longues et peuvent être source d'interruptions de service, ce qui décourage les modifications et les évolutions du réseau.

— passage à l'échelle : L'impossibilité d'avoir un réseau qui s'adapte au trafic a obligé les opérateurs à sur-provisionner leurs réseaux.

— Dépendance aux constructeurs : Les constructeurs réalisent des produits avec des durées de vie et un manque de standard et d'interface ouverte, ce qui réduit, pour les opérateurs réseaux, la possibilité d'adapter le réseau à leurs propres besoins.

ces inconvénients ont contribué à l'apparition des réseaux SDN.

1.12.2 Les avantages de réseau SDN :

Offrant un réseau qui peut dynamiquement s'adapter afin de répondre aux divers besoins des entreprises, SDN fournit également les avantages suivants [22],[23], [24]

— Le déplacement du plan de contrôle dans la partie logiciel permet un accès et une administration dynamique. L'administrateur réseau peut adapter le trafic depuis une console centrale sans avoir à configurer les équipements individuellement. L'administrateur peut changer n'importe quelle règle d'un équipement réseau si nécessaire.

— Livrer, agilité et la flexibilité : l'abstraction du contrôle du relayage permet aux administrateurs d'ajouter dynamiquement le réseau au trafic. SDN aide les organisations

à déployer rapidement de nouvelles applications, les services et l'infrastructure de répondre rapidement à l'évolution des objectifs et objectifs d'affaires.

— Configuration automatique : SDN permet aux administrateurs réseau de configurer, administrer et automatiser. Ces programmes peuvent être développés par eux-mêmes car ils ne dépendent plus de logiciels propriétaires.

— Basé sur des standards ouverts et vendeur-indépendant : L'implémentation à travers des standards ouverts permet de simplifier l'architecture réseau car les instructions sont fournies par un ou plusieurs contrôleurs au lieu de multiple équipements propriétaires.

— Réduire les dépenses d'investissement : SDN limite potentiellement la nécessité d'acheter construite à cet effet, du matériel réseau ASIC, et au lieu supporte pay-as-you-grow modèles

SDN permet un contrôle algorithmique du réseau d'éléments de réseau (tels que les matériels ou logiciels commutateurs / routeurs qui sont de plus en plus programmable, ce qui rend plus facile à concevoir, déployer et gérer les réseaux à grande échelle la capacité d'automatiser le provisionnement et l'orchestration optimise le service, disponibilité et la fiabilité en réduisant le temps de gestion globale et le risque d'erreur humaine.

— Activer l'innovation : SDN permet aux organisations de créer de nouveaux types d'applications, des services et des modèles d'affaires qui peuvent offrir de nouvelles sources de revenus et plus de valeur à partir du réseau.

1.13 Les inconvénients de réseau SDN :

Toutefois si les SDN ont beaucoup d'avantages, leur mise en œuvre n'est pas si aisée qu'il n'y parait, et les opérateurs font face à des difficultés notables dans leur mise en œuvre :

1. la migration vers les SDN est complexe, l'introduction d'un élément de contrôle supplémentaire au cœur du réseau rend la phase projet délicate. Ensuite, un tel projet peut susciter des craintes quant à la sécurité réseau. Les réseaux traditionnels sont conçus afin d'être surs et compartimentés, l'introduction d'un élément logiciel externe, qui plus est à des fins de contrôle/gestion est un élément qui par nature même augmente l'ouverture du réseau et le rend plus vulnérable.

2. Par ailleurs, les SDN ne bénéficient pas encore d'un standard complètement mur, certes OpenFlow est un standard clé, mais son utilisation dans le monde télécoms n'est pas

encore normée, et paradoxalement des problèmes d'interopérabilité peuvent se poser. Enfin, une trop grande variété d'équipementier peut constituer un problème en ce qui concerne le rapport de force entre client et fournisseurs, mais aussi en termes de références à gérer par le SDN.[25]

3. Les SDN sont à la croisée des univers télécoms et informatique et permettent d'envisager une plus grande flexibilité et une meilleure efficacité du réseau en permettant un fonctionnement similaire à des infrastructures ou des plateformes de type Cloud. Le réseau peut évoluer plus rapidement et devient "programmable" et plus évolutif. C'est pourquoi des opérateurs majeurs tels que ATT, Telefonica, Deutsche Telecom, Vodafone, Orange ou Verizon Wireless, ainsi que des constructeurs tels qu'Alcatel-Lucent, Cisco, Huawei, IBM, Juniper Networks et Nokia ont ainsi mis en place une initiative afin de normer les SDN dans les télécoms et les intégrer au cœur de leurs futurs réseaux [26].

1.14 Conclusion :

Le Software Defined Networking annonce des changements importants sur les réseaux dans les années à venir, c'est une technologie émergente qui est susceptible de révolutionner les activités de réseau traditionnelles. Ceux-ci vont voir leur architecture profondément évoluer, facilitant des nouveaux usages, permettant aux administrateurs réseau de gérer et de contrôler automatiquement et dynamiquement un grand nombre de dispositifs, de services, de topologie, de trajets de trafic et de gestion de paquets (qualité de service) en utilisant un langage et des API de haut niveau.

La gestion inclut le provisionnement , le fonctionnement, la surveillance, l'optimisation et la gestion des FCAPS (faults, Configuration, accounting, performance, and security) dans un environnement multi-locataire. Aucun domaine ne semble épargné : WAN, Datacenters, campus, sécurité... L'enjeu pour les administrateurs réseau est d'accompagner cette nouvelle étape afin de pouvoir tirer profit de ces nouvelles capacités. Le chapitre suivant décrira en détail les notions théoriques

Chapitre 2

Attributes Based Encryption -ABE-

2.1 Introduction

Le cryptage est une méthode de codage des données qui protège la confidentialité de son contenu contre les attaquants non autorisés. Traditionnellement, le cryptage a été considéré comme un outil permettant une communication sécurisée entre un expéditeur (crypteur) et un destinataire d'informations ciblé. Dans la cryptographie traditionnelle, un message est crypté pour un récepteur spécifique en utilisant la clé publique du destinataire.

Un expéditeur (éditeur de données) a généralement besoin de connaître les identités des destinataires (abonnés aux données) et doit partager les informations d'identification avec eux. L'intention est qu'un expéditeur crypte des données qui ne peuvent être déchiffrées et lues que par un destinataire exact [27].

Le cryptage basé sur l'attribut (ABE) est une nouvelle vision du cryptage qui va au-delà de ces restrictions traditionnelles en permettant un contrôle d'accès basé sur une politique exible qui est crypto graphiquement (ou mathématiquement) appliqué, ce concept de sécurité a été proposé pour la 1er fois par Sahai et Waters (2005). Sur la base de leur travail, Goyal et al. (2006) ont proposé un système ABE de contrôle d'accès à grain fin, qui prend en charge toute formule d'accès monotone [28] .

Dans ce qui suit, nous présentons trois principaux modèles/ mécanismes de contrôle d'accès , la cryptographie ABE ,ses différents types et les architectures ABE les plus récents.

2.2 Contrôle d'accès :

2.2.1 introduction :

Le contrôle d'accès est un ensemble de contrôles visant à restreindre l'accès à certaines ressources. Si on y pense, les contrôles d'accès sont partout autour de nous. Une porte à votre chambre, les gardes vous permettant d'entrer dans l'immeuble en voyant votre carte d'accès, en balayant votre carte et en scannant vos doigts sur le système biométrique, une file d'attente à la cantine ou vos informations d'accès à FB, tous sont exemples de différents types de contrôle d'accès.

Dans cette partie , nous nous concentrons sur les trois principaux modèles/ mécanismes de contrôle d'accès.

2.2.2 Les modèles de contrôle d'accès :

Discretionary Access Control DAC :

Comme son nom l'indique, ce modèle de contrôle d'accès est basé sur la discrétion d'un utilisateur. C'est-à-dire que le propriétaire de la ressource peut donner des droits d'accès sur cette ressource à d'autres utilisateurs sur la base de sa discrétion.

Donc, les décisions sont prises directement pour les sujets. Pour ce faire, des listes de contrôle d'accès (ACL) sont utilisées . La liste de contrôle d'accès contrôle qui a accès à la ressource et le propriétaire des données définit les droits ou les autorisations. Les autorisations identifient les actions que le sujet peut effectuer sur l'objet. La plupart des systèmes d'exploitation, y compris les fenêtres, les serveurs d'Unix sont basées sur le mode DAC.

Le problème de sécurité principal de DAC est qu'il ne fournit pas une assurance réelle - les restrictions d'accès peuvent être facilement contournées et cela peut augmenter le risque que les données soient rendues accessibles aux utilisateurs qui ne devraient pas nécessairement avoir accès.

Mandatory Access Control ou MAC :

Dans ce modèle, les utilisateurs / propriétaires n'ont pas le privilège de décider qui peut accéder à leurs fichiers. Ici, le système d'exploitation est le décideur qui prend le pas sur les souhaits de l'utilisateur. Dans ce modèle, tous les sujets (utilisateurs) et objets (ressources) sont

classés et associés à une étiquette de sécurité. Les étiquettes de sécurité du sujet et de l'objet ainsi que la politique de sécurité déterminent si le sujet peut accéder à l'objet.

Les règles d'accès des sujets aux objets sont définies par le responsable de la sécurité, configurées par l'administrateur, appliquées par le système d'exploitation et prises en charge par les technologies de sécurité Par exemple, certaines données peuvent avoir un label «top secret» ou de niveau 1. D'autres informations peuvent avoir un niveau "secret" ou de niveau 2. D'autres informations peuvent avoir un niveau "gratuit" ou de niveau 3.

Ainsi, les données ne peuvent être accessibles que par des personnes ayant un certain niveau d'autorisation. Si nous n'avons pas une autorisation suffisante, nous ne pouvons pas accéder à ces données. Par exemple, si nous avons le niveau d'autorisation 2, nous pouvons accéder aux données étiquetées avec «secret» et «gratuit», mais nous ne pouvons pas accéder aux informations étiquetées avec «top-secret». Si nous avons le niveau d'autorisation 1, nous pouvons accéder à toutes les données. Il s'agit d'un modèle de contrôle d'accès plus strict et plutôt statique que le DAC.

Le contrôle d'accès obligatoire est de loin l'environnement de contrôle d'accès le plus sécurisé mais il nécessite une planification considérable avant de pouvoir être mise en œuvre efficacement. Une fois implémenté, il impose également une charge de gestion système élevée en raison de la nécessité de mettre constamment à jour les étiquettes d'objet et de compte pour accueillir de nouvelles données, de nouveaux utilisateurs et des changements dans la catégorisation et la classification des utilisateurs existants.[29]

Role Based Access Control RBAC :

Dans ce modèle, l'accès à une ressource est régi en fonction du rôle que le sujet détient au sein d'une organisation. C'est un hybride entre MAC et DAC. Le rôle peut être un poste, une appartenance à un groupe ou un niveau d'accès sécurisé.

Les utilisateurs sont membres de certains rôles et cela leur donne accès à certaines ressources de l'organisation. RBAC est également connu comme contrôle d'accès non discrétionnaire parce que l'utilisateur hérite des privilèges liés à son rôle. L'utilisateur n'a aucun contrôle sur le rôle qui lui sera assigné. Trois règles principales sont définies pour RBAC :

1. **Attribution de rôle :** un sujet peut exécuter une transaction uniquement si le sujet a sélectionné ou été attribué à un rôle approprié.
2. **Autorisation de rôle :** Le rôle actif d'un sujet doit être autorisé pour le sujet. Avec la

règle 1 ci-dessus, cette règle garantit que les utilisateurs ne peuvent prendre en charge que les rôles pour lesquels ils sont autorisés.

3. **Autorisation de transaction :** un sujet peut exécuter une transaction uniquement si la transaction est autorisée pour le rôle actif du sujet. Avec les règles 1 et 2, cette règle garantit que les utilisateurs ne peuvent exécuter que les transactions pour lesquelles ils sont autorisés.

   Des contraintes supplémentaires peuvent également être appliquées et les rôles peuvent être combinés dans une hiérarchie où les rôles de niveau supérieur englobent les autorisations appartenant à des sous-rôles de niveau inférieur.

Ce dernier mécanisme / modèle ressemble un peu à notre sujet principal ou modèle de cryptage ABE ils ont presque le même concept. Nous verrons comment cela est dans la deuxième partie de ce chapitre.

2.3 Attribute-based Encryption ABE :

2.3.1 Concept :

Le cryptage basé sur l'attribut (ABE) est une technique de cryptage qui généralise le rôle fonctionnel des identités et des clés. En un schéma de chiffrement asymétrique traditionnel, les identités se rapportent à une clé publique distincte /tuples clés privées. Dans ABE, les concepts de clé publique et de clé privée sont remplacés par des ensembles d'attributs, qui résument les propriétés réelles de l'utilisateur.

Cet ensemble d'attributs est utilisé comme identité pour générer une clé secrète, ainsi que la structure d'accès qui effectue le contrôle d'accès. Les informations d'identification cryptographiques sont émises par une partie de confiance centrale appelée autorité d'attribut (attribute authority), qui est en possession d'une clé principale globale pour la génération de clé.

Etant donné que les utilisateurs sont associés à un ensemble d'attributs, ils peuvent essayer de modifier et d'échanger certaines fonctionnalités/attributs et composants de clés privée pour obtenir plus d'autorité et la capacité de décrypter plus d'informations et de champs. Cependant, les systèmes ABE sont résistants à la collusion [30] , ce qui signifie que les clés des différents utilisateurs sont incompatibles en raison de la construction cryptographique. Cette méthode de chiffrement combine avec succès le cryptage et le contrôle d'accès en un seul système et est

idéal pour partager des secrets entre les groupes. L'utilisation de systèmes ABE peut avoir les avantages suivants : (1) réduire les frais généraux de communication sur Internet, et (2) fournir un accès précis contrôle [31].

2.3.2 Politiques d'accès :

Il existe deux type de chiffrement / politiques d'accès basé sur les attributs, le cryptage basé sur l'attribut clé-politique KP-ABE (key-policy) et le cryptage basé sur l'attribut texte chiffré-politique CP-ABE (Ciphertext-policy).

Politique d'accès sur la clé (Key-policy attribute based encryption (KP-ABE)) :

Dans un système de chiffrement basé sur les attributs de la politique clé (KP-ABE), les textes chiffrés sont étiquetés par l'expéditeur avec un ensemble d'attributs descriptifs, tandis que la clé privée de l'utilisateur est émise par l'autorité d'attribut de confiance (également appelée structure d'accès) qui spécifie quel type de chiffrements la clé peut décrypter.
Cela implique que la politique d'accès est attachée à la clé privée de l'utilisateur et utilise l'ensemble d'attributs de l'utilisateur pour décrire les données cryptées. Si un ensemble d'attributs de la clé de confidentialité satisfait la politique d'accès, l'utilisateur décrypte les données cryptées. Sinon, l'utilisateur ne peut pas obtenir les données cryptées [32].
Un inconvénient de KP-ABE est que la politique d'accès est intégrée dans la clé privée des utilisateurs, de sorte que le propriétaire des données qui crypte les données à un contrôle mineur sur qui peut déchiffrer. De plus, le propriétaire des données doit faire confiance à l'émetteur principal et employer un serveur de confiance pour stocker tous les expressifs en clair.Le système KP-ABE comprend les quatre algorithmes suivants [33] :

- **Setup/Configuration :** Cet algorithme prend en entrée un paramètre de sécurité et renvoie la clé publique PK et une clé secrète principale du système MK. PK est utilisé par les expéditeurs de message pour le cryptage. MK est utilisé pour générer des clés secrètes utilisateur et n'est connu que par l'Autorité.
- **Encryption/Chiffrer (M, PK, Y) :** Cet algorithme prend un message M, la clé publique PK et un ensemble d'attributs Y en entrée. La sortie est le texte chiffré/cryptogramme
- **Key Generation/Génération clé (MK, T) :** Cet algorithme prend en entrée une structure d'accès (politique d'accès) T et la clé secrète principale MK. Il sort une clé secrète SK qui permet à l'utilisateur de déchiffrer un message chiffré sous un ensemble

d'attributs si et seulement si égal à T.

— **Decryption/Déchiffrement (SK, E) :** Il prend en entrée la clé secrète SK de l'utilisateur pour la structure d'accès (politique d'accès) T et le texte chiffré E, qui a été chiffré sous l'ensemble d'attributs. Cet algorithme génère le message M si et seulement si l'ensemble d'attributs satisfait la structure d'accès T de l'utilisateur.

Exemple [34]/ L'idée principale de l'ABE de Key-Policy (KP-ABE) est présentée dans la Figure 6, qui est une version simplifiée de (Wang, Zhang, Schooler Ion 2014). Alice souhaite produire un document chiffré. Pour ce faire, elle demande la clé publique du MKG et la crypte, en utilisant un ensemble d'attributs qui décrit le document. Elle peut ensuite le publier avec l'ensemble d'attributs. Bob, qui est l'un des consommateurs du document d'Alice, récupère le document crypté et demande au MKG la clé publique et sa clé privée. La clé privée de Bob inclut une politique d'accès qui régule ce que Bob est autorisé à voir, basée sur un ensemble d'attributs dans la politique. Si la politique dans la clé privée de Bob est satisfaite par les attributs fournis dans le message d'Alice, il peut déchiffrer le document. En cryptant un document, Alice, si elle le souhaite, peut publier le document sur un zone de stockage de tiers, rendant ainsi le document plus disponible pour le Destinataires. Une fois que Bob a récupéré sa clé privée, il peut déchiffrer tout message chiffré avec la clé publique tant que sa politique est satisfaite. Ainsi, aucune communication avec le MKG n'est nécessaire après la récupération initiale de la clé.

Politique d'accès sur le message chiffré (Ciphertext-policy attribute based encryption (CP-ABE)) :

Contrairement à KP-ABE, dans le système CP-ABE, la clé privée d'un utilisateur est associée à un ensemble d'attributs. Plus précisément CP-ABE fournit une méthode évolutive de cryptage des données où l'utilisateur crypteur définit l'ensemble d'attributs, puis l'utilisateur décrypteur doit conserver l'ensemble d'attributs pour déchiffrer le cryptogramme [30]. Par conséquent, différents utilisateurs sont autorisés à déchiffrer différents blocs de données dans les différents politiques d'accès. Ainsi, le mécanisme CP-ABE est conceptuellement plus proche de la méthode traditionnelle de contrôle d'accès basé sur le rôle. Le système CP-ABE comprend les quatre algorithmes suivants [33] :

— **Setup/Configuration :** Cet algorithme prend en entrée un paramètre de sécurité et renvoie la clé publique PK ainsi qu'une clé secrète de système MK.PK est utilisé par les

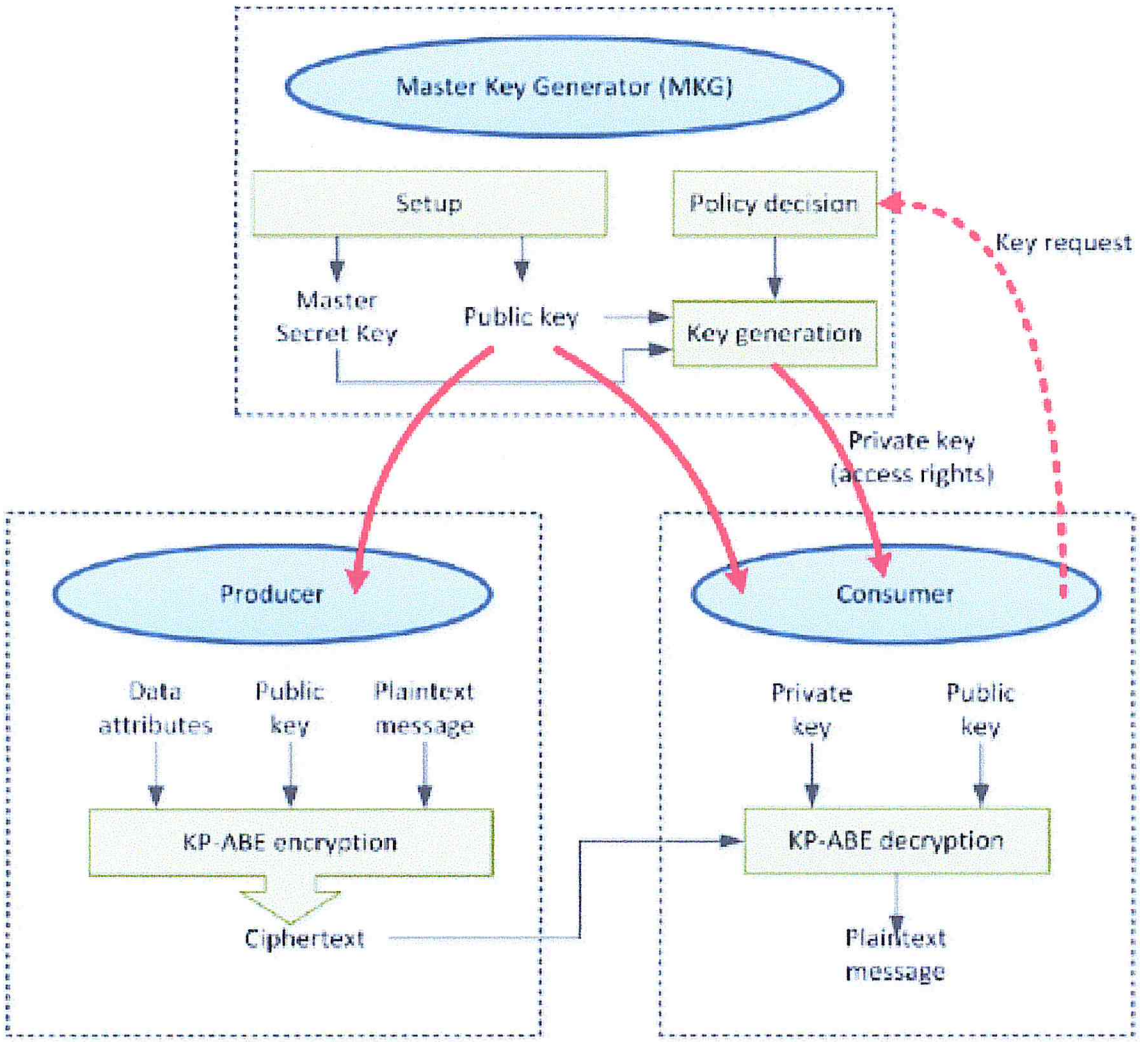


FIGURE 2.1 – Cryptage basé sur les attributs de la politique-clé.[34]

expéditeurs de message pour le cryptage. MK est utilisé pour générer des clés secrètes utilisateur et n'est connu que par l'autorité.

— **Encryption/Chiffrer (M, PK, Y) :** Cet algorithme prend en entrée le paramètre public PK, un message M, et une structure d'accès T. La sortie est le texte chiffré/cryptogramme CT.

— **Key Generation/Génération clé (MK, T) :** Cet algorithme prend en entrée un ensemble d'attributs Y associés à l'utilisateur et la clé secrète principale MK. Il sort une clé secrète SK qui permet à l'utilisateur de déchiffrer un message chiffré sous une arborescence d'accès T si et seulement si correspond à T.

— **Decryption/Déchiffrement (SK, E) :** Cet algorithme prend en entrée le texte chiffré CT et une clé secrète SK pour un ensemble d'attributs. Il renvoie le message M si et seulement si satisfait la structure d'accès associée au texte chiffré CT.

Exemple [34] : L'idée principale de CP-ABE est présentée dans la figure 7, qui est une version simplifiée de (Wang et al. 2014). Alice veut chiffrer un document. Elle extrait la clé publique du fichier MKG et l'utilise avec une structure d'accès qu'elle a créée en fonction d'une politique de contrôle d'accès pour chiffrer le document. Elle peut ensuite afficher le document dans une zone de stockage de son choix. Bob récupère sa clé privée, contenant ses attributs descriptifs, ainsi que la clé publique du MKG. Si les attributs de la clé privée correspondent à l'attribut dans l'accès structure d'une manière correcte, Bob peut procéder à déchiffrer le document.

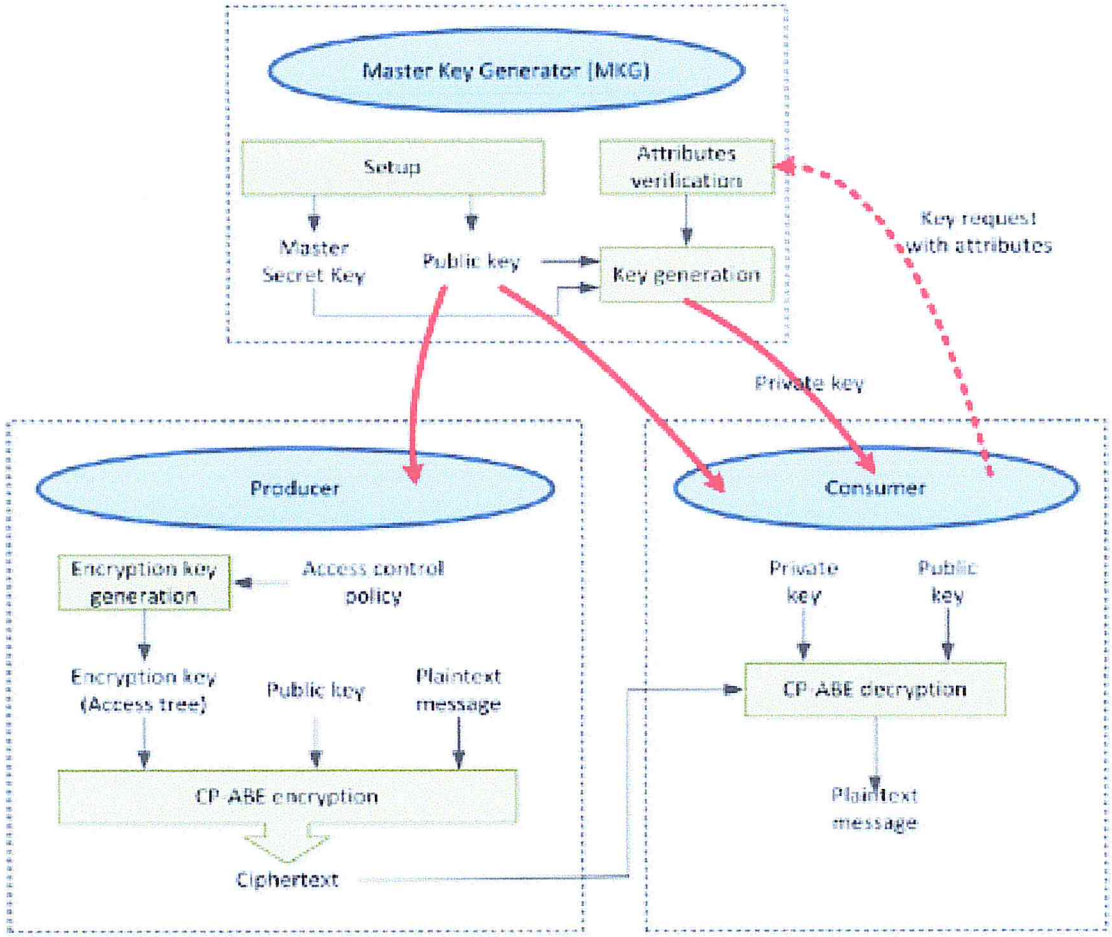


FIGURE 2.2 – Cryptage basé sur les attributs Ciphertext-policy.[34]

2.3.3 Les critères d'un chiffrement basé sur un attribut idéal (Exigence) :

En tant que système de chiffrement, ABE doit répondre à l'exigence de confidentialité des données. Puisque ABE a également intégré le contrôle d'accès, l'authentification de l'utilisateur et la révocation doivent être satisfaites. Nous présentons dans qui ce suit un résumé des critères, que les schémas de cryptage basés sur les attributs idéaux, sont listés comme suit [31] :

— **La confidentialité des données :** Le propriétaire des données chiffre les données avant de télécharger les données sur le serveur de stockage. Par conséquent, l'utilisateur non autorisé et le serveur de stockage ne peuvent pas connaître les données de chiffrement.

— **Contrôle d'accès à grain fin :** Chaque utilisateur a son propre droit d'accès qui peut être différent pour chaque utilisateur. Même si les utilisateurs existent dans le même groupe, leur droit d'accès peut ne pas être le même.

— **Évolutivité :** Lorsque les utilisateurs autorisés augmentent, le système peut fonctionner efficacement. Le nombre d'utilisateurs autorisés ne peut donc pas affecter les

performances du système.

— **Révocation de l'utilisateur :** Si l'utilisateur quitte le système, le système peut révoquer son droit d'accès directement du système. L'utilisateur révocable ne peut accéder à aucune donnée stockée, car son droit d'accès a été révoqué.

— **Responsabilité des utilisateurs :** Si l'utilisateur autorisé est malhonnête, il partagera sa clé privée attribut avec l'autre utilisateur non autorisé. Il provoque le problème que la clé illégale partagerait parmi les utilisateurs non autorisés.

— **Collusion résistant :** Les utilisateurs ne peuvent pas combiner leurs attributs pour déchiffrer les données chiffrées. Puisque chaque attribut est lié au polynôme ou au nombre aléatoire, les différents utilisateurs ne peuvent pas s'entendre.

2.3.4 Architecture des Systèmes ABE :

Architecture Centralisée :

Dans cette architecture, le système de chiffrement/contrôle d'accès ABE les clés privées sont délivré par une seule autorité c'est à dire il existe une seule autorité centrale pour la gestion des clés, cette autorité devrait être en mesure de vérifier tous les attributs les informations d'identification délivrées pour chaque utilisateur du système. Ces systèmes/architectures peuvent être utilisés pour partager des informations dans une entité ou d'une organisation Dans cette architecture le système ABE se compose de quatre éléments principaux [35] :

1. **Le propriétaire des données(DO) :** qui est responsable de la détermination des règles d'accès.
2. **Utilisateurs (user) :** qui a un ensemble d'attributs .Il est autorisé à accéder aux données stockées si la politique d'accès associé au texte chiffré qui a été défini par le propriétaire de donnée est satisfaite par ses attributs.
3. **Fournisseur (fournisseur de service Cloud) :** cette entité fournit des services d'accès et d'externalisation de données aux propriétaires et aux utilisateurs.
4. **Autorité centrale :** il s'agit d'une partie qui est chargée d'autoriser, de révoquer et mettre à jour les clés des utilisateurs.

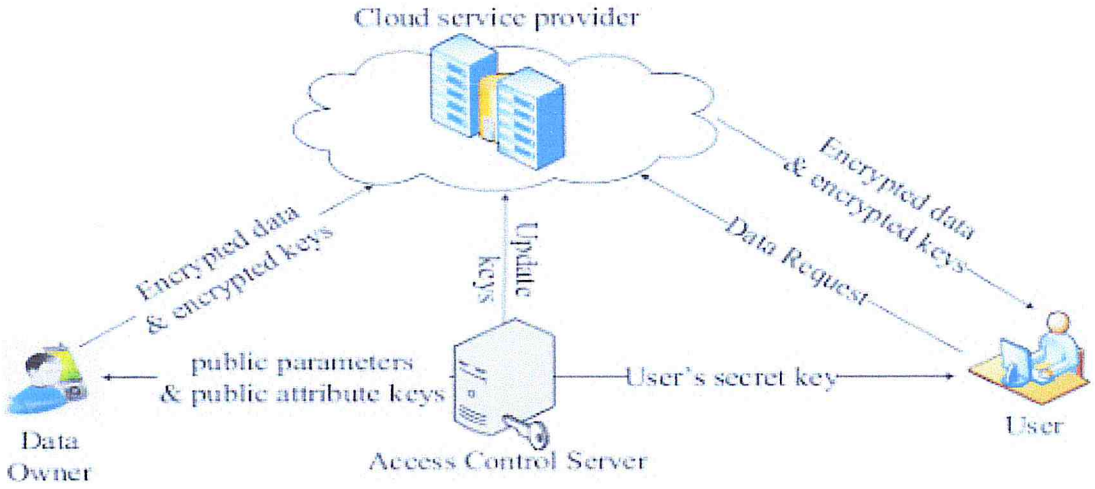


FIGURE 2.3 – Architecture centralisé -Single_authority attribute-based encryption-.[35]

Architecture Décentralisée :

Un système ABE d'autorité unique ou le système ABE centralisé présente les inconvénients suivants :

— Tous les attributs du système sont gérés par l'autorité unique. Échec ou corruption de l'autorité affecte l'ensemble du système.

— Les clés privées sont distribuées par l'autorité unique afin que l'autorité unique puisse décrypter tout texte chiffré dans le système.

Pour vaincre les inconvénients d'un système basé sur des attributs d'autorité unique. La notion de schéma de chiffrement basé sur un attribut d'autorité multiple a d'abord été proposée par Chase [36], Les schémas de chiffrement basés sur des attributs multi-autorités sont des schémas de chiffrement basés sur des attributs de politique clé multi-autorité (MA-KPABE) ou des schémas de chiffrement basés sur des attributs de politique chiffrée multi-autorité (MACPABE). Le système à N autorités de confiance et utilise un du ces autorité comme une autorité centrale de confiance (CA) qui est responsable de la génération des clés de configuration pour chaque utilisateur sur la base du GID (global identifiers) de l'utilisateur qu'est un numéro de série et qui aide les autorités à distingues les utilisateurs au-delà de leurs attributs. LA (CA) détient aussi une clé maitresse pour déchiffrer les données. Chacune des autorités utilise un pseudo-aléatoire (PRF) pour générer les clés secrètes aléatoires pour chaque utilisateur sur la base du GID correspondante pour empêcher la collusion. Si l'utilisateur dispose d'une clé qui satisfaire chacun des autorités. La (CA) autorise l'utilisateur à déchiffrer la donnée.

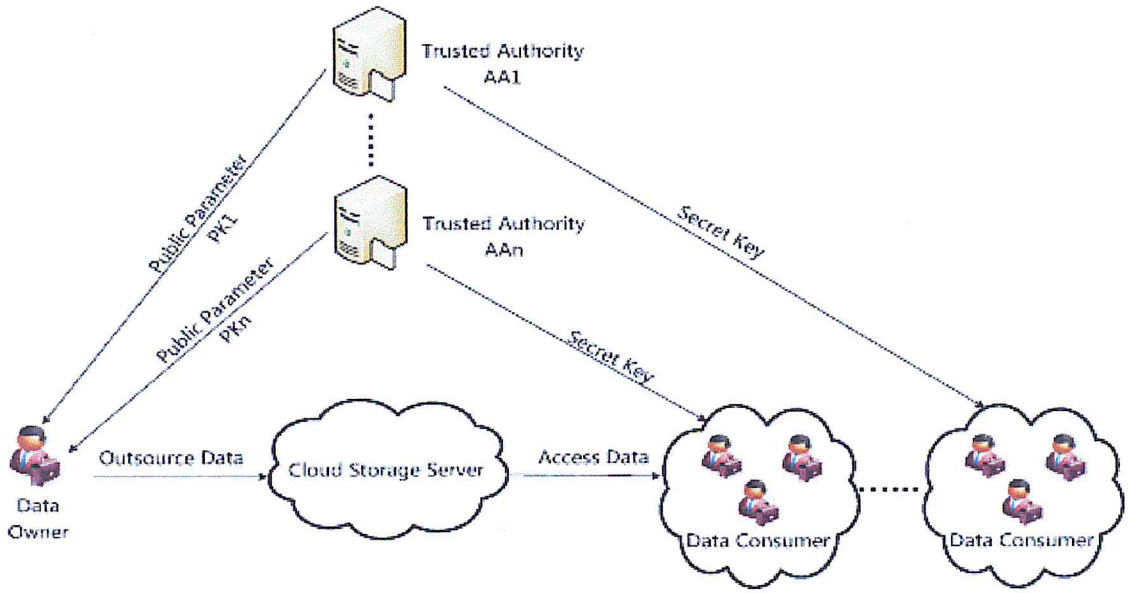


FIGURE 2.4 – : Architecture décentralisé -Multi_authority attribute-based encryption-.[35]

2.4 Conclusion :

Le chiffrement basé sur les attributs est une technique largement utilisée pour contrôle d'accès. Il a été utilisé pour affiner les utilisateurs d'accéder à l'information. ABE a l'avantage significatif en réalisant un cryptage un-à-plusieurs flexible au lieu de un à un ; il est envisagé comme un outil prometteur pour résoudre le problème du partage sécurisé et précis des données et du contrôle d'accès décentralisé. ABE a aussi l'avantage de la force de la clé, permettant aux utilisateurs d'avoir un cryptage plus fort, que d'autres cryptages. La cryptanalyse a révélé que la complexité de l'algorithme est de bon ordre et ne peut être rendue vulnérable.

Ces deux chapitres (1 et 2) transcrit le travail de documentation et de recherche qui a été mené dans l'objectif de vulgariser le plus possible la technologie SDN, qui reste sujette à plusieurs interprétations, Nous avons notamment cité quelques définitions de SDN et explicité le rôle du contrôleur au sein d'une architecture SDN, ainsi que le fonctionnement du Protocol OpenFlow. Nous avons introduit ensuit le contrôle d'accès ABE proposée pour soutenir le contrôle d'accès et garantir la confidentialité des données. Nous avons notamment cité les deux type de politique d'accès basé sur les attributs. Ces éléments nous sont utiles lors de la mise en place de la solution proposée. Dans le chapitre qui suit, nous allons entamer la partie étude et présentation du matériel .

Chapitre 3

ONOS et OVS

3.1 Introduction :

Dans ce chapitre, nous définirons les pré-requis de la solution qui vise à mettre en place un contrôle d'accès basé sur les attributs ABE dans une architecture réseau basé SDN.
Nous commençons par définir les outils important , on précise le choix de contrôleur et le commutateur. Nous finirons par donner une architecture globale du système, et nous terminons ce chapitre par une conclusion.

3.2 SPECIFICATIONS ET BESIONS DE LA SOLUTION :

Afin de mettre en place la solution proposés, plusieurs composants et APIs nous ont été nécessaires, la suit de cette section a pour but de définir les outils qui constituent la solution.

3.2.1 Contrôleur :

Toute architecture SDN nécessite un ou plusieurs contrôleurs, celui-ci constitue le cerveau du réseau. Il permet de configurer le réseau en agissant sur les commutateurs, de faire du monitoring et récupérer les remontées d'informations du plan de données. Il y a, cependant, une variété de contrôleurs Open Source sur le marché, ce qui rend difficile le choix de celui qui convient à nos besoins et possède de meilleures caractéristiques / fonctionnalités, efficacité et réseaux d'application.

TABLE 3.1 – Comparaison basée sur les fonctionnalités des contrôleurs SDN open source les plus populaires.

Contrôleur	Implémentation	GUI	Distribué/Centralisé	API South-bound	API North-bound	Plateforme	Développeurs	Description
NOX	C++/ Python	Python +QT4	Centralisé	OF 1.0	REST API	Plus Pris en charge sur Linux	Nicira	Le premier contrôleur OpenFlow [37].
POX	Python	Python +QT4	Centralisé	OF 1.0	REST API	Linux, MAC OS, et Windows	Nicira	Contrôleur prenant en charge OpenFlow disposant d'une API de haut niveau comprenant un graphe de topologie et un support de virtualisation [38].
Open-Day-light	JAVA	Basé Web	Distribué	OF1.0, 1.3,1.4, NET CONF/ YANG, OVSDB, PCEP, BGP/LS, LISP, SNMP	REST API	Linux, MAC OS, et Windows	Linux Foundation With Memberships Covering Over 40 Companies, Such As Cisco, IBM, NEC	Plate-forme pour la construction d'applications réseau programmables et définies par logiciel [39].
ONOS	JAVA	Basé Web	Distribué	OF1.0, 1.3, NETCONF	REST API	Linux, MAC OS, et Windows	ON.LAB, At&T, Ciena, Cisco, Ericsson, Fujitsu, Huawei, Intel, Nec, Nsf.Ntt Comunnication, Sk Telecom	Un cluster d'avion de contrôle évolutif Open Source offrant interface graphique et OpenFlow ainsi que le support NETCONF [40].
FloodLight	JAVA	Basé Web/ Java	Centralisé	OF 1.0, 1.3	REST API	Linux, MAC OS, et Windows	Big Switch Networks	Contrôleur OpenFlow, forké depuis le contrôleur Beacon [41].

Un résumé de comparaison entre des contrôleurs SDN open source les plus populaires basée sur leurs fonctionnalités dans le Tableau 1, avec leur plate-forme de développement. Les deux catégories de contrôleurs couvertes dans le tableau sont les contrôleurs généraux et spéciaux.

Open Network Operating System (ONOS) :

ONOS est un contrôleur SDN open-source récemment lancé, il s'agit d'un système d'exploitation réseau (NOS) conçu pour aider les fournisseurs de services réseau à créer des réseaux définis par logiciel, conçus pour une évolutivité, une disponibilité et une performance élevées. Même si elle est spécifiquement conçue pour répondre aux besoins des fournisseurs de services, ONOS peut également agir en tant que plan de contrôle SDN (réseau défini par logiciel) pour les réseaux locaux d'entreprise (LAN) et les réseaux de centres de données. La plateforme ONOS comprend [42]

- Une plate-forme et un ensemble d'applications qui agissent comme un contrôleur SDN réparti, modulaire et extensible.
- Gestion, configuration et déploiement simplifiés de nouveaux logiciels, matériels et services.
- Une architecture scale-out pour fournir la résilience et l'évolutivité nécessaires pour répondre aux rigueurs des environnements de support de production.

ARCHITECTURE ONOS :

L'architecture ONOS est conçue spécifiquement pour les besoins de performances, de haute disponibilité et d'échelle des réseaux de classe opérateur, avec des abstractions bien définies. L'architecture ONOS peut être vue comme une collection à quatre niveaux de plusieurs sous-systèmes (également appelés services), où chaque sous-système réalise un service et est implémenté comme une combinaison de composants présents dans quatre couches différentes : application, abstractions /API NorthBound, noyau(cœur) et protocoles/ abstractions SouthBound.

Dans certaines littératures et documents de recherche, nous pourrions trouver l'architecture onos définie comme une collection à trois niveaux de multiples 'sous-systèmes', ce qui signifie qu'ils présentent l'architecture onos en trois couches : couche application, couche noyau/cœur et couche protocoles SouthBound.

Dans la figure 1 ci-dessous, nous présentons une architecture de haut niveau de l'ONOS [43]

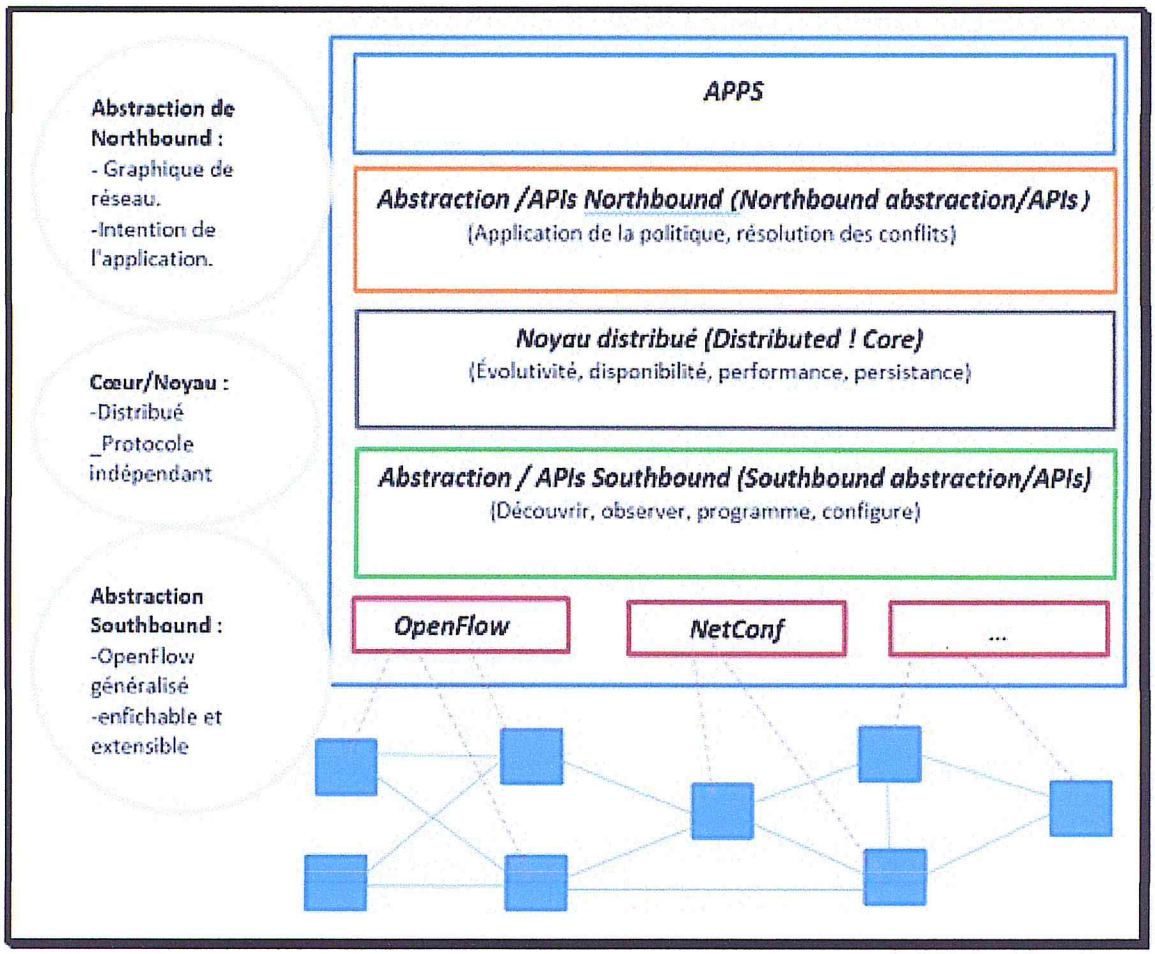


FIGURE 3.1 – Architecture ONOS [43].

cette figure 3.1 montre les différentes couches ou les niveaux de l'architecture onos qui sont considérés comme un sous-systèmes, ils sont : application, abstractions /API NorthBound, noyau(cœur) et protocoles/ abstractions SouthBound, où abstractions SouthBound communique avec l'infrastructure les commutateurs et routeur il découvre, observe, programme, configure les flux du réseau. Noyau(cœur),il est responsable de servir et de garder évolutivité, disponibilité, performance et persistance de réseau .abstractions NorthBound communique avec lacouche application qui est utilisée par l utilisateur.

Pourquoi Onos :

Open Network Operationg System (ONOS) a été créé pour produire le système d'exploitation de réseau Open Source qui permettra aux fournisseurs de services de créer de véritables réseaux définis par logiciel ,il a un certain nombre de buts et de fonctions que nous pouvons mentionner, il libère les développeurs d'applications réseau de connaître les subtilités

du matériel propriétaire, permettre aux opérateurs de se libérer du la complexité des interfaces et des protocoles propriétaires, et aussi de permettre à l'innovation de se produire à la fois pour le matériel et le logiciel du réseau, indépendamment, à leur propre échelle de temps.
ONOS est conçu pour fournir des caractéristiques de classe opérateur telles que l'évolutivité, la haute disponibilité, les performances en termes de débit (intentions d'application par seconde) et de latence (temps de traitement des événements réseau). Abstraction / APIs NorthBound pour faciliter la création de nouveaux services en utilisant ONOS afin d'apporter l'agilité de style web aux réseaux.L'abstraction SouthBound avec des plug-ins de périphérique / protocole afin que ONOS puisse fournir un contrôle SDN pour les boîtes blanches activées par OpenFlow ainsi que pour les périphériques hérités. Cela permet une migration facile vers SDN basée sur des boîtes blanches.
ONOS comprend de nombreux avantages fondamentaux :

- Architectures modulaires et abstraites, architectures hautement cohérentes
- Test facile, maintenance et gestion du réseau.
- Contrôleurs évolutifs, fiables et faciles à gérer.
- Noyau distribué, abstraction NorthBound et SouthBound, modularité logicielle.
- Ajout facile et maintenance des serveurs.
- CORD offre de nombreuses activités importantes, telles que :
  - Access as a Service (AaaS),
  - as a Service (SaaS),
  - Internet as a Service (IaaS),
  - Content Distributed Networking (CDN),
  - Monitoring as a Service (MaaS).

3.2.2 Commutateur :

L'interconnexion des machines virtuelles et le transport des flux codés nécessitent l'utilisation de commutateurs. Etant dans des environnements SDN, ces commutateurs doivent présenter les SBIs adéquates.
Il existe, sur le marché, plusieurs solutions adéquates à ce cas de figure, parmi elles, Big Virtual switch, Pica8 et Open vSwitch. Dans un contexte de virtualisation et dans l'optique de mettre en place des topologies réseau maniables et compatibles avec OpenFlow, notre choix s'est porté sur Open vSwitch.

Un résumé de comparaison entre des commutateurs les plus populaires dans le Tableau 2, avec leur langage d'implémentation, catégories et description.

TABLE 3.2 – Commutateurs compatibles OpenFlow communs et piles autonomes.

Switch	Implémentation	Catégories	Description
Open vSwitch	C/Python	Software	OpenFlow utilisée à la fois comme commutateur virtuel et portée sur plusieurs plates-formes matérielles [44].
OpenFlowJ	Java	Software	OpenFlow stack écrit en Java [45].
ofsoftswitch13	C/C++	Software	Le logiciel d'espace utilisateur change la mise en œuvre [46].
OpenFlow Reference	C	Software	Minuscule pile de référence OpenFlow qui suit la spécification [47].
Pica8	C	Physical and software	Plate-forme logicielle pour les puces de commutation matérielles incluant la pile L2 / L3 [48].
Big Switch Networks - Big Virtual Switch	Propriétaire	Physical and software	Application de virtualisation de réseau de centre de données basée sur un commutateur OpenFlow [49].
NEC Programmable Flow Switch Series	Propriétaire	Physical and software	Series offre la virtualisation de réseau, le routage multitrajets, la sécurité et la programmabilité [50].
IBM RackSwitch G8264	Propriétaire	Physical	Offre une connectivité flexible à faible coût pour les serveurs et périphériques de stockage haut vitesse dans les environnements DC [51].
HP 2920, 3500, 3800, 5400 series	Propriétaire	Physical	Série de commutateurs modulaires avancés construits sur ASIC programmables offrant une QoS et une sécurité évolutive [52].

Open vSwitch :

Open vSwitch (OVS) est un commutateur virtuel multicouche. Il permet de virtualiser des commutateurs ayant le même comportement que les commutateurs physiques. OVS supporte les protocoles et APIs de configuration réseau classique (ex : CLI, 802.1q pour les vlan, SNMP, NAT, SPAN/RSPAN, IPSEC pour la sécurité etc.) ainsi que des APIs SouthBound (SBI) tel que OpenFlow et OVSDB, ce qui en fait un outil adéquat pour notre solution. La Figure 2 situe OVS au sein d'une architecture SDN et montre comment se fait l'interconnexion à son niveau.
OVS est utilisé pour mettre en place l'infrastructure réseau qui serra chapoté par le contrôleur et qui interconnectera les nœuds finaux. Les commutateurs OVS contiendront les tables de flux et dialogueront avec le contrôleur grâces à OpenFlow (pour le maintien des tables) et OVSDB pour la configuration [53].
OVS peut servir à interconnecter des machines virtuelles, qu'elles soient au sein d'un même hyperviseur ou se trouvant sur des hyperviseurs différents, il peut servir aussi à interconnecter des machines virtuelles avec d'autres machines ou équipements réseau physiques se trouvant sur des réseaux différents.
De par sa documentation, sa gratuité et les performances qu'il présente, OVS est de nos jours largement utilisé dans le monde de la recherche et de l'entreprise pour la mise en place de la solutions SDN open-source évolutive.

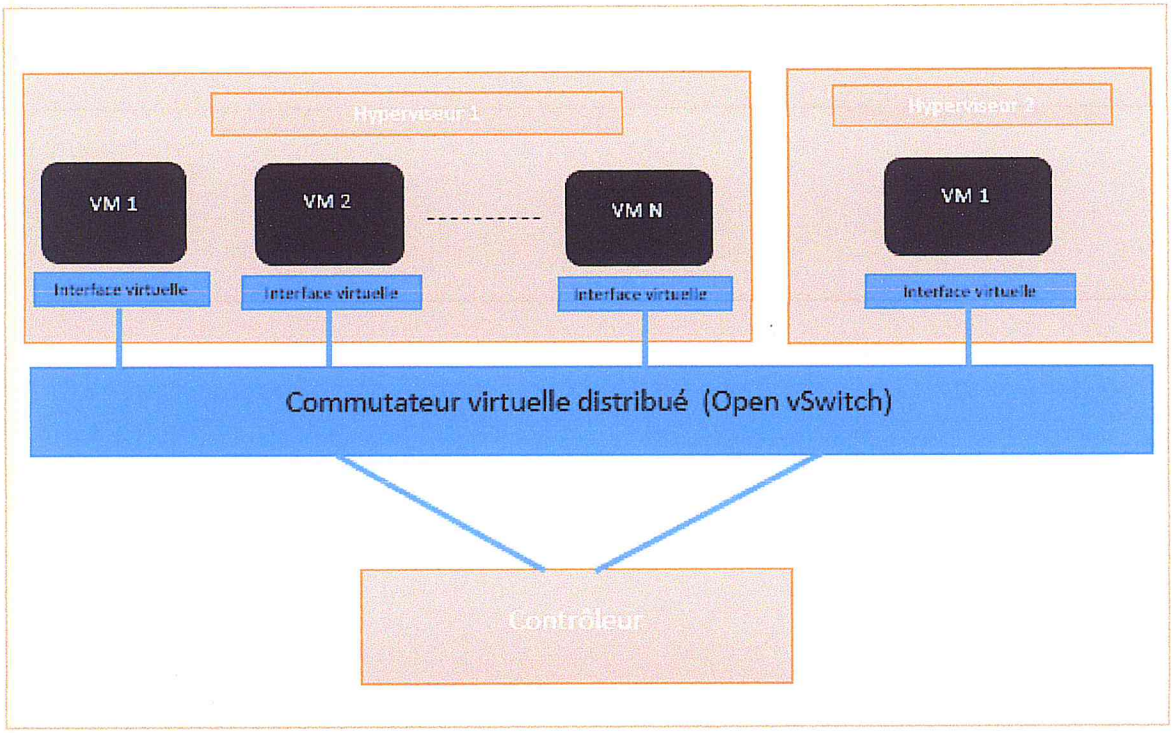


FIGURE 3.2 – Positionnement d'OVS au sein d'un architecture SDN.

Création de système avec ovs :

Open vSwitch peut avoir plus qu'un pont (bridge), le pont contienne des entrées qui décrivent les commutateurs dans l'instance Open vSwitch, chaque pont peut être connecté à un contrôleur SDN, et aussi chaque pont contient un certain nombre de ports, chaque port représente un port virtuel sur le commutateur virtuel, dont les données peuvent circuler à travers. Ils sont également utilisés comme ports en ce qui concerne les règles de OpenFlow. Chaque port peut être liée à une interface qui est une interface du réseau réel, physique.

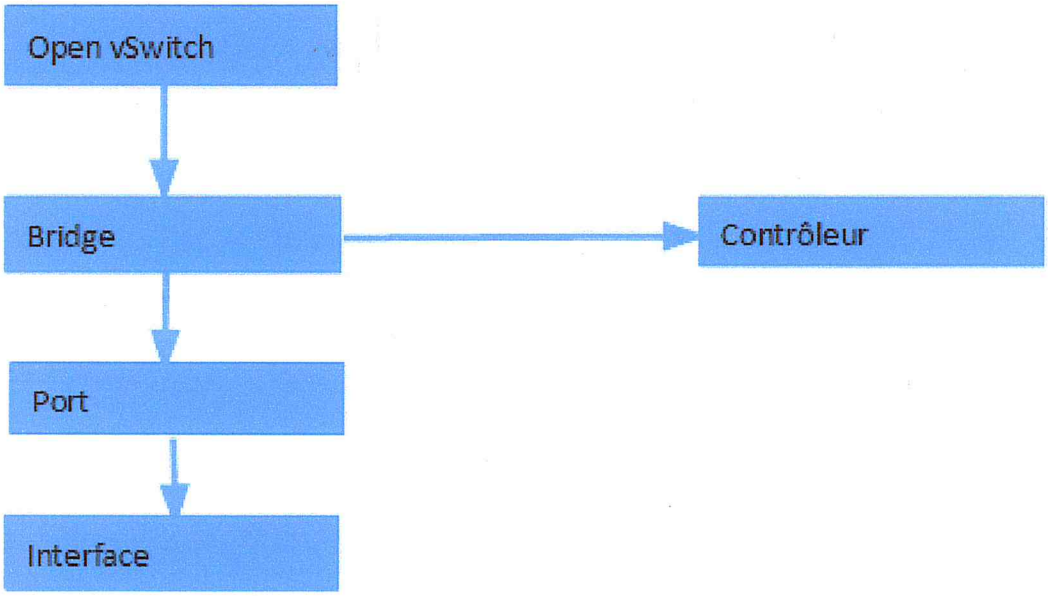


FIGURE 3.3 – création d'un système avec OVS.

3.3 Conclusion :

Dans ce chapitre nous avons cité et défini les utiles qui seront utilisés dans la réalisation de la solution proposée et nous avons précisé les raisons pour lesquelles nous avons choisi le contrôleur onos et le switch OVS, nous avons conclus ce chapitre par l'emplacement d'OVS dans un réseau SDN. Nous détaillerons plus dans ce qui suit le rôle et l'utilité de OpenVswitch et onos dans la conception et la réalisation d'un système de contrôle d'accès basé sur le chiffrement par attributs dans une architecture réseau basé SDN.

Chapitre 4

Conception ET Implémentation

4.1 Introduction :

Ce chapitre est structuré en deux sections. La première section décrit et détaille la conception de la solution qui vise à mettre en place un système de contrôle d'accès basé sur les attributs (ABE) dans une architecture réseau basé SDN. La deuxième section est rédigé dans l'optique d'offrir un manuel de mise en place d'une architecture SDN supportant le système de contrôle d'accès basé sur le chiffrement par attributs. Le travail documenté ci-dessous vise à expliquer la manière dont nous avons intégré concrètement la stratégie de contrôle d'accès ABE au sein du réseau et à présenter l'architecture conçue afin d'évaluer la stratégie de système de contrôle d'accès ABE, nous terminons ce chapitre par une conclusion.

4.2 SECTION 1 Conception :

4.2.1 Architecture du système :

Les utilisateurs du système (médecins, infirmiers, directeur de l'établissement? ...) demandent l'accès ils/elles peuvent avoir une clé d'accès s'ils/elles ont les autorisations nécessaires.

Ils auront des clés personnelles/IDs associées à leurs attributs, le contrôleur vérifie les informations envoyées par ces utilisateurs, si leurs clés + attributs correspondent aux politiques d'accès définir, ils/elles auront le privilège d'avoir une clé cp-abe pour l'utilisées plus tard.

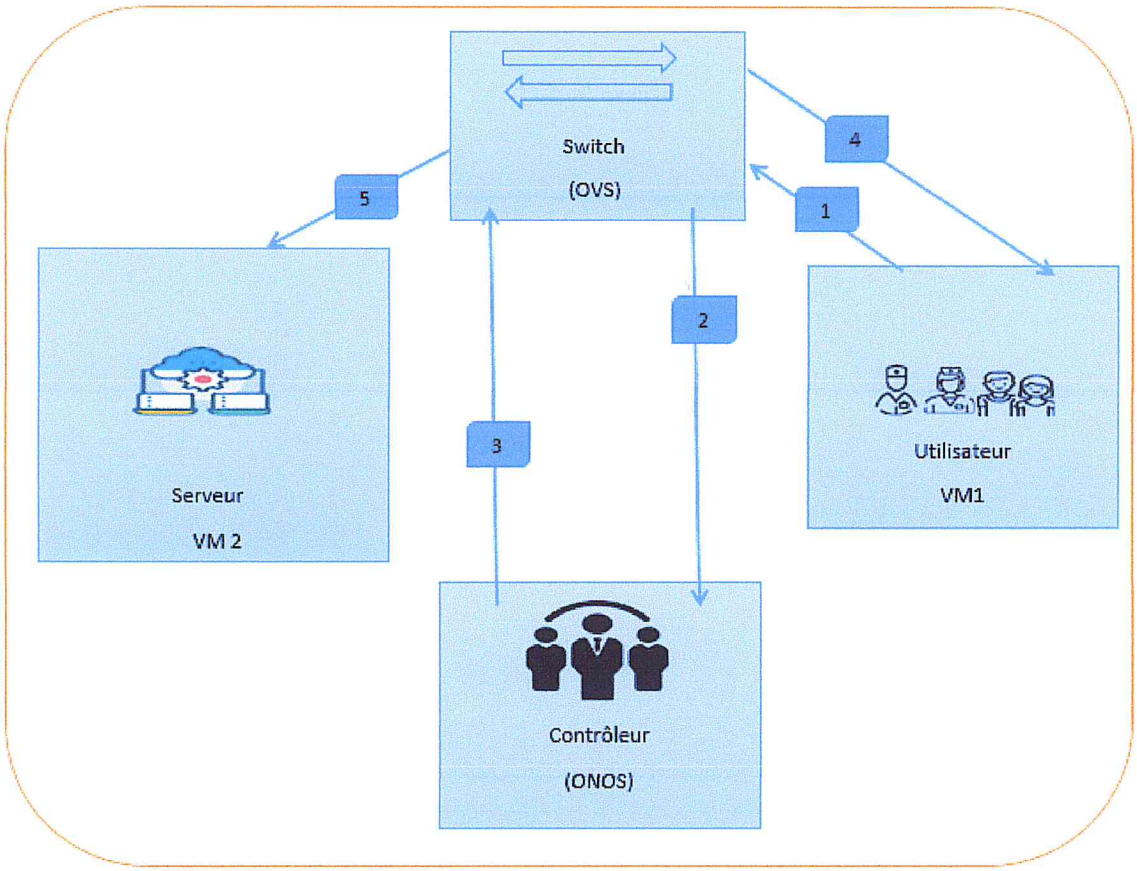


FIGURE 4.1 – Architecture de système.

* **L'utilisateur :**
  1 L'utilisateur envoie un fichier texte avec l'ensemble de ses attributs par exemple (nom, prénom, matière, années de travail) au serveur pour recevoir une clé lui permettant d'accéder / décrypter les données.
  4 L'utilisateur reçoit une réponse du contrôleur via le switch, ce pourrait être une clé pour déchiffrer les données plus tard ou un message d'erreur.
* **Serveur :**
  5 le serveur dans cette architecture n'a pas de travail à faire son rôle ici est d'ouvrir sa socket afin que l'utilisateur peut envoyer ses données.
* **Le contrôleur :**
  2 Le contrôleur intercepte le paquet pour analyser les données et extraire les attributs et exécute la politique d'accès choisi pour générer une clé si l'attribut satisfait la politique de contrôle d'accès, sinon une erreur de massage.
  3 Le contrôleur envoie sa réponse à l'utilisateur via le switch OVS, une clé si la politique d'accès est satisfaite, sinon un message d'erreur.
* **Switch :**
  - le rôle de switch OVS est de interconnecté l'utilisateur avec le serveur et le contrôleur qui va intercepter la demande de l'utilisateur et exécuter son contrôle d'accès.

4.2.2 Attribute Based Encryption chiffrement basé sur l'attributs :

L'ABE est une méthode de chiffrement qui combine le cryptage et le contrôle d'accès en un seul système , dans ce travail nous sommes intéressées seulement par la partie de contrôle d'accès.

Choix de type de chiffrement basé sur l'attributs :

Dans KP-ABE les textes chiffrés sont associés à des ensembles d'attributs descriptifs, et les clés des utilisateurs sont associées à des politiques, le propriétaire n'exerce aucun contrôle sur l'accès aux données chiffrées parce que les politiques d'accès sont définies dans les clés
Dans CP-ABE, le propriétaire doit être capable de décider qui doit ou ne pas avoir accès aux données chiffrées. CP-ABE permet un nouveau type de contrôle d'accès chiffré ou les clés privées de l'utilisateur sont spécifiées par des politiques d'accès.

Pour cela nous pensons que CP-ABE est le mieux adapté pour notre système pour imposer le contrôle d'accès sur les données. Pour un ensemble d'attribut envoyé par un utilisateur l'algorithme CP-ABE vérifie si cet ensemble peut générer une clé à utiliser plus tard pour décrypter ou accéder à un stockage de données ou non (l'ensemble d'attribut ne génère pas de clé implique accès refusé).

Politique d'accès CP-ABE :

La politique d'accès est définie par un administrateur ou un propriétaire de donnée, Cette politique d'accès peut être stocker dans une base de données liée à un contrôleur (ONOS) et aussi elle peut être stocker dans un fichier texte, comme elle peut être défini dans le code c'est-à-dire elle peut être une parie de code, dans ce travail on va utiliser le fichier texte pour définir notre politique d'accès choisie.

Dans cp-abe, une clé privée / id de chaque utilisateur est associée à un ensemble d'attributs, et au moment de demander d'accès à un centre de données/serveur une clé secrète est générée en appliquant une politique d'accès à la clé privée et l'ensemble d'attributs.

Les politiques d'accès sont définies au moment du cryptage par le propriétaire de données sur les attributs et une clé privée /id en utilisant un opérateur logique dans un fichier texte de conditions d'accès. Chaque politique d'accès est définie comme un accès en arborescence, une structure d'accès arborescent est spécifique aux conditions d'accès où les nœuds internes sont des opérateurs logiques et les nœuds externes sont les différents attributs.

L'accès est garanti uniquement pour les utilisateurs dont les attributs au moment de la génération de clé satisfont la structure d'accès arborescent i.e. la politique d'accès.

Les conditions d'accès sont des formules logiques construites en utilisant des valeurs spécifiées dans la liste d'attributs combinée avec des opérateurs logiques. Les Opérateurs logiques peuvent être utilisés pour définir les conditions d'accès telles que AND, OR et t de n porte seuil (où t de n attributs doit correspondre dans les conditions d'accès). Où l'opérateur AND est n de n porte seuil et opérateur OR est 1 de n porte seuil. Un utilisateur ne peut obtenir une clé secrète que s'il possède un ensemble d'attributs correspondant à une politique d'accès.

4.2.3 Conception UML :

Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système.

Il présente un aperçu graphique des fonctionnalités fournies par les acteurs du système du point de vue d'un observateur externe.

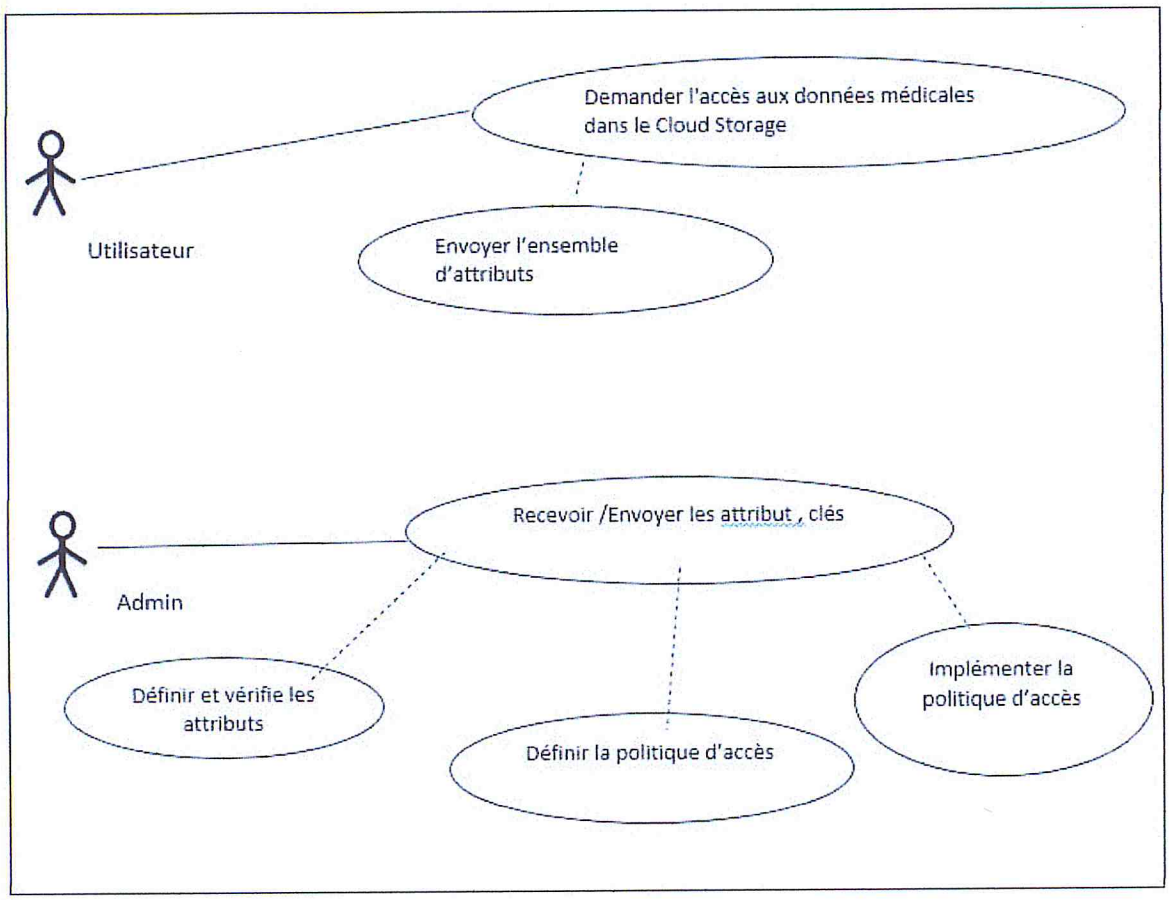


FIGURE 4.2 – Diagramme de cas d'utilisation .

Cette figure,figure 4.2 décrit les fonctionnalité fournis par chacun de l'utilisateur et le contrôleur/administrateur, où l'utilisateur demande l'accès au stockage de données en envoyant son ensemble d'attributs.
Le contrôleur ou l'administrateur a comme travail ou son rôle est de définir la politique d'accès, vérifie-la avec les attributs reçus et envoie sa réponse.

Diagramme de Séquence :

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs aux les objets au cœur d'un système interagissent en s'échangent des messages.et les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

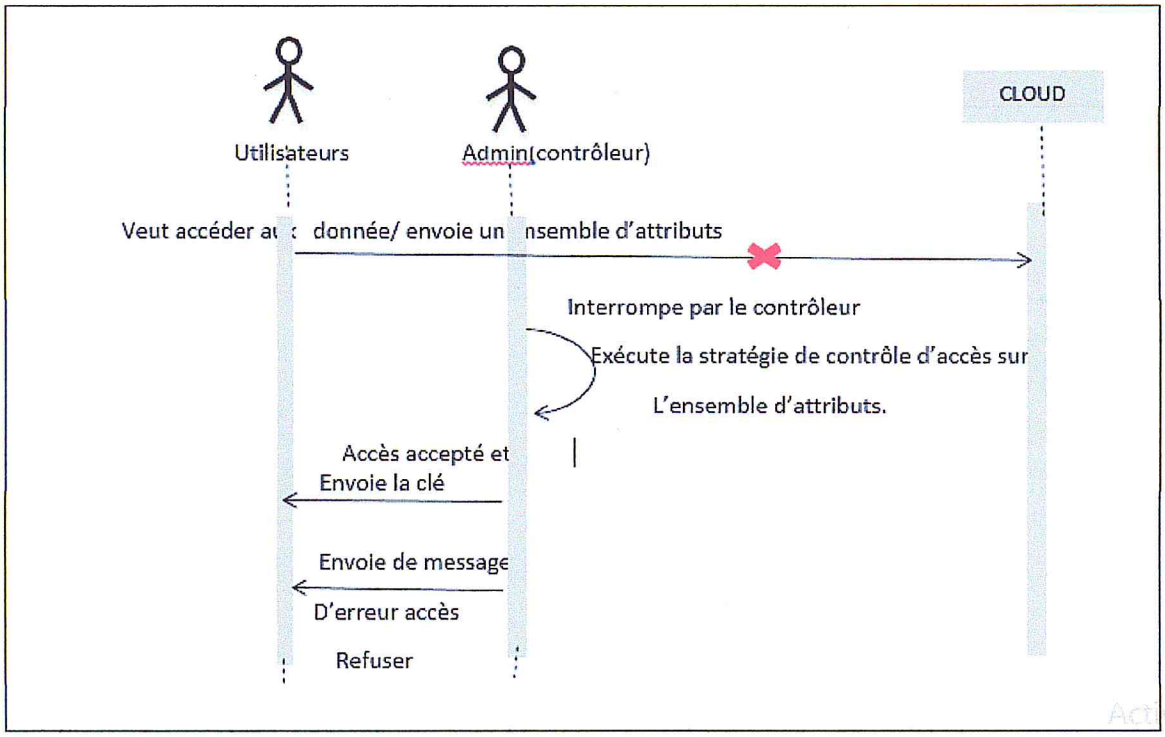


FIGURE 4.3 – Diagramme de séquence.

Cette figure (figure 4.4) décrit comment les éléments du notre système interagissent entre eux en utilisant l'échange de message, où l'utilisateur interagi avec notre système via une interface utilisateur.

L'administrateur ou bien le contrôleur, nous pouvons le considérer comme le système ici, interagi avec l'utilisateur en utilisant le packet-in et le packet-out en lisant les données de packet-in et en envoyant la réponse en utilisant le packet-out.

4.2.4 Algorithmes générales de système :

Algorithme d'interception des paquets

Cette algorithme (partie de code) explique comment le contrôleur SDN, ONOS va intercepter le paquet(ipv_4 protocole UDP) dans le réseau qui est envoyé par l'utilisateur afin d'analyser le paquet et lire les données ou les attributs de l'utilisateur, pour qu'il puisse exécuter son politique d'accès plus tard sur ces attributs.

L'algorithme prend comme entré un packet-in puis il l'analysera jusqu'à ce qu'il obtienne les données.

Algorithm 1 interception de paquets

Require: *packet_in*
Ensure: *Data*
  while *packet_in*! = *null* **do**
    Ethernet pkt ← *packet_in.Parsed*();
    if *pkt instanceof IPv4* **then**
      Ip4 ← *pkt.getpayload*();
      Source ← *ip4.getSourceAddress*();
      Destination ← *ip4.getDestinationAddress*();
    end if
    if *pkt instanceof UDP* **then**
      Udp_pkt ← *ip4.getpayload*() :
      Source_port ← *Udp_pkt.getSourcePort*();
      Destination_port ← *Udp_pkt.getDestinationPort*();
      Payload ← *Udp_pkt.getPayload*();
      Data ← *Payload.getPayload*();
    end if
  end while
  return *Data*;

Algorithme de génération des clés :

Algorithm 2 génération des clés

Require: *Data*;
Ensure: *privet_key*;
  while *Data*! = *null* **do**
    ParseData;
    Getmaster_keyfromData;
    Getset_attributefromData;
  end while*Access_policy* (*master_key,set_attribute*);
  return *privet_key*;

Algorithm 3 access_policy_function

Require: *master_key set_attribute*;
Ensure: *privet_key*; *Access_policy* (*master_key,set_attribute*);
  Random.generaekey();
  return *privet_key*;

ces deux algorithmes travaillent ensemble, ils prennent les données qui sont l'ensemble des attributs d'utilisateurs puis exécute la politique d'accès sur ces attributs afin de générer une clé secrète à l'utilisateur pour l'utiliser si la politique d'accès est vérifiée ou un message d'erreur sinon.

4.3 SECTION 2 Implémentation :

4.3.1 Environnement de développement :

Pour l'implémentation de notre système, nous avons utilisé trois machines de développement (trois machine virtuelles).

Machine 1 :

La premier machine est utilisée pour agir comme un client (utilisateur) qui va envoyer une demande/ message au serveur.

Configuration :

Système d'exploitation : LINUX [Ubuntu 16.04 LTS)

RAM : 2 GB

Processeur : 2 core CPU

Espace disque : 10 GB HDD

Machine 2 :

La deuxième machine est utilisée pour agir comme un serveur (Cloud).

Configuration :

Système d'exploitation : LINUX [Ubuntu 16.04 LTS)

RAM : 2 GB

Processeur : 2 core CPU

Espace disque : 10 GB HDD

Machine 3 :

La troisième machine est une machine virtuelle dédiée pour installer et configurer notre contrôleur ONOS.

Configuration :

Système d'exploitation : LINUX [Ubuntu 16.04 LTS)

RAM : 2 GB

Processeur : 2 core CPU

Espace disque : 10 GB HDD

Le switch OVS va être installé dans l'hyperviseur qui contient les trois machines virtuelles pour les liées.

4.3.2 Mise en place de l'environnement :

Contrôleur :

concernant l'implémentation du contrôleur nous avons choisi de l'installer sur une machine virtuelle.

A travers cette partie nous nous intéressons au contrôleur et aux étapes nécessaires à sa mise en place.

- **Etape 1 vérification de prérequis :** Mise en place de JAVA8. $ Sudo apt-get update
  $ Sudo apt-get install software-properties-common -y
  $ Sudo add-apt-repository ppa :webupd8team/java -y
  $ Sudo apt-get update
  $ Sudo apt-get install oracle-java8-installer oracle-java8-set-default -y mise en place de KARAF et MAVEN. $ cd ; mkdir Downloads Applications
  $ cd Downloads
  $ wget http ://archive.apache.org/dist/karaf/3.0.5/apache-karaf-3.0.5.tar.gz
  $ wget http ://archive.apache.org/dist/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz
  $ tar -zxvf apache-karaf-3.0.5.tar.gz -C ../Applications/
  $ tar -zxvf apache-maven-3.3.9-bin.tar.gz -C ../Applications/

- **Etape 2 clonage de contrôleur ONOS :** $ git clone https ://gerrit.onosproject.org/onos -b version
  $ cd onos git checkout version
- **Etape 3 configuration des modules nécessaire :** $ gedit /.bashrc
  on ajoute cette line a la fin de fichier et on l'enregistre.
  . /onos/tools/dev/bash$_{profile}$
  $$source.bashrc$$
  $$env$$
  $$env|grepJA$$
  $$env|grepKA$$

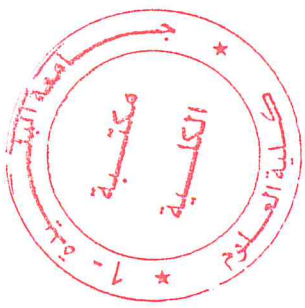


FIGURE 4.4 – lancement d'onos.

$export ONOS_ROOT = /onos$

$source $ONOS_ROOT/tools/dev/bash_profile$

$cd /onos$

$$ONOS_ROOT/tools/build/onos - buck build onos - -show - output$$

$onos $tools/build/onos - buck run onos - local - -clean debug$ dans un nouveau terminal $ cd onos

onos$tools/test/bin/onos localhost ou address de la machine

A ce niveau, nous avons obtenu onos, la figure 4 montre le démarrage d'onos :

Switch :

Cette partie décrite comment installer l'Open vSwitch. $ Sudo apt-get install openvswitch

Demarche de application d'onos :

dans cette partie de la deuxième section nous avons essayé de mettre en place une démonstration de la marche de notre application ONOS, SIR : la première chose à faire est de créer un pont avec 3 différentes portes et chaque machine virtuelle avec une de ces portes.

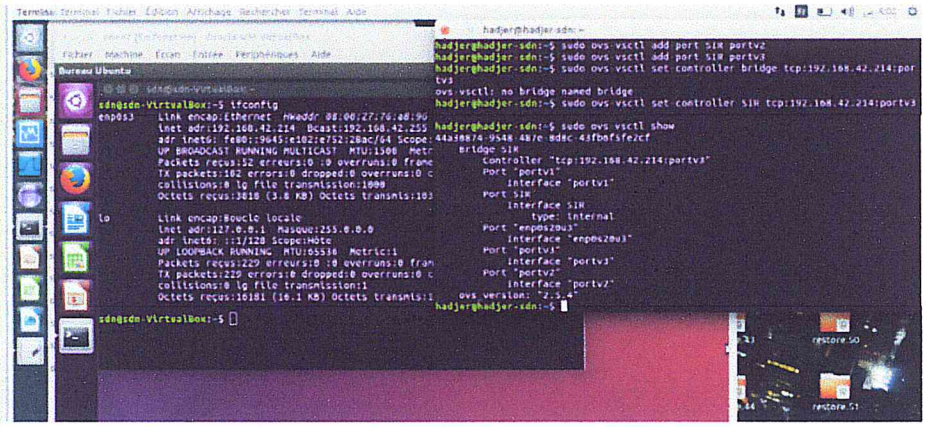


FIGURE 4.5 – création et activation de pont et portes

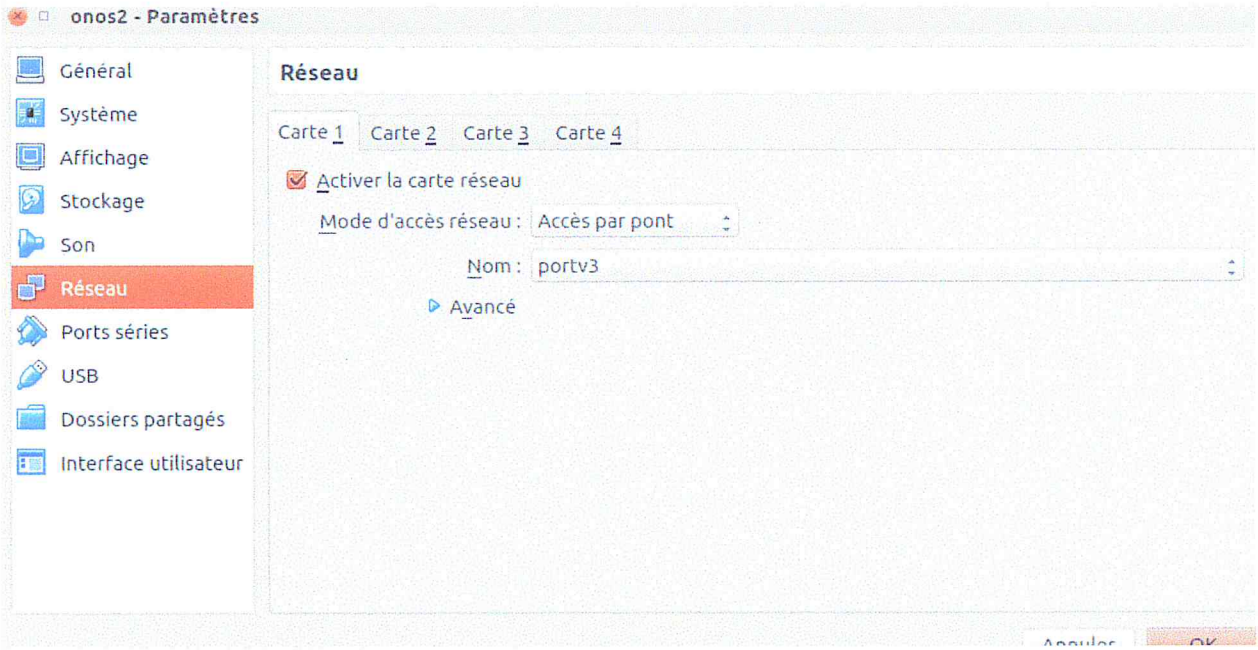


FIGURE 4.6 – Mode d'accès réseau dans la vm onos.

où le client dans la machine vm1 est lié à ovs pont via la porte 1, la machine serveur vm2 est liée à ovs pont via la porte 2 et la machine virtuelle du contrôleur onos 2 est liée au pont ovs via la porte 3.

le pont ovs est connecté à Internet et à travers le pont ovs tous les 3 machines sont également connectés les uns aux autres et à Internet, le pont ovs donné à toutes les 3 machines client, serveur et onos, adresses ip comme le montre la figure 4.9 Après avoir créé le pont et ses portes, et configuré le vms avec le pont, nous pouvons connaître la tête de l'application ONOS, SIR et l'installer sur le contrôleur ONOS et l'activer,après l'avoir construit. Une fois l'application installée et activée, nous pouvons lancer notre communication entre les hôtes ou entre le client et le serveur en utilisant les Sockets,afin que l'onos interrompt la communication et intercepte

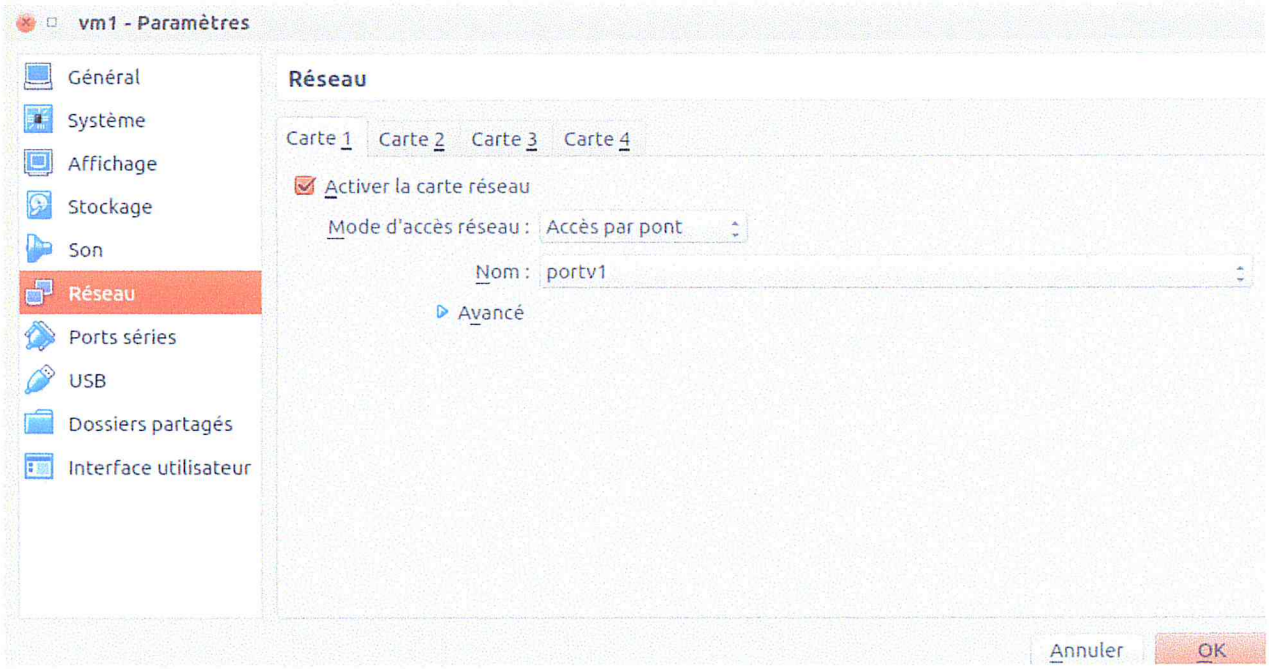


FIGURE 4.7 – Mode d'accès réseau dans la vm1.

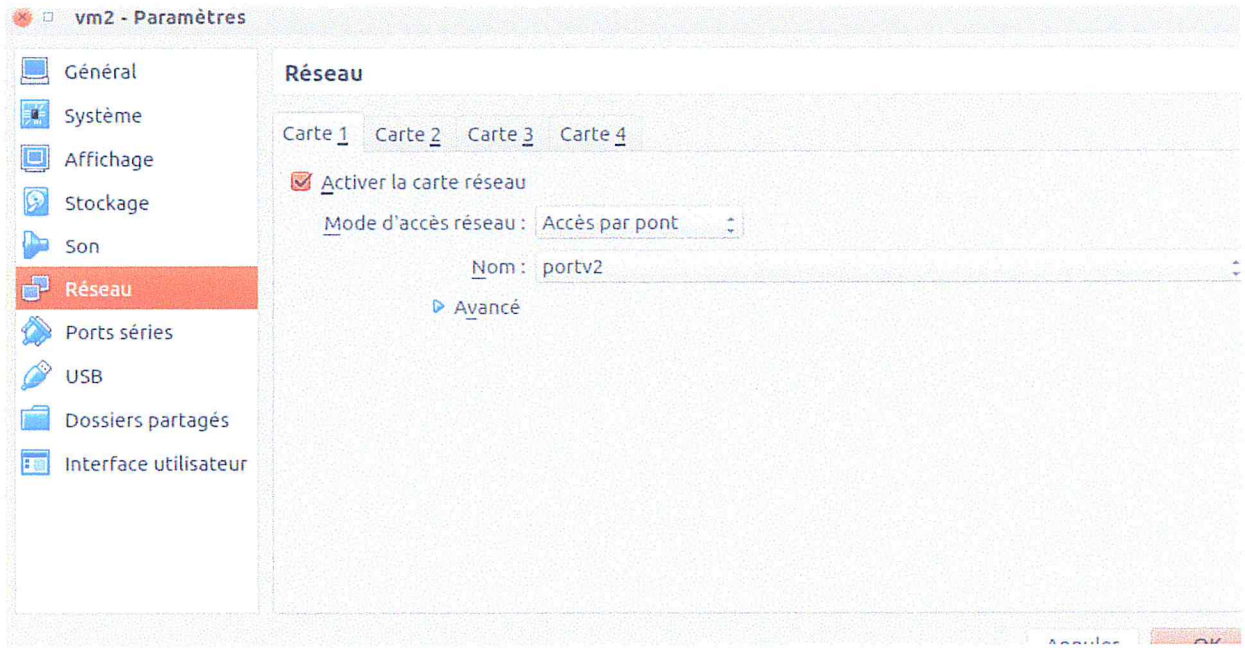


FIGURE 4.8 – Mode d'accès réseau dans la vm 2.

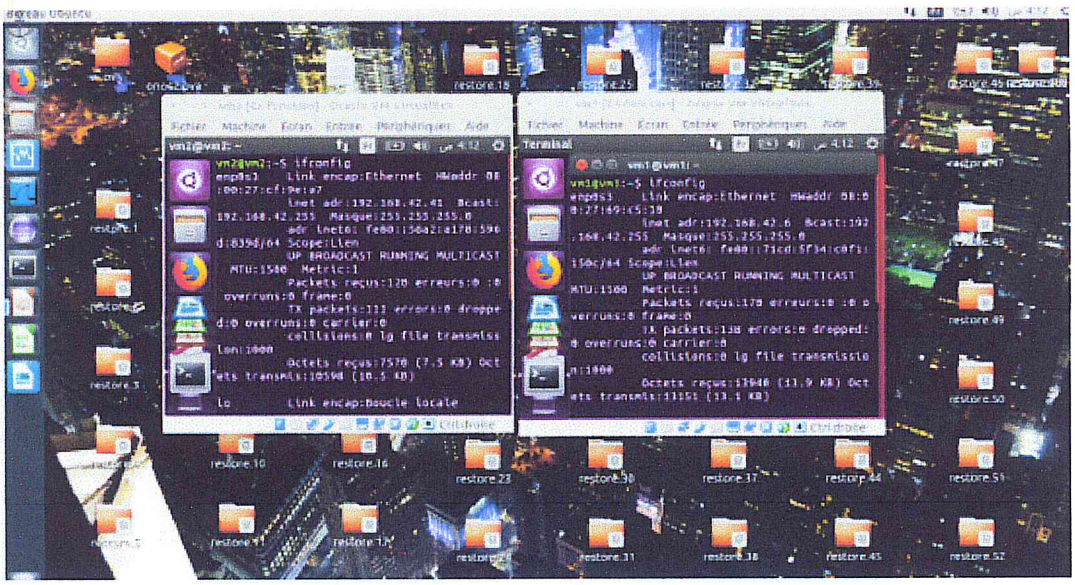


FIGURE 4.9 – Mode d'accès réseau dans la vm1.

le paquet envoyé par l'utilisateur au serveur pour obtenir les données et générer la clé secrète pour l'utilisateur. la figure montre l'affichage du journal dans le terminal onos où nous pouvons voir les informations de paquets interceptés tels que les adresses de source et de destination, le protocole et les données envoyées par le client à partir de son interface. En parlant de l'interface de l'utilisateur, les figures montrent comment et ce que l'utilisateur enverra comme données au serveur qui sera interrompu par l'application SIR dans le contrôleur ONOS pour générer la clé et l'envoyer à l'utilisateur. la figure est une vue globale de l'interface utilisateur où l'utilisateur définit son ensemble d'attributs (fonction, spesialité, grade, année et ID) afin de l'envoyer comme indiqué sur la figure, pour lui obtenir sa clé secrète en retour.

Une fois que l'utilisateur a envoyé son ensemble d'attributs, une réponse lui sera envoyée par le contrôleur avec une clé secrète lui indiquant la zone de texte que l'utilisateur a le choix de télécharger ultérieurement dans un fichier texte en cliquant sur le bouton clé privée. les figures montre le processus de réception et de téléchargement de la clé. cette dernière figure, figure 4.15 présente le fichier téléchargé par l'utilisateur qui est la clé reçue.

nous avons essayé dans ces dernières étapes de ce chapitre, de démontrer le fonctionnement de l'application ONOS et le rôle de l'utilisateur via son interface.

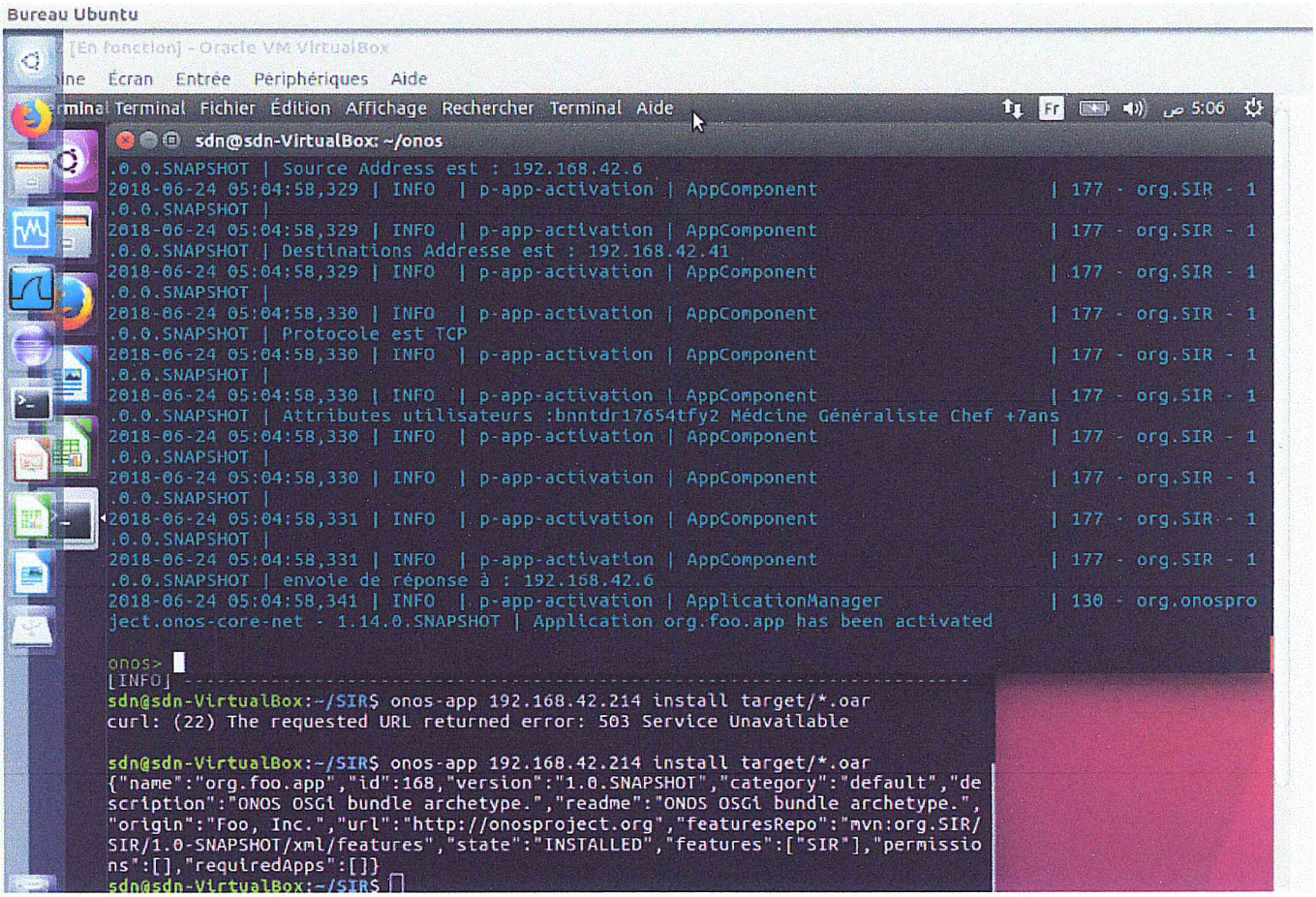


FIGURE 4.10 – démarche d'application.

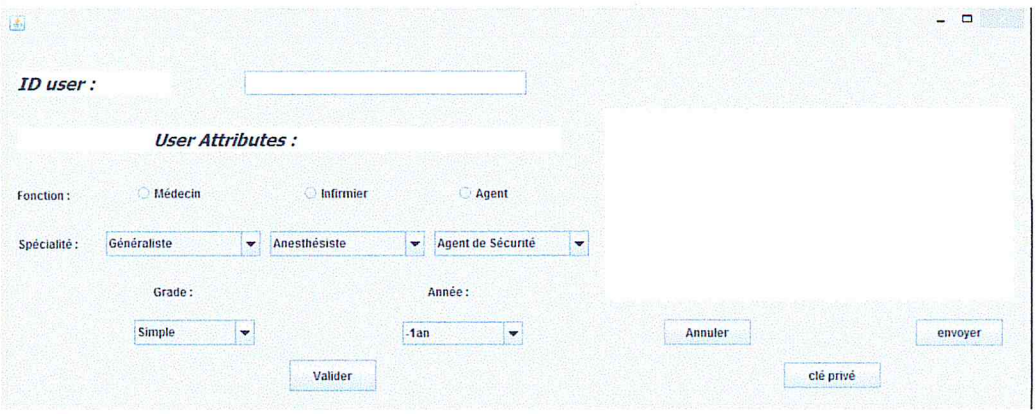


FIGURE 4.11 – interface utilisateurs.

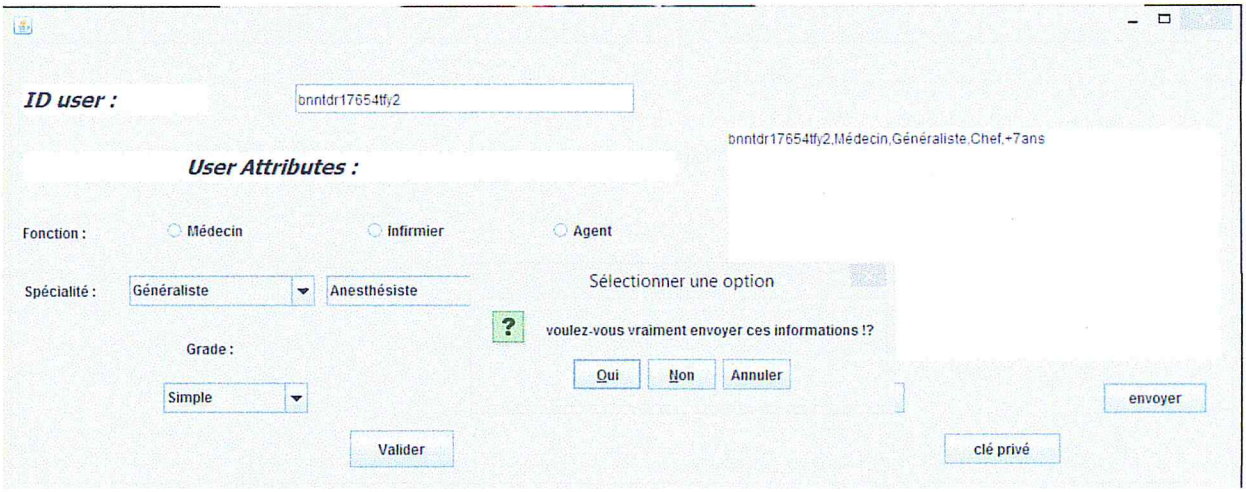


FIGURE 4.12 – envoie des attributes .

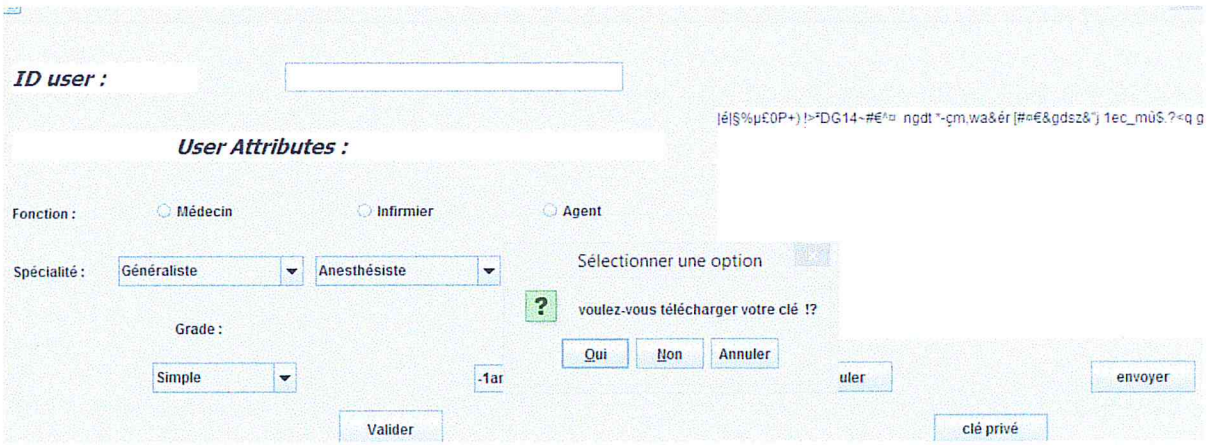


FIGURE 4.13 – recevoir la clé .

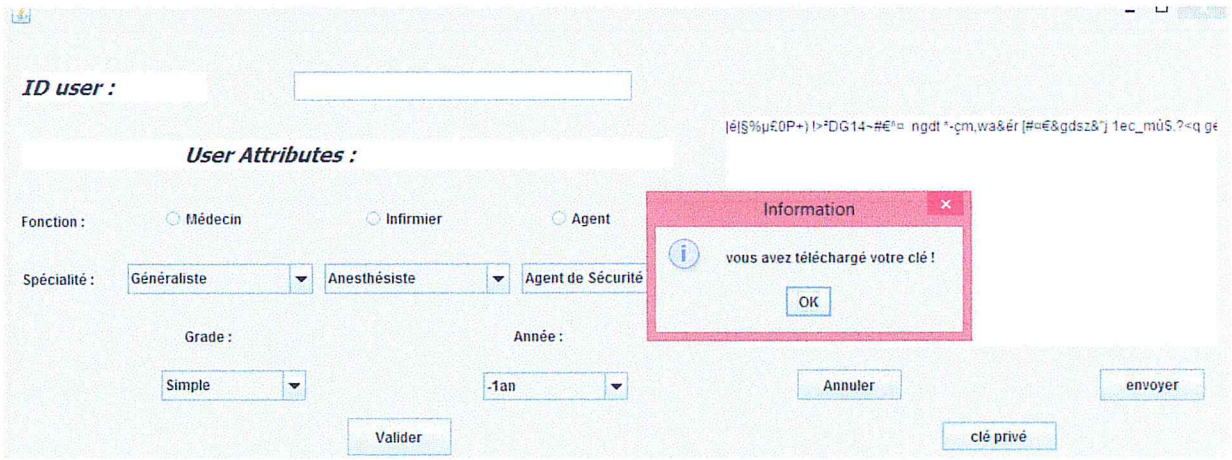


FIGURE 4.14 – télécharger la clé .

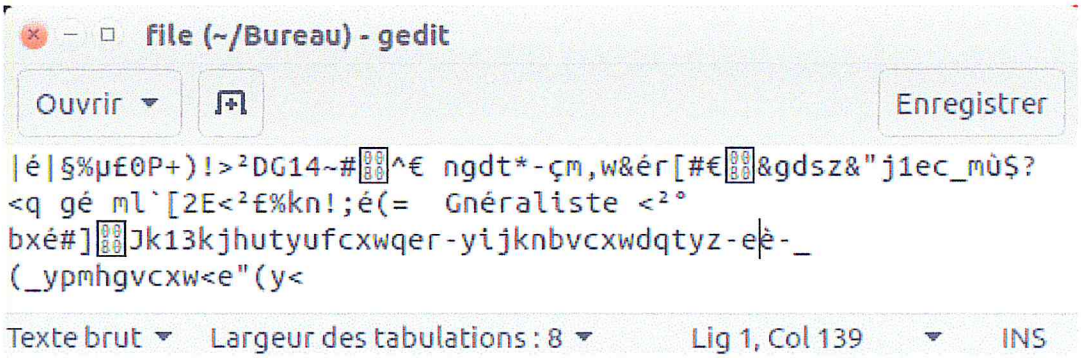


FIGURE 4.15 – la clé secrte.

4.4 Conclusion :

Dans ce chapitre, nous avons présenté notre approche pour la conception d'un système de contrôle d'accès basé sur le chiffrement par attributs dans une architecture réseau SDN. La deuxième partie a permis de montrer la manière avec laquelle nous avons procédée dans la mise en place de la solution en sa globalité et l'intégration du contrôle d'accès ABE en son sein. Nous avons aussi présenté les points essentiels de l'implémentation et la programmation de cette conception / application. Cette solution reste adaptable à différents cas de figures et donne la possibilité d'évolutions futures.

Conclusion :

Sdn promet d'introduire la flexibilité et la programmabilité dans le réseau en extrayant le plan de contrôle d'un réseau dans une entité contrôleur dédiée.

Ce dernier rend le réseau fiable et programmable en utilisant des applications implémentées dans le contrôleur,en utilisant cet aspect ou fonctionnalité de contrôleur un système de contrôle d'accès peut être créé ou mis en œuvre dans le contrôleur.

Contrôle d'accès est l'une des tâches les plus importantes dans la gestion des systèmes, réseaux, serveurs, Clouds, bases de données ... etc.

L'objectif est de permettre à tous les utilisateurs légitimes d'avoir exactement les niveaux d'accès qu'ils sont supposés avoir et d'empêcher tout utilisateur ou toute demande illégitime d'accéder à des ressources ou des ressources internes. Le cryptage à base d'attributs présente à côté du cryptage et du décryptage de documents un système de contrôle d'accès à grain fin de documents.

Dans notre project de fin d'étude nous avons conçu et implémenté un système de contrôle d'accès dans un réseau basé SDN en utilisant le chiffrement basé sur les attributs ABE.

Nous avons présenté un cp-abe-app, une application de contrôle d'accès dans un réseau défini par logiciel avec une nouvelle méthode d'application de la politique en utilisant la politique d'accès au cryptage basé sur les attributs.

Notre système est utilisé pour garantir un contrôle d'accès précis au stockage des données (données médicales) sur un serveur(cloud) en interceptant la demande d'un utilisateur à l'aide d'une application de contrôleur qui utilise les attributs de l'utilisateur et une politique d'accès mise en œuvre pour générer une clé secrète si l'ensemble d'attributs vérifie la politique d'accès définie.

Ce travail été pour nous un grand rapport et trés bénéfique, il nous a permis d'acquérir de nouvelle connaissances et technologies :

- Nous avons eu la possibilité de connaître et d'expérimenter le nouvel aspect du réseau programmable et flexible SDN.
- aussi le système de cryptage/décryptage et contrôle d'accès ABE, qui non seulement est un outil de décryptage / cryptage des données mais crée également un système de contrôle d'accès utilisant une politique d'accès construite à l'aide d'un ensemble d'attributs.
- Nous avons pu travailler sur le contrôleur SDN ONOS en apprenant comment créer,

programmer et activer une application ONOS dans le réseau et tout sur le protocole openflow.

— Ajoutant à cela, nous arrivons à connaître OVS le commutateur virtuel qui relie le contrôleur au réseau.

perspective :

Dans ce travail, nous avons prouvé que les applications de contrôleur SDN peuvent être utilisées pas juste pour contrôler et surveiller le réseau et flux d'entrée/sortie, mais aussi pour créer de vraies applications comme une application de contrôle d'accès qui n'est pas faite pour vérifier l'adresse IP ou mac address, dans notre cas un système de contrôle d'accès ou une application pour accéder au stockage de données.

Pour les travaux futurs, cette application peut être utilisée et complétée en ajoutant le côté serveur à ces équations, et cela peut être fait en relayant l'utilisateur au serveur après lui avoir envoyé la clé secrète générée après vérification de la politique d'accès définie, afin que il peut télécharger et déchiffrer les données.

En ajoutant à cela, nous pouvons ajouter aussi un autre aspects de chiffrement basé sur attributs tel que la révocation des attributs .

Références

[1] https ://www.lemagit.fr/conseil/Introduction-aux-technologies-SDN-evolution-ou-revolution-architecturale

[2] Messaoud AOUADJ AirNet : le modèle de virtualisation "Edge-Fabric" comme plan de controle pour les réseaux programmables THESE l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier).

[3] Software Defined Networking for Systems and Network Administration Programs Bruce Hartpence,Rossi Rosario The USENIX Journal of Education in System Administration Volume 2, Number 1 • November 2016.

[4] SOFTWARE-DEFINED NETWORKING(SDN) A New Approach to Networking Anju Ann Joseph Semester : VII Batch : C B-Tech Seminar Sept 2013 [En ligne]. Disponible : https ://www.slideshare.net/anjuann1232/sdn-new .

[5] SOFTWARE-DEFINED NETWORKING(SDN) A New Approach to Networking Anju Ann Joseph Semester : VII Batch : C B-Tech Seminar Sept 2013 [En ligne]. Disponible : https ://www.slideshare.net/anjuann1232/sdn-new .

[6] Fouad Benamrane Etude des Performances des Architectures du Plan de Contrôle des Réseaux 'Software-Defined Networks', January 2017 .

[7] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown et S. Shenker, "NOX : Towards an Operating System for Networks", ACM SIGCOMM Computer Communication Review,38(3), pp. 105–110, 2008.

[8] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown et S. Shenker, "Ethane : Taking Control of the Enterprise", Proceedings of the Conference on Applications, Technologies, Architectures,and Protocols for Computer Communications, pp. 1–12, 2007.

[9] https ://www.ciena.com/insights/what-is/What-is-SDN.html 11/12/2017 00 :37

[10] Software-Defined Networking :The New Norm for Networks ONF White Paper April 13, 2012

[11] Software-Defined Networking (SDN) Definition, [En ligne]. Disponible : https ://www.opennetworking.org/sdn-resources/sdn-definition, 2016 10/12/2017 23 :26

[12] B. Heller, R. Sherwood and N. McKeown, "The controller placement problem," in Proceedings of the first workshop on Hot topics in software defined networks. ACM,(2012) : 7-12.

[13] S. Schmid and J. Suomela, "Exploiting locality in distributed sdn control," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM,(2013) :121-126.

[14] Canini, Marco, et al., "Software transactional networking : Concurrent and consistent policy composition," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM,(2013) : 1-6

[15] La qualité de service au niveau iaas : vers l'utilisation du concept software-defined networking (SDN) par Kodzo Mawuessénam PARKOO Amadou Hampâté-Bâ de Dakar - Master 2015.

[16] La qualité de service au niveau iaas : vers l'utilisation du concept software-defined networking (SDN) par Kodzo Mawuessénam PARKOO Amadou Hampâté-Bâ de Dakar - Master 2015.

[17] Software-Defined Networking : A Comprehensive Survey 8 Oct 2014 Diego Kreutz, Member, IEEE, Fernando M. V. Ramos, Member, IEEE, Paulo Ve-

rissimo, Fellow, IEEE,Christian Esteve Rothenberg, Member, IEEE, Siamak Azodolm

[18] Enric Caceres, « Le Protocole OpenFlow dans l'Architecture SDN », EFORT 2016.Ahmed Sonba et Hassan Abdalkreim, « Performance Comparison Of the state of the art Openflow Controllers », mémoire de Master , Université Halmstad Sweden, Décembre 2014.

[19] Software-Defined Networking and Network Programmability : Use Cases for Defense and Intelligence Communities Mark "Mitch" Mitchiner Solutions Architect U.S. Federal Area Reema Prasad Solutions Architect U.S. Federal Area

[20] Considerations for Software Defined Networking (SDN) : Approaches and use cases Kapil Bakshi Cisco Systems, Inc., 13635 Dulles Technology Drive, Herndon, VA 20171, USA

[21] A Survey of Software-Defined Networking : Past, Present, and Future of Programmable Networks 13 February 2014 Bruno Astuto A. Nunes Marc Mendonca Xuan-Nam Nguyen

[22] Thomas Paradis, « Software-Defined Networkin », mémoire de Master, School of Information and Communication Technology KTH Royal Institute of Technology Stockholm, Sweden, le 20 Janvier.

[23] Software-Defined Networking :A Comprehensive Survey Diego Kreutz, Member, IEEE, Fernando M. V. Ramos, Member, IEEE, Paulo Verissimo, Fellow, IEEE,Christian Esteve Rothenberg, Member, IEEE, Siamak Azodolmolky, Senior Member, IEEE,and Steve Uhlig, Member, IEEE.

[24] Advancing Software-Defined Networks : A Survey October 12, 2017 by JACOB H. COX , JR. JOAQUIN CHUNG SEAN DONOVAN JARED IVEY

USSELL J. CLARK (Member, IEEE), GEORGE RILEY AND HENRY L. OWEN, III (Senior Member, IEEE).

[25] Explication des trois modèles de SDN [en ligne] Disponible : https ://www.lemagit.fr/conseil/Explication-des-trois-modeles-de-SDN

[26] Les inconvénients d'une infrastructure programmable [en ligne] Disponible : https ://www.lemagit.fr/conseil/Les-inconvenients-dune-infrastructure-programmable

[27] Performance Evaluation of Attribute-Based Encryption : Toward Data Privacy in the IoT.

[28] Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers

[29] http ://www.techotopia.com/index.php/Mandatory,_Discretionary,_Role_and _Rule_Based _Access_Control

[30] Cheng-Chi Lee, Pei-Shan Chung, and Min-Shiang Hwang,A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments

[31] Chi-Wei Liu, Wei-Fu Hsien, Chou-Chen Yang, and Min-Shiang Hwang,A Survey of Attribute-based Access Control with User Revocation in Cloud Data Storage

[32] Parmar Vipul Kumar Ja. RajaniKanth Aluvalu,Key Policy Attribute Based Encryption (KP-ABE) : A Review

[33] Object-Based Security with Attribute-Based Encryption

[34] Protection des données par une méthode cryptographique basée attributs

[35] Protection des données par une méthode cryptographique basée attributs

[36] M. Chase. "Multi-authority attribute based encryption". In TCC, pages 515-534, 2007

[37] Netfpga platform, http ://netfpga.org.

[38] OpenFlow Specification (ONF), https ://www.opennetworking .org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf.

[39] Devolved Control of ATM Networks, http ://www.cl.cam.ac.uk/research/srg/netos/old-projects/dcan/pub.

[40] Cisco OnePK, https ://developer.cisco.com/site/onepk/.

[41] D.Erickson, "The beacon openflow controller," inProceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13), pp. 13–18, Hong Kong, August 2013.

[42] https ://www.opennetworking.org/platforms/onos/ (26/02/2018 12 :33)

[43] http ://sdnhub.org/tutorials/onos/]

[44] R. Enns, "NETCONF Configuration Protocol," RFC 4741 (Pro-posed Standard), Obsoleted by RFC 6241, December 2006.

[45] CERIAS : GeoPlex : Universal Service Platform for IP Network-based Services—10/17/1997, 2014, http ://www.cerias .purdue.edu/.

[46] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In VINI veritas : realistic and controlled network experimenta- tion," ACM SIGCOMM Computer Communication Review,vol. 36,no.4,pp.3–14,2006.

[47] M.Sridharan,K.Duda,I.Ganga,A.Greenberg,G.Lin,M.Pear- son et al., NVGRE : network virtualization using generic routing encapsulation, Internet Engineering Task Force, September 2011.

[48] B. Davie and J. Gross, STT : A Stateless Transport Tunneling Protocol for Network Virtualization (STT), Internet Engineering Task Force, 2012.

[49] Q.Duan,Y.H.Yan,andA.V.Vasilakos,"Asurveyonservice- oriented network virtualization toward convergence of net- working and cloud computing," IEEE Transactions on Network and Service Management,vol.9,no.4,pp.373–392,2012.

[50] Q.Duan,Y.H.Yan,andA.V.Vasilakos,"Asurveyonservice- oriented network virtualization toward convergence of net- working and cloud computing," IEEE Transactions on Network and Service Management,vol.9,no.4,pp.373–392,2012.

[51] Ryu, http ://osrg.github.com/ryu/.

[52] Y.Jarraya,T.Madi,andM.Debbabi,"Asurveyandalayeredtax- onomy of software-defined networking," IEEE Communications Surveys Tutorials,vol.16,no.4,pp.1955–1980,2014.

[53] Pfaff,B., Davie, B. (2013. The Open vSwitch database management protocol

1 Annexe

Appendix A : OpenVSwitch commandes

Appendix A.1 : Création de switch (bridge)

```
$ sudo ovs-vsctl add-br nom-switch
```

Le résultat de cette commande est :

```
# ovs-vsctl show
051fd463-96f5-4529-a55a-ff0b03004ffc
    Bridge "br0"
        Port "br0"
            Interface "br0"
                type: internal
    ovs_version: "2.3.0"
```

Appendix A.2 : une interface existante a switch

```
$ sudo ovs-vsctl add-port nom-switch nom-port
$ sudo ifconfig nom-port 0
$ sudo ifconfig nom-switch up
$ sudo dhclient nom-switch // pour configurer le switch
```

Appendix A.3 : Ajouter un périphérique TAP

1. Nous ajoutons une interface pour la connecter à notre virtuel machine plus tard .

```
$ sudo ip tuntap add mode tap nom-tap
```

2. Ajouter le nouveau périphérique TAP à notre switch

```
$ sudo ovs-vsctl add-port nom-switch nom-tap
```

ifconfig nom-tap up Le resultat de cette étape est :

```
> sudo ovs-vsctl show

9e72385f-ed0a-40fd-97f3-21d49cbf60f3
    Bridge my_bridge
        Port my_bridge
            Interface my_bridge
                type: internal
        Port "virt_port"
            Interface "virt_port"
    ovs_version: "2.5.0"
```

Appendix A.4 : liaison du switch à un contrôleur

```
$ sudo ovs-vsctl set-controller nom-switch tcp :Addrsse-ip-de- switch :nom-port
```

Appendix B : ONOS commandes

Appendix B.1 : Accès à ONOS

Teminal1 : // lancement du serveur ONOS

```
$ cd onos
$ tools/build/onos-buck run onos-local – clean debug
```

Termina2 : // lancement du ONOS

```
$ cd onos
$ tools/test/bin/onos @ip //ou localhost pour lancement local
```

Appendix B.2 : Création Application ONOS

```
$ onos-create-app app org.nom-app nom-app 1.0-SNAPSHOT
```

Appendix B.3 : Installation du l'application ONOS

```
$ cd nom-app
$ mci // mvn clean install
$ onos-app Addrsse-ip install ! target/*.oar
```

Dans le 2eme terminal

```
Onos> apps -a -s // pour voir les apps ONOS
Onos> app activate org.foo.app
```