RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE Dr. TAHAR MOULAY SAIDA

FACULTE : TECHNOLOGIE

DEPARTEMENT : INFORMATIQUE

MÉMOIRE DE MASTER

Option : sic

Sécurité informatique et cryptographie

Etude et analyse de la cryptographie basé sur attribut-ABE

Présenté par :

Lakhache omar

Fettehallah djamila

Encadré par :

Dr Houacine abdelkrim

Année Universitaire 2020-2021

بسم الله الرحمن الرحيم

ملخص:

باعتبار أن حماية المعلومات أصبح مشكل عويص عدة أعمال أنجزت في هذا السياق و لا يزال اقتراح حلول جديدة قائما

في حين تعاني الكثير من هذه الأعمال نقص واضح خاصة عند استغلال المعلومة عن طريق الشبكة و بهدف حماية المعلومة بطريقة أفضل أثناء استعمال الشبكة مع الأخذ بعين الاعتبار الموارد المحدودة قمنا باقتراح نظام التشفير مختلط يسمح باستخدام الطريقة المتناظرة ويقوم على أساس دمج المزايا الحسنة

بحيث أن الهدف الرئيسي من هذه الدراسة يرتكز أساسا على عنصرين هامين :1-الوقت المحدد لمعالجة المعلومة 2-مستوى تامين وحماية المعلومة في حد ذاتها

كلمات مفتاحيه:

تشفير،امن، المفتاح العمومي،الإسناد هوية

Résumé :

Actuellement, La protection des données sensibles est devenue une problématique clé. Plusieurs travaux sont réalisés et plusieurs solutions sont proposées théoriquement.

Cependant, La majorité de ceux-ci ne prennent pas en considération les contraintes imposées par les différents réseaux.

Afin d'assurer la sécurité des données sensibles échangées au sein des différents réseaux en tenant compte de la capacité des ressources limitées de ses terminaux, nous avons proposé un cryptosystème hybride qui utilise la méthode symétrique qui est essentiellement une combinaison des avantages de chacune.

Ainsi, nous avons développé une application qui utilise le cryptosystème proposé. L'efficacité de cette étude dépend de deux paramètres : la durée nécessaire au traitement des données et le niveau de sécurité fourni par l'application en fonction des ressources disponibles. Les résultats obtenus encouragent la continuité de notre travail.

Mots clés :

cryptographie, sécurité, clé publique, identité, attribut.

ABSTRACT

Currently, the protection of sensitive data become a key issue.Several works are carried out and several solutions are theoretically proposed. However, the majority of these do not take into consideration the constraints imposed by different networks. In order to ensure the security of sensitive data exchanged within different networks, taking into account the capacity of the limited resources of its terminals, we have proposed a hybrid cryptosystem which uses the symmetric method and, which is essentially a combination of the benefits of each.

Thus, we have developed an application that uses the proposed cryptosystem. The effectiveness of this study depends on two parameters: the time required to process the data and the level of security provided by the application according to the available resources. The results obtained encourage the continuity of our work.

Word Key :

cryptography securite ,public Key, identite,attribut

Remrciement

Nous tenons avant tout de remercier le bon DIEU qui nous a donné la volonté et le courage pour réaliser ce modeste travail. Du fond du cœur nous remercions nos chers parents qui nous ont toujours guidé, encouragé et qui ont fait de leurs mieux pour que nous arrivons là aujourd'hui.

A Mr l'encadreur ainsi Que les membres de jury trouvent ici l'expression de nos sincères remerciements pour l'honneur qu'ils nous font en acceptant de juger notre travail

Merci à tous

Dédicace

À nos parents pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de nos études, À nos chères frères et sœurs pour leur encouragement constant et leur soutien moral,

A toutes nos familles pour les accompagner tout au long de notre carrière universitaire, Que cette œuvre soit la réalisation de vos soi-disant désirs et une évasion de votre soutien indéfectible, Merci d'être toujours avec nous

Table des matières

1 Introduction generale……01

CHAPITRE « I » Introduction à la cryptographie

1-Introduction ……02

2- Définition……02

3-Définition de la cryptographie ……02

4- L'usage de la cryptographie ……03

5- Mécanisme de la cryptographie……04

6-Confidentialité et algorithmes de chiffrement ……05

6.1- Les avantages et inconvénients de la cryptographie à clé publique comparé à la cryptographie à clé privée ……06

7-Différence entre chiffrement et codage……07

8- Définition de la cryptanalyse ……08

8.1- Les niveaux d'attaques……09

9- Conclusion ……10

CHAPITRE « II » - IBE : Identity-Based Encryption La cryptographie basée sur l'identité

Introduction ……24

1-Définitions ……24

1.1- La cryptographie à clé publique ……24

1.2- La cryptographie basée sur l'identité ……25

2-La Sécurité des Systèmes de chiffrement basés sur l'Identité ……26

2.1- La Sécurité sémantique ……..……26

2.1 -La Sécurité du texte chiffré choisi ……27

2.2- La Sécurité sémantique d'un schéma basé sur l'identité ……28

3- La Sécurité du texte chiffré choisi d'un schéma IBE ……..28

3.1- Comparaison cryptographie à clé publique /traditionnelle……28

3.2 -L'authenticité Dans un système IBE……..29

3.3 Enregistrement à une autorité ……..29

3.4 La révocation des clés ……30

3.5 La distribution des clés ……30

3.6- La sécurité de la clé ……31

3.7 Délégation des clés ……31

4.conclusion……31

CHAPITRE III - Ciphertext-Policy Attribute Based ENCRYPTION (CP-ABE)

1 – Introduction……19

2 – C'est quoi CP-ABE ?……19

2.1- L'idée Dans le protocole "CP-ABE"……20

2.2- Les étapes Le protocole "CP-ABE" ……21

2.3-Structure d'accès associée à ABE……21

2.4- Contrôle d'accès CP-ABE ……22

3-Chiffrement KP-ABE ……23

4 -Comparaison entre KP-ABE et CP-ABE ……23

5-conclusion ……24

CHAPITRE IVEtude et Analyse

Introduction…….…….25
D'abord qu'est-ce que docker ……25

Comment installer docker sur Windows …….26

Quels sont les avantages du docker ……...26

Docker vers virtuelle machine……...27

Support du système d'exploitation……..…28

Comparaison entre une virtuelle machine et conteneur docker…….…30

Le système d'exploitation UBUNTU……30

Compasants……..31

Interface UBUNTU……..32

Création d'une nouvelle machine Virtuelle sous le nom UBUNTO……....32
L'installation du docker……..……..……..33
DockerHub…….……..……..33
Chargement de l'image OpenABE……..34
L'explication du CP-ABE……..35
Le Scenario de la cryptographie des fichiers……..35

Phase1 initialisation des clés ……..36

Phase 2 génération des clés :……..37

Phase 3 : le chiffrement……...37

Phase 4 le déchiffrement ……...38

Phase 4 le déchiffrement ……..39

Conclusion chapitre ……....40

Conclusion Générale……49

Table des Figures

Figure 1.1 schéma de cryptage…….15

Figure 1.2 cryptographies à clé publique……..……..17

Figure 1.3 chiffrements et codage……..20

Figure 2.4 cryptographies basées sur l'identité…….26

Figure 3.5 : Phases d'initialisation et de génération de clés……..33

Figure 3.6 : Phases de chiffrement et déchiffrement……..34

Figure 3.7 – Arbre d'accès pour un exemple de politique simple…….35

Figure 3.8_CP-ABE (Ciphertext-Policy Attribute Based Encryption)……36

Figure 3.9 KP-ABE (Key-Policy Attribut Based Encryption)……37

Figure 4.10 : Interface docker …….40

Figure 4.11 : la différence entre la virtualisation et la conteneurisation……41

Figure 4.12 Support du S.E…….44

Figure 4.13 interface IBUNTU……..49

Figure 4.14 installation IBUNTU……..50

Figure 4.15 installation docker…….……..…….51

Figure 4.16 installation docker HUB…….52

Figure 4.17 Chargement de l'image OpenABE……..53

Figure 4.18 L'explication du CP-ABE ...……54

Figure 4.19 Scenario de la cryptographie des fichiers ……..55

Figure 4.20 :initialisation des clés…….56

Figure 4.21 Phase générations des clés ……57

Figure 4.22 Phase le chiffrement…..……..58

Figure 4.23 Phase le déchiffrement……59

Figure 4.24 Phase le déchiffrement……60

Introduction générale

1-Introduction

Petite Vue sur défis majeurs relevant lors de la conception d'un système de chiffrement Sachant que le système de chiffrement fait inter venir des opérations cryptographiques plus ou moins complexes, dépendamment du mode de chiffrement choisi ,dont l'objectif est la conception et la réalisation d'un système de chiffrement qui assure la confidentialité et l'intégrité par une approches de La cryptographie traditionnelle qui nous montre généralement l'existence deux modes de chiffrements (symétrique et asymétrique).Ou l'avantage principale dans Le chiffrement symétrique est d'être rapide mais présente plusieurs lacunes. Entre chaque paire de participants, on prévoit une clé secrète pour pouvoir établir une communication sécuritaire.

Nous proposons une solution basée sur la cryptographie à clé publique dont le but ultime était de soulager l'embarras des certificats qui sont essentiels dans les systèmes de chiffrement à clé publique conventionnels pour garantir l'authenticité des clés publiques.

CHAPITRE « I »

Introduction à la cryptographie

1.1 Introduction

L'origine de la cryptologie mot réside dans la Grèce antique. La cryptologie est un mot composé de deux éléments : «cryptos », qui signifie caché et « logos » qui signifie mot. La cryptologie est aussi vieille que l'écriture elle-même, et a été utilisé depuis des milliers d'années pour assurer les communications militaires et diplomatiques. par exemple, le célèbre empereur romain Jule César utilisait un algorithme de chiffrement pour protéger les messages à ses troupes. Dans le domaine de l'un de cryptologie peut voir deux visions : la cryptographie et la cryptanalyse. Le cryptographe cherche des méthodes pour assurer la sûreté et la sécurité des conversations alors que le Crypto analyse tente de défaire le travail ancien en brisant ses systèmes. La cryptographie traditionnelle est l'étude des méthodes permettant de transmettre des données de manière confidentielle et la cryptanalyse, à l'inverse est l'étude des procédés cryptographiques, qui dépendent d'un paramètre appelé clé.

1.2. Définition

La cryptographie est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse. Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-d permettant de les rendre inintelligibles sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préférera le verbe chiffré. La cryptanalyse, à l'inverse, est l'étude des procédés cryptographiques dans le but de trouver des faiblesses, en particulier, de pouvoir décrypter des messages chiffrés. Le décryptement est l'action consistant à trouver le message en clair sans connaitre la clef de déchiffrement. La cryptologie, étymologiquement « la science du secret », ne peut être vraiment considérée comme une science que depuis peu de temps. Cette science englobe la cryptographie (l'écriture secrète) et la cryptanalyse (l'analyse de cette dernière)

1.3. Définition de la cryptographie

La cryptographie est l'art de chiffrer, coder les messages est devenue aujourd'hui une science à part entière. Au croisement des mathématiques, de l'informatique, et parfois même de la physique, elle permet ce dont les civilisations ont besoin depuis qu'elles existent : le maintien du secret. Pour éviter une guerre, protéger un peuple, il est parfois nécessaire de cacher des choses.

Figure I.1 : Schéma de cryptage

1.4. L'usage de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. Désormais, la cryptographie sert non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité.

La confidentialité : consiste à rendre l'information intelligible à d'autres personnes que les acteurs de la transaction.

L'intégrité : vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication.

L'authentification : consiste à assurer l'identité d'un utilisateur, c.-à-d. de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

Le non répudiation : de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

1.5. Mécanisme de la cryptographie

Un algorithme de cryptographie ou un chiffrement est une fonction mathématique utilisée lors du processus de cryptage et de décryptage. Cet algorithme est associé à une clé (un mot, un nombre ou une phrase), afin de crypter une donnée. Avec des clés différentes, le résultat du cryptage variera également. La sécurité des données cryptées repose entièrement sur deux éléments : l'invulnérabilité de l'algorithme de cryptographie et la confidentialité de la clé.

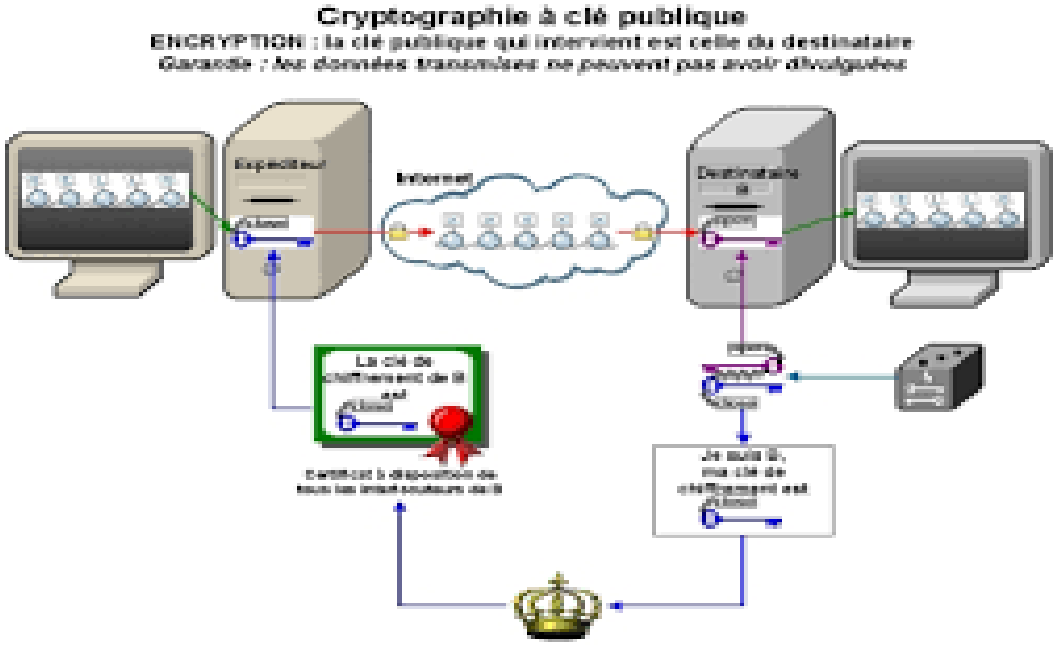

Figure 1.2 cryptographie à clé publique

On appelle clé une valeur utilisée dans un algorithme de cryptographie, afin de chiffrer une donnée. Il s'agit en fait d'un nombre complexe dont la taille se mesure en bits. On peut imaginer que la valeur correspondant à 1024 bits est absolument gigantesque.

Plus la clé est grande, plus elle contribue à élever la sécurité à la solution. Toutefois, c'est la combinaison d'algorithme complexe et de clés importantes qui seront la garantie d'une solution bien sécurisée. Les clés doivent être stockées de manière sécurisée et de manière à ce que seul leur propriétaire soit en mesure de les atteindre et de les utiliser.

1.6. Confidentialité et algorithmes de chiffrement

La confidentialité est le premier problème posé à la cryptographie. Il se résout par la notion de chiffrement. Il existe deux grandes familles d'algorithmes cryptographiques à base de clef :

Les algorithmes à clef secrète ou algorithmes symétriques, et les algorithmes à clef publique ou algorithmes asymétriques

Chiffrement symétrique ou clef secrète : dans la cryptographie conventionnelle, les clefs de chiffrement et de déchiffrement sont identiques : c'est la clef secrète, qui doit être connue des tiers communiquant et d'eux seuls. Le procédé de chiffrement est dit symétrique.

Les algorithmes symétriques sont de deux types :

• Les algorithmes de chiffrement en continu, qui agissent sur le message en clair un bit à la fois ;

• Les algorithmes de chiffrement par bloc, qui opèrent sur le message en clair par groupes de bits appelés blocs.

Les principaux algorithmes à clé privée sont :

-Blowfish

- DES/3DES

- IDEA

Chiffrement asymétrique ou à clef public : avec les algorithmes asymétriques, les clefs de chiffrement et de déchiffrement sont distinctes et ne peuvent se déduire l'une de l'autre. On peut donc rendre l'une des deux publique tandis que l'autre reste privée. C'est pourquoi on parle de chiffrement à clef publique. Si la clef publique sert au chiffrement, tout le monde peut chiffrer un message, que seul le propriétaire de clef privé pourra déchiffrer. On assure ainsi la confidentialité. Certains algorithmes permettent d'utiliser la clef privée pour chiffrer. Dans ce cas, n'importe qui pourra déchiffrer, mais seul le possesseur de clef privée peut chiffrer.

1.6.1. Les avantages et inconvénients de la cryptographie à clé publique comparé à la cryptographie à clé privée

Le premier avantage de la cryptographie à clé publique est d'améliorer la sécurité elle permet d'échanger des messages de manière sécurisée sans aucun dispositif de sécurité. L'expéditeur et le destinataire n'ont plus besoin de partager des clés secrètes via une voie de transmission sécurisée. Les communications impliquent uniquement l'utilisation de clés publiques et plus aucune clé privée n'est transmise ou partagée.

Avec un système à clé secrète, au contraire, il existe toujours le risque de voir la clé récupérée par une personne tierce quand elle est transmise d'un correspondant à l'autre. Toute personne interceptant la clé lors d'un transfert peut ensuite lire, modifier et falsifier toutes les informations cryptées ou authentifiées avec cette clé. De la norme de cryptage de données DES au code secret de Jules César, la distribution des clés reste le problème majeur du cryptage conventionnel. (Autrement dit, comment faire parvenir la clé à son destinataire sans qu'aucune personne ne l'intercepte ?) Les moyens à déployer pour garantir la distribution sécurisée des clés correspondants sont très onéreux, ce qui constitue un inconvénient supplémentaire.

Le cryptage à clé publique représente une révolution technologique qui offert à tout citoyen la possibilité d'utiliser une cryptographie robuste. En effet, la cryptographie conventionnelle était auparavant la seule méthode pour transmettre des informations secrètes. Les couts d'institutions disposants de moyens suffisants, telles que gouvernements et banque.

Un autre avantage majeur des systèmes à clé publique est qu'ils permettent l'authentification des messages par signature électronique, ce qui peut aussi servir devant un juge, par exemple.

L'inconvénient des systèmes à clé publique est leur vitesse contrairement aux méthodes à clé secrète qui sont plus rapide. Ils sont particulièrement adaptés à la transmission de grandes quantités de données. Mais les deux méthodes peuvent être combinées de manière à obtenir le meilleur de leurs systèmes. Pour le cryptage, la meilleure solution est d'utiliser un système à clé publique pour crypter une clé secrète qui sera alors utilisée pour crypter fichiers et message.

1.7. Différence entre chiffrement et codage

Les opérations de chiffrement et du codage font partie de la théorie de l'information. La différence essentielle réside dans la volonté de protéger les informations et d'empêcher des tierces personnes d'accéder aux données dans le cas du chiffrement. Le codage consiste à transformer de l'information (des données) vers un ensemble de mots. Chacun de ces mots est constitué de symboles. La compression est un codage : on transforme les données vers un ensemble de mots adéquats destinés à réduire la taille mais il n'y a pas de volonté de dissimuler (bien que cela se fasse implicitement en rendant plus difficile d'accès le contenu).

Le « code » dans le sens cryptographique du terme travaille au niveau de la sémantique (les mots ou les phrases). Par exemple, un code pourra remplacer le mot « avion » par un numéro. Le chiffrement travaille sur des composants plus élémentaires du message, les lettres ou les bits, sans s'intéresser à la signification du contenu. Un code nécessite une table de conversion, aussi appelée « dictionnaire » (code book en anglais). Ceci étant, « code » et « chiffrement » sont souvent employés de manière synonyme malgré cette différence

On peut aussi considérer que le chiffrement doit résister à un adversaire « intelligent » qui peut attaquer de plusieurs manières alors que le codage est destiné à une transmission sur un canal qui peut être potentiellement bruité. Ce bruit est un phénomène aléatoire qui n'a pas « d'intelligence » intrinsèque mais peut toutefois être décrit mathématiquement

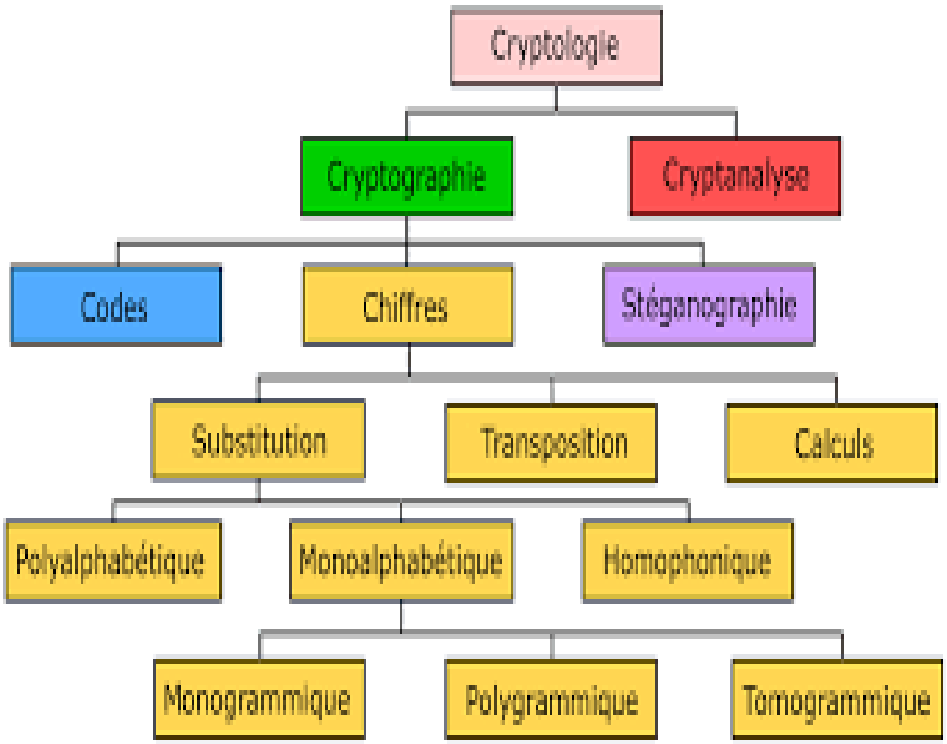

Figure 1.3 différence chiffrement et codage

1.8 Définition de la cryptanalyse

La cryptanalyse s'oppose en quelque sorte à la cryptographie, c'est l'étude des faiblesses des systèmes cryptographiques, elle est effectuée généralement par un intrus qui met en œuvre des méthodes afin de retrouver des informations secrètes tel que la clé, message en clair à partir d'informations considérées comme publique (cryptogramme, algorithmes), la cryptanalyse est une des disciplines de la cryptologie. Dans la cryptanalyse on part du principe que l'homme est faible et facilement soudoya le, ainsi la force d'un système doit reposer sur la force du principe utilisé. Si le but de la cryptographie est d'élaborer des méthodes de protection, le but de la cryptanalyse est au contraire de casser ces protections. Une tentative de cryptanalyse d'un système est appelé une attaque, et elle peut conduire à différents résultats :

Cassage complet : le cryptanalyse retrouve la clef de déchiffrement.

Obtention globale : le cryptanalyse trouve un algorithme équivalent à l'algorithme de déchiffrement, mais qui ne nécessite pas la connaissance de la clef de déchiffrement.

Obtention locale : le cryptanalyse retrouve le message en clair correspondant à un message chiffrer.

Obtention d'information : le cryptanalyse obtient quelque indication sur le message en clair ou la clef (certains bits de la clef, un renseignement sur la forme du message en clair).

D'une manière générale, on suppose toujours que le cryptanalyse connait le détail des algorithmes, fonctions mathématiques ou protocoles employés. Même si ce n'est pas toujours le cas en pratique, il serait risqué de se baser sur le secret des mécanismes utilisés pour assurer la sécurité d'un système, d'autant plus que l'usage grandissant de l'informatique rend de plus en plus facile la reconstitution de l'algorithme à partir du programme.

8.1 Les niveaux d'attaques

L'intrus peut effectuer quatre niveaux d'attaques, l'attaque est une tentative de cryptanalyse.

L'attaque par cryptogramme (par message chiffré seulement) : ou le cryptanalyse ne connait qu'un ensemble de message chiffrés, il peut soit retrouver seulement les messages en clair, soit retrouver la clef. En pratique, il est très souvent possible de deviner certaines propriétés du message en clair (format ASCII, présence d'un mot particulier, …), ce qui permet de valider ou non le décryptement.

L'attaque à message en clair connu : ou le cryptanalyse connait non seulement les messages chiffrés mais aussi les messages en clair correspondants, son but est alors de retrouver la clef. Du fait de la présence, dans la plupart des messages chiffrés, de parties connue (en-têtes de paquets, chants communs à tous les fichiers d'un type donné,…).

L'attaque à message en clair choisi : ou le cryptanalyse peut, de plus choisir des messages en clair à chiffrer et donc utiliser des messages apportant plus

d'informations sur la clef. Si le cryptanalyse peut de plus adapter ses choix en fonction des messages chiffrés précédents, on parle d'attaque adaptative.

L'attaque à message chiffré choisi : qui l'inverse de la précédente, le cryptanalyse peut choisir des messages chiffrés pour lesquels il connaitra le message en clair correspondant ; sa tâche est alors de retrouver la clef. Ce type d'attaques est principalement utilisé contre les systèmes à clef publique, pour retrouver la clef privée.

9. Conclusion

Un concepteur de système cryptographique est toujours en train d'essayer d'élaborer un système de chiffrement plus sure mais en même temps des intrus essayent de casser ce dernier, ils se livrent constamment une bataille mais les enjeux sont énorme.

CHAPITRE « II »

IBE : Identity-Based-Encryption

La cryptographie basée sur l'identité

Introduction :

La cryptographie basée sur l'identité Évidemment l'application la plus marquante des couplages est la réalisation de la cryptographie à clé publique basée sur l'identité (ID-PKC), le concept qui a été proposé par SHARNIR . Bien que l'idée de ID-PKC remonte à 1984, sa réalisation complète demeurait un problème ouvert durant plusieurs années. Dans son article original, Shamir donne un exemple concret d'un schéma de signature basé sur l'identité. Le chiffrement basé sur l'identité, par contre, s'est avéré une tâche beaucoup plus difficile. En 2001, Boneh et Franklin proposaient la première implémentation complète d'un système de chiffrement basé sur l'identité, en utilisant la bilinéarité des couplages.

1. Définitions

Dans cette partie, nous discutons de la cryptographie à clé publique en général, donnons la définition de la cryptographie basée sur l'identité, et définitions de certaines notions en lien avec la cryptographie basée sur l'identité.

1.1 La cryptographie à clé publique : dans la cryptographie à clé publique, chaque utilisateur U possède une paire de clés (Pu, Su) qui consiste en une clé publique et une clé secrète. La paire de clés est générée soit par l'utilisateur lui-même, ou par une certaine autorité centrale. Dans ce dernier cas, un canal sécuritaire est nécessaire pour acheminer cette paire de clés à l'utilisateur. La génération d'une paire de clés est effectuée, tout d'abord, en choisissant aléatoirement une clé secrète Su Puis on applique une fonction à sens unique à Su pour obtenir Pu. Ainsi, les autres usagers ne peuvent pas obtenir la clé secrète à partir de la clé publique Pu. Par exemple, dans le système de chiffrement ELGAMAL, la clé secrète **S** est choisie aléatoirement et la clé publique correspondante est définie par **h**(g), où g est un générateur du groupe publiquement connu. Calculer **S** à partir de **h** et g revient à calculer le logarithme discret.

Les autres participants utilisent la clé publique de l'utilisateur U afin de lui envoyer des messages chiffrés, ou bien pour vérifier ses signatures. Ceci peut être effectué par n'importe qui, étant donné que la clé **Pu** est une connaissance publique.

Cependant, le souci principal dans un schéma à clé publique est l'authenticité de la clé publique.

1.1 *La cryptographie basée sur l'identité* :

En 1984 Shamir a inventé le concept de la cryptographie basé sur l'identité qui s'adresse au problème de l'authenticité de différentes manières. Son idée originale avait pour objectif d'éviter le besoin pour l'authentification, en s'assurant que la valeur réelle de la clé publique d'un utilisateur est dérivée directement à partir d'une information disponible publiquement ce qui identifie uniquement et incontestablement cet utilisateur.

Cette information est désignée par une identité (digitale) d'un utilisateur. Dépendamment de l'application, l'identité peut être composée de (une ou une combinaison de) nom de l'utilisateur, numéro de l'assurance sociale, numéro de téléphone, adresse de courriel et possiblement d'autres informations personnelles. Dans ce cas, une clé publique d'un utilisateur est aisément disponible à n'importe qui connaissant son identité.

Ainsi, on n'a pas besoin de rechercher sa clé publique. D'ailleurs, le fait qu'il n'y a pas de doute sur l'authenticité de la clé publique évite le recours à des certificats. Cependant, la réalisation du lien entre les usagers et leurs identités digitales est loin d'être évident. Citons le cas où il y a plusieurs usagers qui possèdent le même nom. ... etc. Rappelons que clans les systèmes à clé publique conventionnels, la paire de clés est générée en choisissant aléatoirement une clé secrète et en appliquant une certaine fonction à sens unique pour calculer la clé publique.

Dans le cas de la cryptographie basée sur l'identité, la paire de clés est obtenue différemment. Tout d'abord, La clé publique est déterminée uniquement à partir de l'identité de l'usager, au lieu d'être calculée à partir d'une clé secrète. Ainsi, la génération de clés ne peut pas être effectuée par les usagers eux-mêmes. Précisément, si un usager sait comment générer la clé privée correspondante à sa clé publique (son identité), il pourrait aussi générer les clés privées des autres usagers.

Cependant, nous avons besoin d'un tiers appelé Générateur de Clés Privées (PKG) . Après une certaine procédure d'authentification (similaire à l'authentification à l'AC dans le cas de PKI), le PKG génère la clé privée de l'usager. Le PKG détient ce privilège (la génération des clés privées) car il possède une information secrète communément appelée la clé maître.

L'information correspondant à la clé maître et qui est disponible publiquement. Notons que ce schéma exige toujours un canal sécuritaire pour acheminer la clé privée à partir du PKG vers le bon usager.

La bilinéarité des couplages peut être utilisée pour transformer le système de chiffrement ELGAMAL en une variante du chiffrement basé sur l'identité.

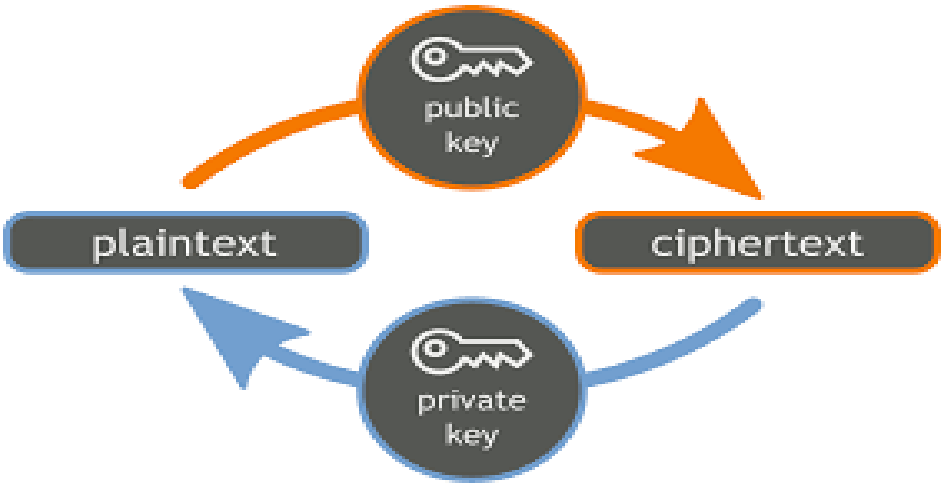

Figure 2.4 cryptographies basées sur l'identité

2 *La Sécurité des Systèmes de chiffrement basés sur l'Identité*

Dans un schéma de chiffrement à clé publique, il y a plusieurs concepts de sécurité. Les plus importants sont la sécurité sémantique et la sécurité de texte chiffré choisi. Cette dernière est une notion de sécurité suffisamment forte et acceptable pour la plupart des applications y compris les systèmes de chiffrement traditionnels (non basés sur l'identité). Les notions de la sécurité sémantique et la sécurité du texte chiffré choisi sont décrites ci-après.

2.1 *La Sécurité sémantique*

Un système de chiffrement à clé publique traditionnel est sémantiquement sécuritaire si un adversaire qui est donné un texte chiffré avec une clé publique connue n'apprend rien au sujet du texte clair correspondant. Plus particulièrement, considérons le jeu suivant entre un challengeur et un adversaire :

1. l'adversaire reçoit une clé publique aléatoirement choisie par le challengeur

2. l'adversaire produit cieux messages clairs de même taille M0 et M1 et reçoit, à partir du challengeur, le texte chiffré du message Mb, où b est choisi aléatoirement dans {O, 1},

3. l'adversaire produit (devine) b' et gagne le jeu si b b'. Le système à clé pub1icue est dit sémantiquement sécuritaire s'il n'existe pas d'adversaire capable de gagner ce jeu en temps polynomial avec une probabilité non-négligeable.

2.1 .1 La sécurité du texte chiffré choisi

Pour la définition de la sécurité du texte chiffré choisi, un jeu similaire est considéré, sauf que cette fois-ci, on attribue à l'adversaire une possibilité supplémentaire de pouvoir lancer au challenger un certain nombre de requêtes de déchiffrement. C'est-à-dire. L'adversaire présente au challenger un texte chiffré de son choix et le challenger lui renvoie le texte clair correspondant en utilisant la clé privée correspondante à la clé publique donnée à l'adversaire.

La seule restriction est que l'adversaire ne peut pas lancer une requête de chiffrement sur le texte chiffré (produit par le challenger) sur lequel on veut le tester. Un système de chiffrement à clé publique standard (non basé sur l'identité) est appelé sécuritaire contre une attaque à texte chiffré choisi s'il n'existe pas d'adversaire capable de gagner le jeu précédent avec une probabilité non-négligeable et en temps polynomial. La sécurité du texte chiffré choisi implique qu'étant donné un texte chiffré et une clé publique, un adversaire n'apprend rien au sujet du texte clair même s'il peut obtenir les textes clairs d'un certain nombre de textes chiffrés de son choix.

Ces notions de sécurité doivent être renforcées pour satisfaire les besoins des systèmes de chiffrement basés sur l'identité, où toute valeur (comme une chaîne de caractères) peut servir comme étant une clé publique.

En particulier, la clé publique n'est pas aléatoire. Ainsi, l'adversaire pourrait déjà avoir accès à un certain nombre de clés privées correspondant aux clés publiques de son choix.

En outre, un système basé sur l'identité devrait être sécuritaire contre une attaque sur une clé publique (une identité) choisie par l'adversaire au lieu d'être aléatoire.

2.2 La Sécurité sémantique d'un schéma basé sur l'identité

Considérons un adversaire et un challengeur dans un jeu similaire à l'IND-CPA précédent. Cependant, avec un schéma de chiffrement basé sur l'identité, l'adversaire peut choisir la clé publique (l'identité) sur laquelle il veut être testé, au lieu de recevoir une clé publique aléatoire à partir du challenger.

En outre, l'adversaire est autorisé à effectuer un certain nombre de requêtes d'extraction de clés privées. C'est-à-dire, l'adversaire présente une clé publique (une identité) de son choix, au challenger, et obtient la clé privée correspondante. La seule contrainte est que l'adversaire ne peut

pas lancer une requête d'extraction de clé privée sur laquelle on veut le tester. En particulier, cette identité ne devrait pas figurer dans la liste des requêtes d'extraction lancées précédemment.

Un schéma basé sur l'identité serait sémantiquement sécuritaire (IND ID-CPA) s'il n'existe pas d'adversaire pouvant gagner le jeu en temps polynomial et avec une probabilité non-négligeable plus que la moitié.

3 La Sécurité du texte chiffré choisi d'un schéma IBE :

Pour la sécurité du texte chiffré choisi dans un schéma de chiffrement basé sur l'identité, l'adversaire et le challenger jouent encore un jeu similaire à celui de IND ID-CPA précédent. L'adversaire peut choisir l'identité sur laquelle il veut être testé. En plus, l'adversaire peut lancer des requêtes de déchiffrement comme dans le jeu IND-CCA. Dans ce cas, chaque requête de déchiffrement consiste en une paire de texte chiffré choisi et d'une identité. En réponse à cette requête, l'adversaire recevra, à partir du challenger, le texte clair correspondant.

Évidemment, après le choix d'une identité et la réception d'un texte chiffré comme un challenge, un adversaire n'est plus autorisé à lancer une requête de déchiffrement sur cette combinaison en particulier.

3.1Comparaison avec la cryptographie à clé publique traditionnelle

La principale différence entre la cryptographie à clé publique basée sur l'identité et la cryptographie à clé publique traditionnelle réside dans la manière de générer une paire de clés (privée, publique), et les moyens de les vérifier.

Dans une PKI traditionnelle, ceci est réalisé par le biais de certificats. Tandis que dans un système de chiffrement basé sur l'identité, le lien entre l'identité et la clé privée est géré par l'autorité de confiance PKG, alors que l'authenticité Del 'identité peut être vérifiée publiquement.

Dans ce qui suit, nous comparons les systèmes de chiffrement basés sur l'identité aux systèmes à PlU traditionnels, en examinant les aspects suivants l'authenticité des paramètres (publiques) du système, l'enregistrement à une autorité, la révocation des clés, la distribution des clés, la sécurité de la clé maître, et enfin la délégation des clés privées**.**

3.2 L'authenticité des paramètres de système dans un système IBE

Nous supposons qu'un attaquant génère sa propre clé maître et les paramètres système correspondants, et trompe les usagers en leur faisant croire que les paramètres du système "forgés"

sont les vrais. Il s'en suit que, pour toute identité, il peut dériver la clé privée correspondant à sa clé maître. Cependant, il pourrait personnifier le PKG et générer des clés privées aux usagers en réponse à leurs requêtes. Dès lors, il peut déchiffrer tout message chiffré avec ces paramètres forgés. En conclusion, il est important que le PKG garantisse, d'une manière ou d'autre, l'authenticité des vrais paramètres du système. Un problème similaire peut se produire dans une situation de PlU, où les usagers ont besoin d'être sûrs de l'authenticité de la clé publique de l'autorité de certification. Précisément, si un attaquant peut faire croire aux usagers qu'une certaine clé publique de son choix est la clé publique de l'AC, il pourrait alors créer des certificats contenant les clés publiques forgés pour lesquelles il possède la clé secrète. Par conséquent, l'attaquant pourrait lire les messages chiffrés par les clés publiques forgées et créer des signatures qui ont l'air d'être valides. La seule différence avec le schéma IBE est que l'attaquant ne peut pas personnifier l'AC, car les usagers se rendront compte que leur certificat demandé contient une clé publique incorrecte.

3.3 Enregistrement à une autorité

Dans les deux systèmes, un usager qui veut participer a besoin de s'enregistrer à une autorité d'enregistrement, qui est souvent considérée comme étant une partie de 1'AC, respectivement le PKG.

Après l'opération d'authentification, l'autorité d'enregistrement choisit une identité numérique unique à l'usager, par exemple sous forme d'une adresse de courriel. Dans un système à PKI, un usager peut ainsi présenter son identité et une clé publique à l'AC avec une preuve de possession de la clé secrète correspondante.

L'AC publie ensuite un certificat contenant l'identité numérique et la clé publique. Similairement, dans un schéma IBE. L'usager présente son identité au PKG. L'autorité d'enregistrement est responsable de l'unicité de l'identité numérique et le lien entre l'identité et l'usager.

Le PKC génère ensuite la clé privée correspondante à cette identité. Les systèmes basés sur l'identité exigent un canal sécuritaire pour transmettre la clé privée du PKG à l'usager. Cependant, les systèmes IBE semblent fonctionner mieux dans les applications où il est facile de réaliser un canal sécuritaire, ou là où les usagers demandent moins souvent les clés privées

3.4 La révocation des clés

Dans un schéma à PKI, lorsqu'un certificat est révoqué, les autres usagers sont informés par le biais d'une liste de révocation de certificats (CRL)".

Par exemple cela se produit lorsqu'un usager quitte un groupe de travail ou qu'une clé privée est compromise. Dans ce dernier cas, lorsqu'une paire de clés a besoin d'être remplacée après l'expiration du certificat', un usager peut tout simplement générer une nouvelle paire de clés et obtenir un certificat.

Ainsi, la liste de révocation peut bien être utilisée clans un schéma IBE. Cependant, étant donné que la clé publique un usager est dérivée à partir de son identité, il ne peut pas changer d'identité à chaque fois qu'une nouvelle clé privée est exigée. Par exemple, lorsque l'année courante est ajoutée à l'identité, les usagers peuvent seulement utiliser leurs clés durant l'année en cours. Ainsi, les clés privées expirent annuellement et chaque usager devrait demander une autre clé privée chaque année. Que se passe-t-il si une clé privée vient d'être compromise ? Dans ce cas, l'usager devrait attendre jusqu'à la fin de l'année pour pouvoir avoir une autre clé privée. Cette situation pourrait être beaucoup plus rudimentaire en concaténant, par exemple, l'identité avec la date en cours au lieu de l'année en cours. Avec cette approche, la révocation de clés est plus simple ; lorsque un usager quitte la compagnie et sa clé privée a besoin d'être révoquée, la compagnie informe le PKG de ne plus générer de clés pour cet usager.

3.5 La distribution des clés

Incontestablement, le plus grand apport de la cryptographie basée sur l'identité est la simplification de la distribution des clés (publiques). À savoir, toutes les clés publiques peuvent être dérivées à partir de l'identité des usagers. Ainsi, l'obtention de la clé publique d'un autre usager, pour un chiffrement ou une vérification de signature, devient une procédure plus simple et plus transparente. En revanche, dans un schéma à PKI, on doit rechercher la signature de l'AC et la date d'expiration qui pourrait être accomplie en utilisant une CRL mise à jour.

3.6 La sécurité de la clé

Maître Dans un schéma IBE, le PKG représente le seul point de faiblesse. Un attaquant qui est capable de trouver la clé maître du PKG pourrait générer toutes les clés privées, et subséquemment, lire tous les messages chiffrés. Il est ainsi extrêmement important pour le PKG de garder secrètement la clé maître. Afin d'empêcher la sauvegarde de la clé maître dans un seul

endroit, une proposition été utiliser par le calcul multiparti, en se servant de la caractéristique de la bilinéarité des couplages, pour distribuer la clé maître sur plusieurs PKGs. Similairement, un attaquant qui peut obtenir la clé secrète d'une autorité de certification aurait la possibilité de créer des signatures au nom de l'AC pour les nouvelles clés publiques de son choix.

Cependant. La connaissance de la clé secrète de l'AC n'a aucun effet sur les messages chiffrés sous les clés publiques précédentes.

3.7 Délégation des clés

Supposons qu'un usager U qui joue en même temps le rôle d'un PKG, donc il a la possibilité d'extraire les clés privées de son choix, veut partir en voyage pour quelques jours, il installe sur son laptop les clés privées de ces jours-ci et non pas la clé maître. Si le laptop est volé, seulement les clés privées qui y sont installées sont compromises et non pas la clé maître qui reste secrète.

Conclusion

Cette partie supposant que dans une autre application, l'usager U peut déléguer d'autres usagers à utiliser la ligne d'objet d'un courriel comme étant l'identité servant au chiffrement.

Supposons que U a plusieurs employés sous sa responsabilité, chacun est responsable d'une tâche différente (achats, comptabilité, ...). U donne à chaque employé une clé privée correspondant à son identité qui est en lien avec sa tâche. De cette manière, U est sûr que chaque employé peut seulement déchiffrer le courriel qui lui .

CHAPITRE III

Ciphertext-Policy Attribute Based ENCRYPTION (CP-ABE)

1 – Introduction

Ciphertext-Policy Attribut-Based Encryption (CP-ABE) est une technique cryptographique très puissante, elle permet d'implémenter du contrôle d'accès basé sur des rôles (attributs)

2 – C'est quoi CP-ABE ?

CP-ABE pour « Ciphertext Policy Attribut-Based Encryption », est une technique cryptographique qui permet d'implémenter le contrôle d'accès basé sur les rôles. Les rôles sont appelés ici des attributs. La figure 5 montre les phases d'initialisation et de génération de clés. Elles concernent toutes les deux l'autorité des attributs. Cette dernière initialise les paramètres du système en générant les clés maîtresses publiques et secrètes. Ces mêmes clés sont utilisées pour générer les clés utilisateurs en utilisant la liste des attributs de chaque utilisateur.

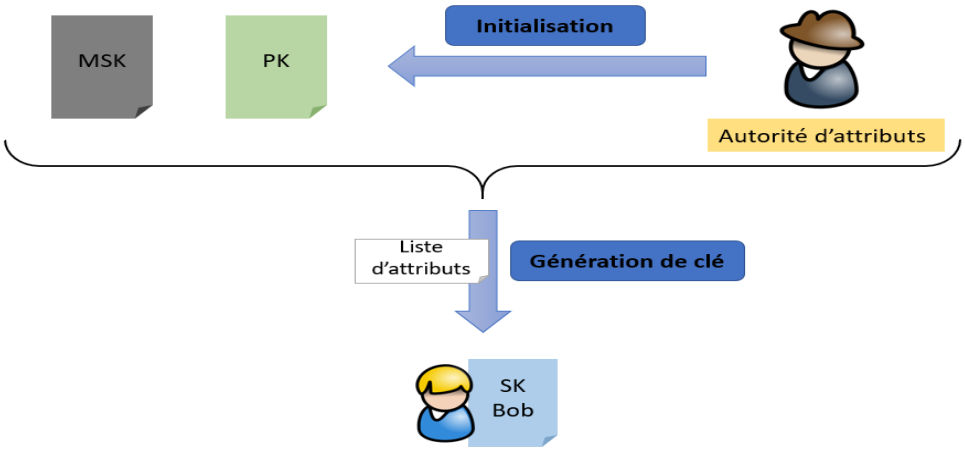

Figure 3.5: Phases d'initialisation et de génération de clés

La figure 3.6 quant à elle, illustre les phases de chiffrement de déchiffrement. Pour chiffrer un fichier, un utilisateur a besoin de la clé maîtresse publique, le fichier en question, et un politique d'accès (Elle est montrée sous forme d'arbre logique sur la figure). En suite, vient l'étape de déchiffrement, où l'utilisateur « bob » utilise sa propre clé secrète, qui contient des attributs satisfaisant la politique d'accès, pour retrouver le contenu du fichier original.

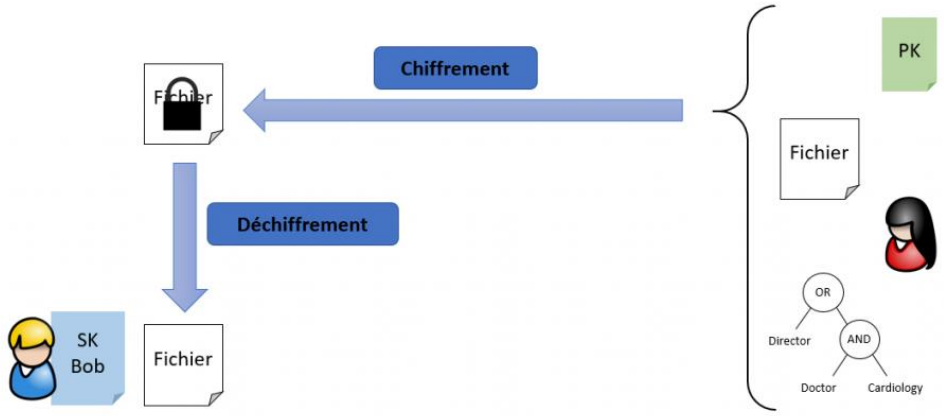

Figure 3.6 : Phases de chiffrement et déchiffrement

2.1 L'idée Dans le protocole "CP-ABE"

Le message est chiffré avec une structure d'accès qui définit quels sont les utilisateurs pouvant accéder au message clair. Les attributs que possède un utilisateur servent à définir sa clé privée et si ces attributs passent la structure d'accès, le message clair est obtenu. La principale propriété du protocole "CP-ABE" est sa résistance contre les collusions, ainsi deux ou plusieurs utilisateurs ne peuvent, en unissant leur clé privée, passer la structure d'accès si, individuellement, aucun d'eux ne possède les attributs pour les passer.

2.2 Les étapes Le protocole “CP-ABE”

Est composé de cinq algorithmes fondamentaux : Initialisation, Chiffrement, Génération de clé, Délégation 1 et Déchiffrage. La première étape du protocole est donc l’initialisation au cours de laquelle la clef publique PK et la clef secrète MK sont produites.

2.3-Structure d’accès associé à ABE

La structure d’accès prend généralement la forme d’un arbre d’accès.

La Figure 3.1 illustre un arbre d’accès pour un exemple de politique d’accès simple :

P = Att1 ∨ ((Att2 ∧ Att3) ∨ (Att4 ∧ Att5)).

Chaque feuille représente un attribut et chaque nœud interne est une porte logique (∧ , ∨). Sécurité des données dans un environnement Internet des Objets .

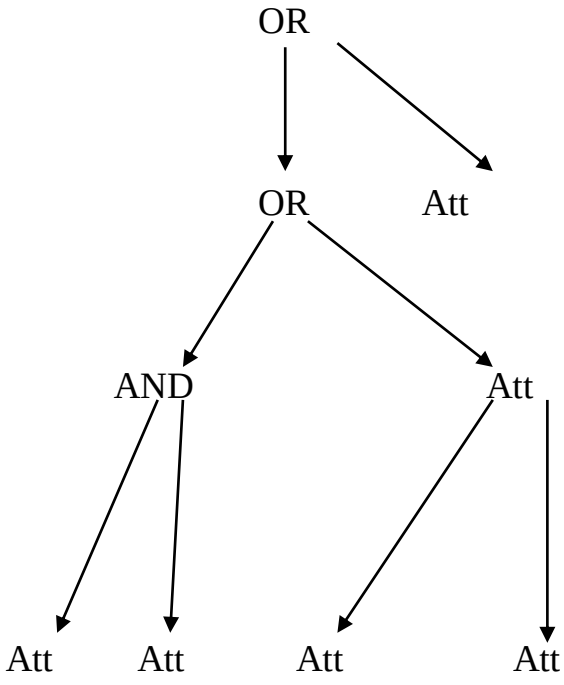

Figure 3.7 – Arbre d’accès pour un exemple de politique simple

2.4 Contrôle d'accès CP-ABE (Ciphertext-Policy Attribut Based Encryption)

Proposée pour la première fois par Béthencourt et al. en 2007, où chaque utilisateur est associé à un ensemble de traits. Sa clé secrète est créée en fonction de ses attributs. Lors du cryptage d'un message, le codeur définit la structure d'accès de seuil pour ses attributs intéressés.

Ce message est ensuite chiffré sur la base de cette architecture d'accès afin que seuls ceux dont les attributs correspondent à l'architecture d'accès puissent le déchiffrer. Grâce à la technologie CP-ABE, les données cryptées peuvent rester confidentielles et protégées contre les attaques de collusion. CP-ABE applique la politique d'accès directement sur les données : la clé de chaque utilisateur est associée à un ensemble d'attributs, et un utilisateur peut déchiffrer un texte chiffré si ses attributs satisfont la politique d'accès définie sur les données

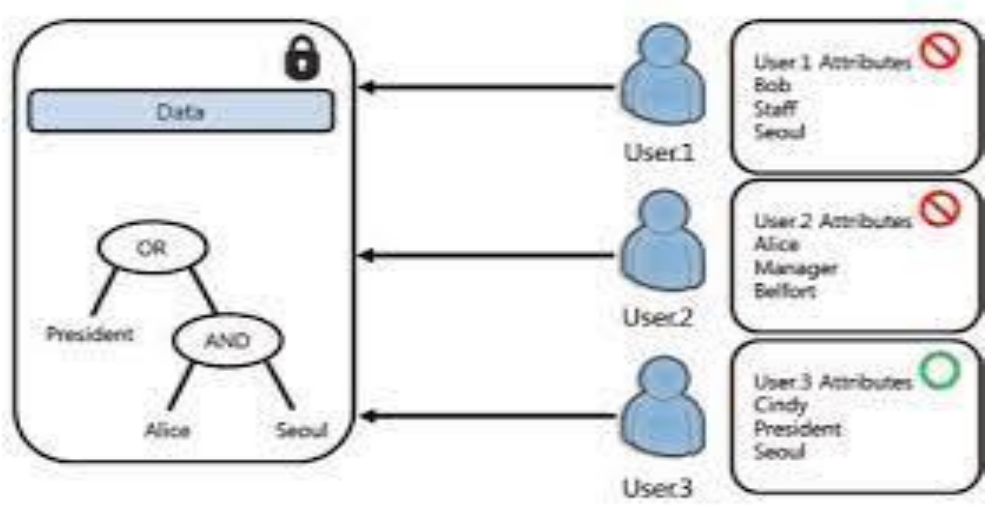

Figure 3.8_CP-ABE (Ciphertext-Policy Attribute Based Encryption)

3 -Chiffrement KP-ABE (Key-Policy Attribute Based Encryption)

Chiffrement "KP-ABE" dans lequel la politique d'accès est intégrée dans la clé secrète, c'est-à-dire que nous décidons pour chaque utilisateur à quels objets il peut

accéder. Nous attachons à chaque texte chiffré un ensemble de fonctionnalités. Une clé secrète spécifique peut être, avec une politique Accès spécifique, déchiquetant uniquement le texte chiffré avec des attributs qui répondent à sa politique d'accès.

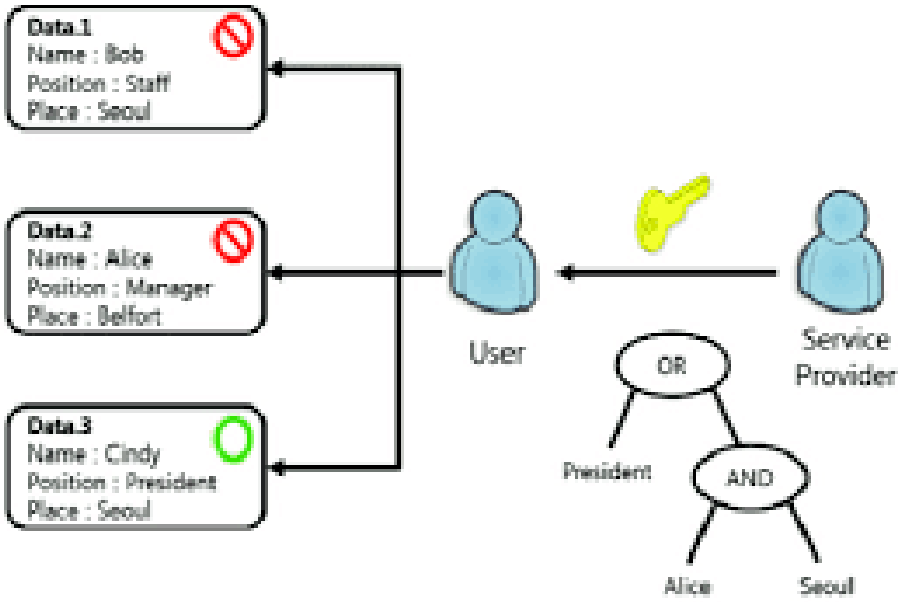

Figure 3.9_ KP-ABE (Key-Policy Attribut Based Encryption)

4- Comparaison entre KP-ABE et CP-ABE

Le nombre d'attributs, en particulier, joue un rôle fondamental dans les performances ABE : le chiffrement dans CP-ABE nécessite de calculer deux exponentiations pour chaque attribut dans la politique d'accès résultante.

De même, le cryptage KP-ABE nécessite deux exponentiations pour chaque attribut appliqué sur le texte chiffré. La complexité de déchiffrement dans CP-ABE est limitée par **L** exponentiations et **2L**'opérations d'appariement [10] , tandis que dans KP-ABE par seulement **l**'opération d'appariement ;

1 est le nombre d'attributs « correspondant » à la politique d'accès (dans CP-ABE) ou à la politique de clé (dans KP-ABE).

À titre d'exemple, il peut y avoir des situations où l'accès aux données doit être limité à un seul modèle de périphériques, libéré après une certaine date (qui peut être représentée sous la forme d'un entier de 64 bits).

La principale déférence est que dans CP-ABE, la politique d'accès est incluse dans le chiffré et les attributs sont inclus dans la clé de déchiffrement, alors que dans KP-ABE, c'est exactement l'inverse.

5- Conclusion

Dans ce chapitre, nous avons présenté un aperçu général sur la sécurité des données personnelles dans l'environnement IOT et le Cloud Nous avons, dans un premier temps, discuté les solutions cryptographiques d'une manière générale et ensuite le la solution chiffrement ABE

CHAPITRE IV

Etude et Analyse

Introduction

Dans ce chapitre nous allons implémenter notre system et présenter quelques notions sur le l'environnement docker, définitions, chiffrement de donnes ainsi que les technologies utilisées pour le développement.

Pour quoi docker ?

D'abord qu'est-ce que docker ?

Docker permet d'embarquer une application dans un ou plusieurs containers logiciels qui pourra s'exécuter sur n'importe quel serveur machine, qu'il soit physique ou virtuel. Docker fonctionne sous Linux comme Windows Server. C'est une technologie qui a pour but de faciliter les déploiements d'application, et la gestion du dimensionnement de l'infrastructure sous-jacente. Elle est proposée par la société Docker, en partie en open source

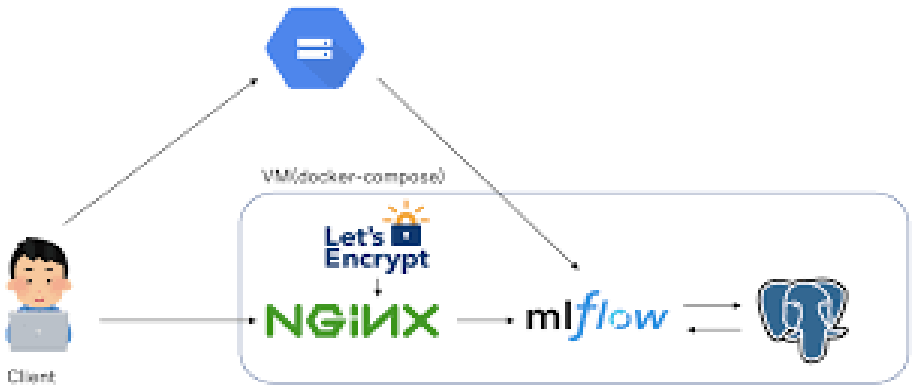

Figure 4.10 : interface DOCKER

Quelle différence avec la virtualisation ? La virtualisation permet, via un hyperviseur, de simuler une ou plusieurs machines physiques, et les exécuter sous forme de machines virtuelles (VM) sur un serveur ou un terminal. Ces VM intègrent elles-mêmes un OS sur lequel des applications sont exécutées. Ce n'est pas le cas du container logiciel. Le container fait en effet directement appel à l'OS de sa machine hôte pour réaliser ses appels système et exécuter ses applications, d'où son extrême légèreté. Dans le cas de Linux, les containers Docker exploitent un composant du noyau Linux baptisé LXC (ou Linux Container). Au format Windows Server, ils s'adossent à une brique équivalente, appelée Windows Server Container. Le moteur Docker normalise ces briques par le biais d'API dans l'optique d'exécuter les applications dans des containers standards, qui sont ensuite portables d'un serveur à l'autre.

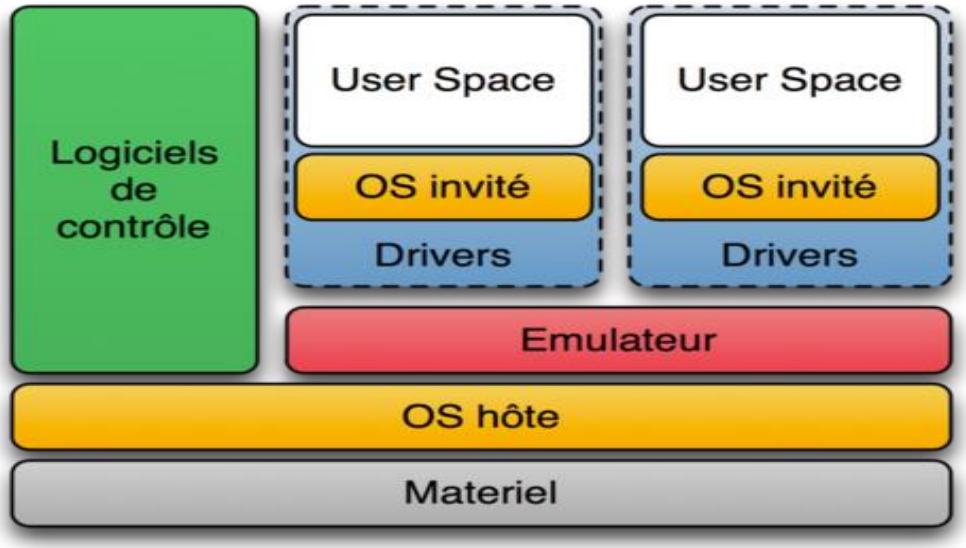

Figure 4.11 : la différence entre la virtualisation et la conteneurisation

Comment installer docker sur Windows ?

Docker propose une version pour poste de travail. Elle fait l'objet d'une déclinaison open source, baptisée le Docker Desktop, dessinée pour manipuler des containers et tester des architectures containérisées. Docker Desktop fait l'objet d'une version Enterprise, payante. Elle permet d'automatiser le déploiement d'environnements de développement Docker sur des dizaines de milliers de postes, puis d'assurer la gestion de leur maintenance et leur cohérence au regard des applications en production (lire l'article : Docker industrialise les développements Docker sur le poste de travail).

Quels sont les avantages du docker ?

Comme le container n'embarque pas d'OS à la différence de la machine virtuelle, il est plus léger que cette dernière. Il n'a pas besoin d'activer un second système d'exploitation pour exécuter ses applications. Cela se traduit par un lancement beaucoup plus rapide. Pour parvenir au même résultat, les environnements de virtualisation ont besoin d'un pool de VM inactives provisionnées à l'avance. Avec Docker, nul besoin de pool, puisqu'un container est bootable en quelques secondes. IBM a publié en 2014 un comparatif de performance entre Docker et KVM. Sa conclusion est sans appel : Docker égale ou excède les performances de l'hyperviseur - et ce dans tous les cas testés. Pour Big Blue, la vélocité des containers Docker se rapproche même de celle des serveurs machines nus. En éliminant la couche de virtualisation, consommatrice en ressources, Docker permettrait de réduire la consommation de RAM de 4 à 30 fois. Publiée en août 2017 par le département de

recherche informatique de l'université de Lund en Suède, une autre étude compare cette fois la performance des containers à celle des machines virtuelles VMware. Elle aboutit également à une conclusion en faveur de Docker.

Les containers Docker, et les applications qu'ils embarquent, sont portables de cloud en cloud. Seule condition : que les cloud en présence soient optimisés pour les accueillir. C'est le cas des principaux d'entre eux : Amazon Web Services, Microsoft Azure, Google Cloud Platform, OVH... Conclusion : il devient possible d'imaginer réaliser du mirroring d'application et de l'équilibrage de charge entre cloud, et pourquoi pas des plans de reprise ou continuité d'activité entre cloud... voire même décider de la reprise d'un projet sur un cloud tiers.

Côté déploiement, Docker présente un intérêt car il permet de limiter les mises à jour au delta du container nécessitant de l'être. Enfin en termes d'urbanisation applicative, Docker en isolant les composants fonctionnels d'une application dans des containers permet de les maintenir indépendamment tous en leur allouant à chacun les ressources machines voulues. C'est le concept d'architecture de microservices.

Docker vers virtuelle machine

En ce qui concerne l'infrastructure cloud, la machine virtuelle a été la norme de référence pour nombre de ses avantages. Cependant, que se passerait-il si vous aviez une alternative à une machine virtuelle plus légère, économique et évolutive. C'est précisément ce que Docker est.

CP-ABE

Docker est une technologie basée sur des conteneurs qui vous permet de développer des applications distribuées. Dans cet article de blog, j'expliquerai les différences entre les machines virtuelles et les conteneurs Docker.

C’est quoi une machine virtuelle ?

Une machine virtuelle est un système qui agit exactement comme un ordinateur.

En termes simples, il permet d'exécuter ce qui semble être sur de nombreux ordinateurs séparés sur du matériel, c'est-à-dire un ordinateur. Chaque machine virtuelle nécessite son système d'exploitation sous-jacent, puis le matériel est virtualisé.

Docker vs. VM

Maintenant voila les différences significatives entre les conteneurs Docker et les machines virtuelles. Eh bien, les différences significatives sont la prise en charge, la sécurité, la portabilité et les performances du système d'exploitation.

Alors discutons chacun de ces termes un par un.

Support du système d'exploitation

La prise en charge du système d'exploitation de la machine virtuelle et du conteneur Docker est très différente. Dans l'image ci-dessus, vous pouvez voir que chaque machine virtuelle a son système d'exploitation invité au-dessus du système d'exploitation hôte, ce qui alourdit les machines virtuelles. Alors que d'un autre côté, les conteneurs Docker partagent le système d'exploitation hôte, et c'est pourquoi ils sont légers.

Le partage du système d'exploitation hôte entre les conteneurs les rend très légers et les aide à démarrer en quelques secondes. Par conséquent, la surcharge de gestion du système de conteneurs est très faible par rapport à celle des machines virtuelles.

Les conteneurs Docker conviennent aux situations où vous souhaitez exécuter plusieurs applications sur un seul noyau de système d'exploitation. Mais si vous avez des applications ou des serveurs qui doivent s'exécuter sur différentes versions de système d'exploitation, des machines virtuelles sont nécessaires.

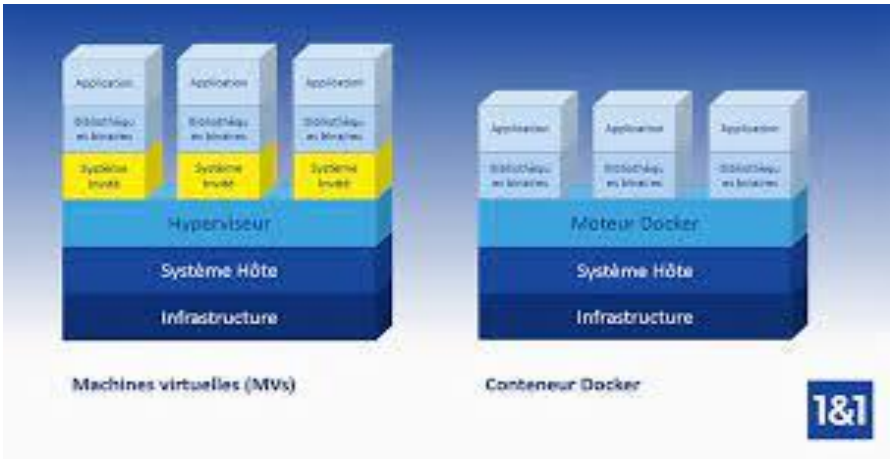

Figure 4.12 Support du S.E

Sécurité

La machine virtuelle ne partage pas le système d'exploitation et il existe une forte isolation dans le noyau hôte. Par conséquent, ils sont plus sûrs que les conteneurs. Un conteneur présente de nombreux risques de sécurité et des vulnérabilités car les conteneurs ont un noyau hôte partagé.

De plus, étant donné que les ressources du docker sont partagées et non espacées de noms, un attaquant peut exploiter tous les conteneurs d'un cluster s'il a accès ne serait-ce qu'à un seul conteneur. Dans une machine virtuelle, vous n'obtenez pas d'accès direct aux ressources, et l'hyperviseur est là pour restreindre l'utilisation des ressources dans une machine virtuelle.

Portabilité

Les conteneurs Docker sont facilement portables car ils ne disposent pas de systèmes d'exploitation séparés. Un conteneur peut être porté sur un autre système d'exploitation et il peut démarrer immédiatement. D'un autre côté, les machines virtuelles ont un système d'exploitation distinct, de sorte que le portage d'une machine virtuelle est difficile par rapport aux conteneurs, et le portage d'une machine virtuelle prend également beaucoup de temps en raison de sa taille. À des fins de développement où les applications doivent être développées et testées sur différentes plates-formes, les conteneurs Docker sont le choix idéal.

Performance

Comparer des machines virtuelles et des conteneurs Docker ne serait pas juste car ils sont tous deux utilisés à des fins différentes. Mais l'architecture légère de docker son fonctionnalité moins gourmande en ressources en fait un meilleur choix qu'une machine virtuelle. En conséquence, quels conteneurs peuvent démarrer très rapidement par rapport à ceux des machines virtuelles, et l'utilisation des ressources varie en fonction de la charge ou du trafic qu'elles contiennent.Contrairement au cas des machines virtuelles, il n'est pas nécessaire d'allouer des ressources de manière permanente aux conteneurs. La mise à l'échelle et la duplication des conteneurs est également une tâche facile par rapport à celle des machines virtuelles, car il n'est pas nécessaire d'y installer un système d'exploitation.

Comparaison entre une virtuelle machine et conteneur docker

Machine virtuelle	Conteneur Docker
Isolation des processus au niveau matériel	Isolation des processus au niveau du système d'exploitation
Chaque machine virtuelle a un système d'exploitation distinct	Chaque conteneur peut partager le système d'exploitation
Bottes en quelques minutes	Démarre en quelques secondes
Les machines virtuelles sont de quelques Go	Les conteneurs sont légers (Ko / Mo)
Les VM prêtes à l'emploi sont difficiles à trouver	Les conteneurs Docker pré-construits sont facilement disponibles
Les VM peuvent facilement se déplacer vers un nouvel hôte	Les conteneurs sont détruits et recréés au lieu de se déplacer
La création de VM prend un temps relativement plus long	Les conteneurs peuvent être créés en quelques secondes
Plus d'utilisation des ressources	Moins d'utilisation des ressources

Tableau : différence Vm / Cd

Le système d'exploitation Ubuntu

« Ubuntu » est un système d'exploitation GNU/Linux basé sur Debian, développé, commercialisé et maintenu pour les ordinateurs individuels (desktop), les serveurs (Server) et les objets connectés (Core) par la société Canonical.

Ubuntu est disponible en deux versions, une qui évolue tous les six mois, et une version LTS, pour Long Term Support (« Support long terme ») qui évolue tous les deux ans. Ubuntu se définit comme « un système d'exploitation utilisé par des millions de PC à travers le monde » et avec une interface « simple, intuitive, et sécurisée ». Elle est la distribution la plus utilisée pour accéder aux

sites web d'après le site Alexa, et le système d'exploitation le plus utilisé pour les serveurs informatiques

Caractéristiques

Ubuntu est proposé gratuitement sous forme d'image disque permettant de créer un live CD ou live USB afin de tester Ubuntu sans installation et sans modifier le système utilisé. L'interface utilisateur est disponible en 146 langues nationales ou régionales (mars 2016). Une nouvelle version sort tous les six mois avec un support de 9 mois. Tous les deux ans (en avril des années paires), la version est une version avec support à long terme dit LTS (long term support) qui est maintenue pendant cinq ans. Sur un SSD l'installation ne dure qu'une dizaine de minutes.

Il inclut un ensemble de logiciels pré-installés (LibreOffice, Firefox, etc.) et sa logithèque permet d'installer toute une gamme de logiciels correspondant aux besoins de chaque utilisateur.

Architecture

Ubuntu est disponible, officiellement pour les architectures x86 (Intel et compatibles), AMD64 et SPARC mais non officiellement pour PowerPC (arrêt du support avec la version 6.10), IA-64 (Intel Itanium) et PlayStation 3 (processeur Cell dérivé du PowerPC d'IBM). À partir de la version 20.4, Ubuntu n'est disponible qu'avec les architectures x64 et AMD64.

Composants

Ubuntu divise tous les logiciels en quatre sections, appelées « composants » ou encore « dépôts », en les classant par licences et par niveau de support utilisateur disponible. Par défaut, une sélection de paquets de *main* (principal) sera installée pour couvrir les besoins élémentaires de la majorité des utilisateurs.

- main (principal) : paquets sous licence considérée comme open source, et pour lesquels le support de Canonical est disponible.
- restricted (restreint) : un ensemble de pilotes et de firmwares dont la licence n'est pas considérée comme libre.
- universe (univers) : un grand choix de open source (développement communautaire sans l'intervention de Canonical).
- multiverse (multivers) : un ensemble de open source soumis à des restrictions légales (non supporté par Canonical)

Il existe cependant d'autres composants secondaires :

- *security* : mises à jour de sécurité ;
- *updates* : corrections de bogues importantes ;
- *backport* : dernières mises à jour issues de la version en développement ;
- *commercial* : applications réalisées par des entreprises partenaires de Canonical.

Un autre composant, *medibuntu* (« Multimedia, Entertainment and Distractions in Ubuntu », anciennement PLF) qui comprenait des paquets légaux en France, mais litigieux dans d'autres pays du fait de l'existence dans certains pays de brevets logiciels était installable. Il a été fermé définitivement fin 2013 avec l'arrivée d'Ubuntu 13.10, après ne plus avoir été maintenu depuis plusieurs mois. À l'exception du support de la bibliothèque *libdvdcss* repris par VideoLAN, aucun remplaçant n'a été développé depuis, les logiciels étant soit présent dans les dépôts officiels, soit ayant des équivalents dans ces mêmes dépôts

Figure 4.13 : Interface IBUNTU

Création d'une nouvelle machine Virtuelle sous le nom UBUNTO

Installation du Ubuntu

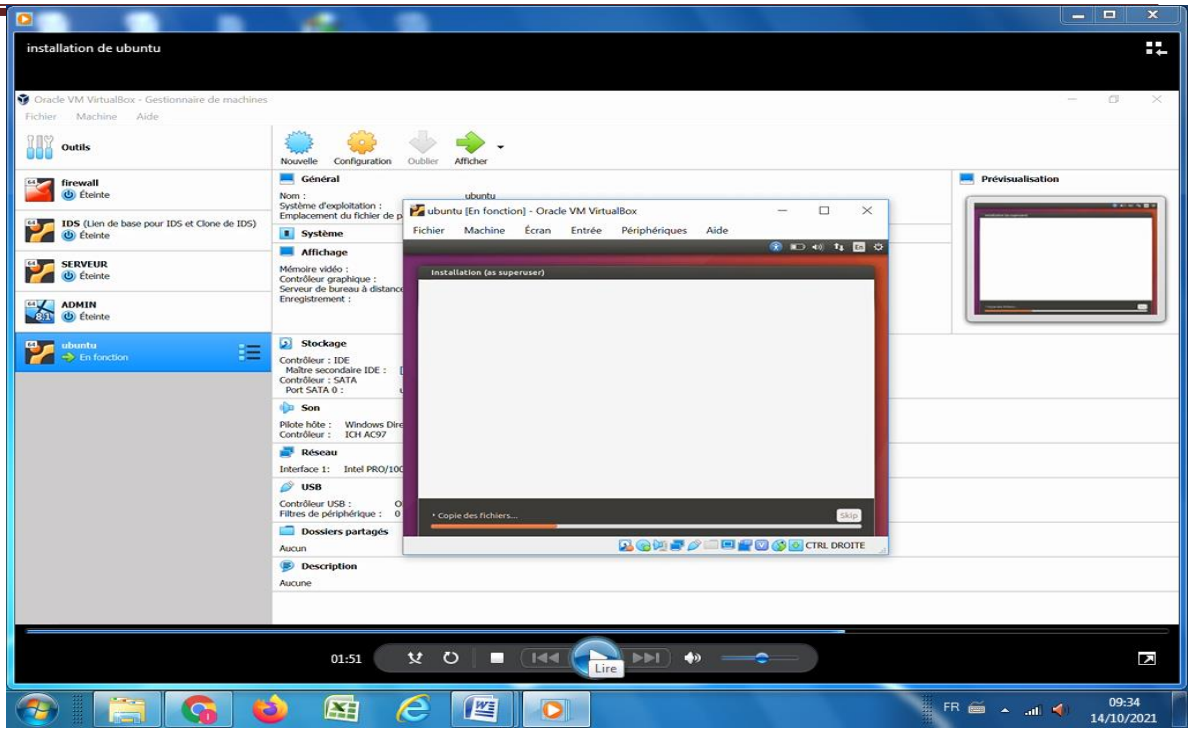

Figure 4.14 : Installation du Ubuntu

L'installation du docker

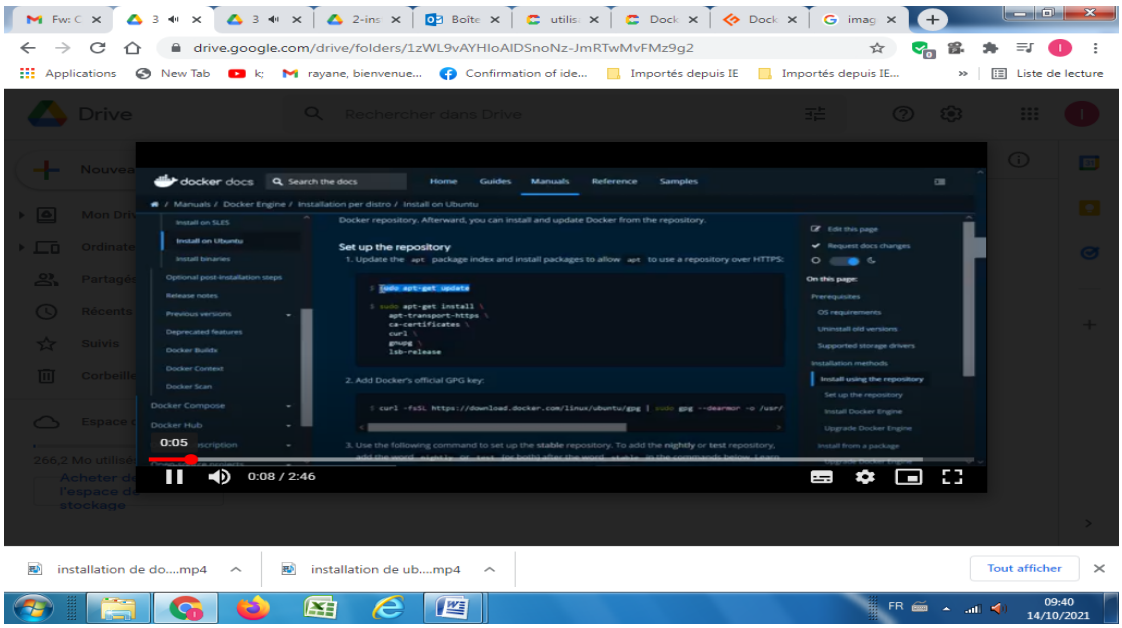

Figure 4.15 :L'installation du docker

Installation du Docker Hub

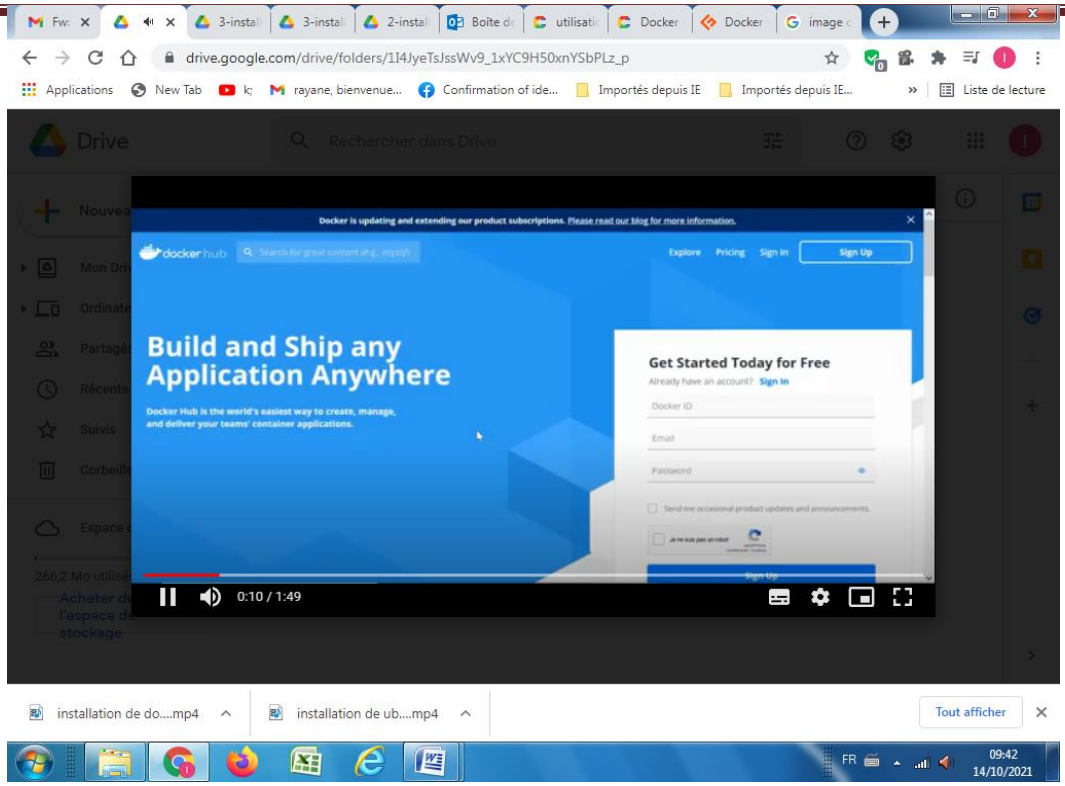

Figure 4.16 :L'installation du Docker Hub

Chargement de l'image OpenABE

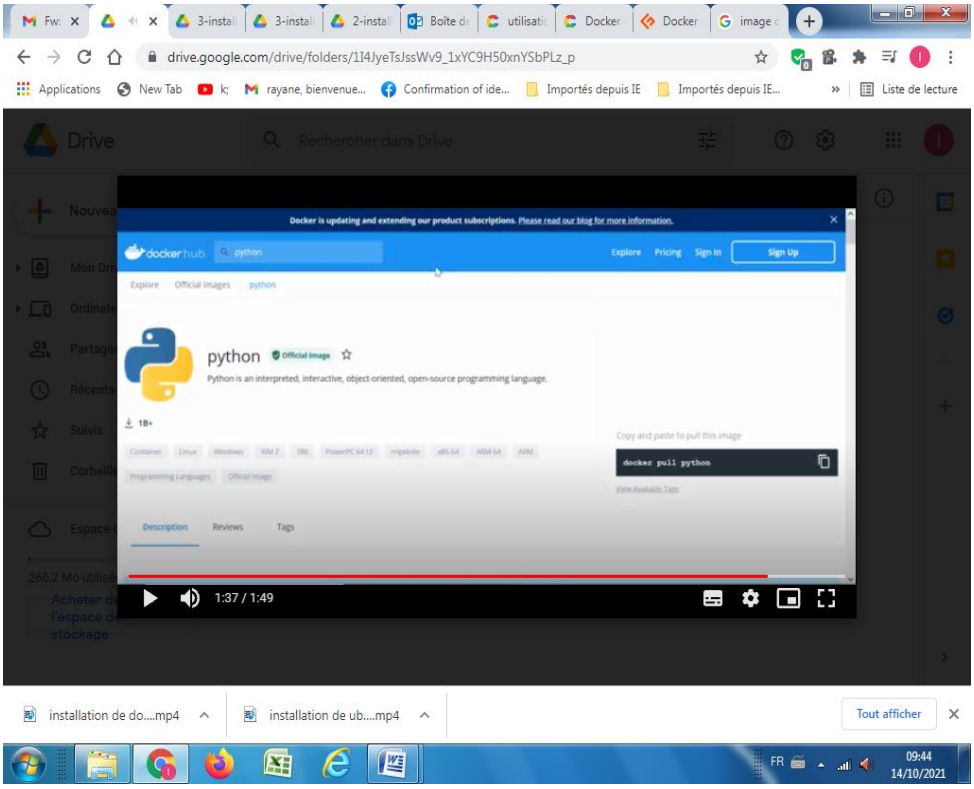

Figure 4.17 : Chargement de l'image OpenABE

L'explication du CP-ABE

Comme on vu dans le dernier chapitre Policy Attibute-BasedEncryption (CP-ABE) est une technique cryptographique très puissante, elle permet d'implémenter du contrôle d'accès basé sur des rôles (RBAC). Dans cet article, on va présenter le fonctionnement le CP-ABE en général, puis, on va utiliser la librairie open ABE (Une bibliothèque qui implémente CP-ABE) pour tester CP-ABE.

Vu la complexité et Afin d'éviter d'installer OpenABE, on va utiliser une image docker toute prête pour ça.

2 – C'est quoi CP-ABE ?

CP-ABE pour « Ciphertext Policy Attibute-BasedEncryption », est une technique cryptographique qui permet d'implémenter le contrôle d'accès basé sur les rôles. Les rôles sont appelés ici des attributs.

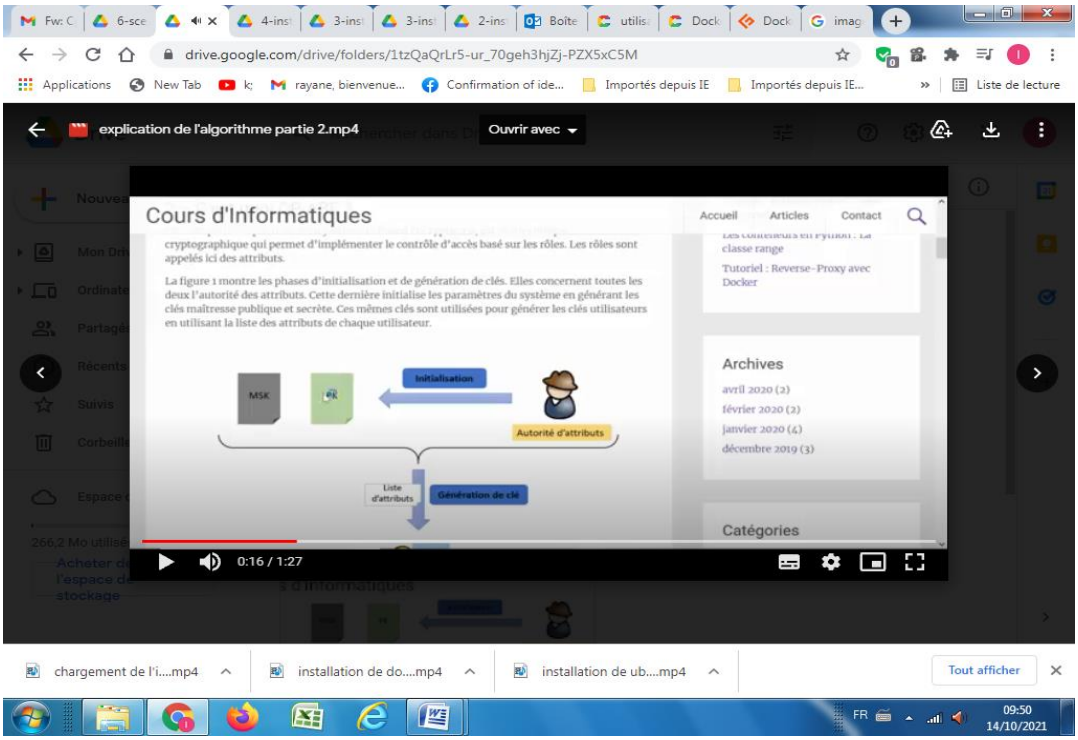

Figure 4.18 : *L'explication du CP-ABE*

Le Scenario du cryptographie des fichiers

Après l'initialisation des clés publiques et privés en va a la section des génération des clés ou chaque acteurs va avoir son clés privés

Acteurs : Mohamed,Sara ,Hahar et Ines

Travail a faire crypté deux fichiers

1-calendrier de la section

2-calendrier des groupes

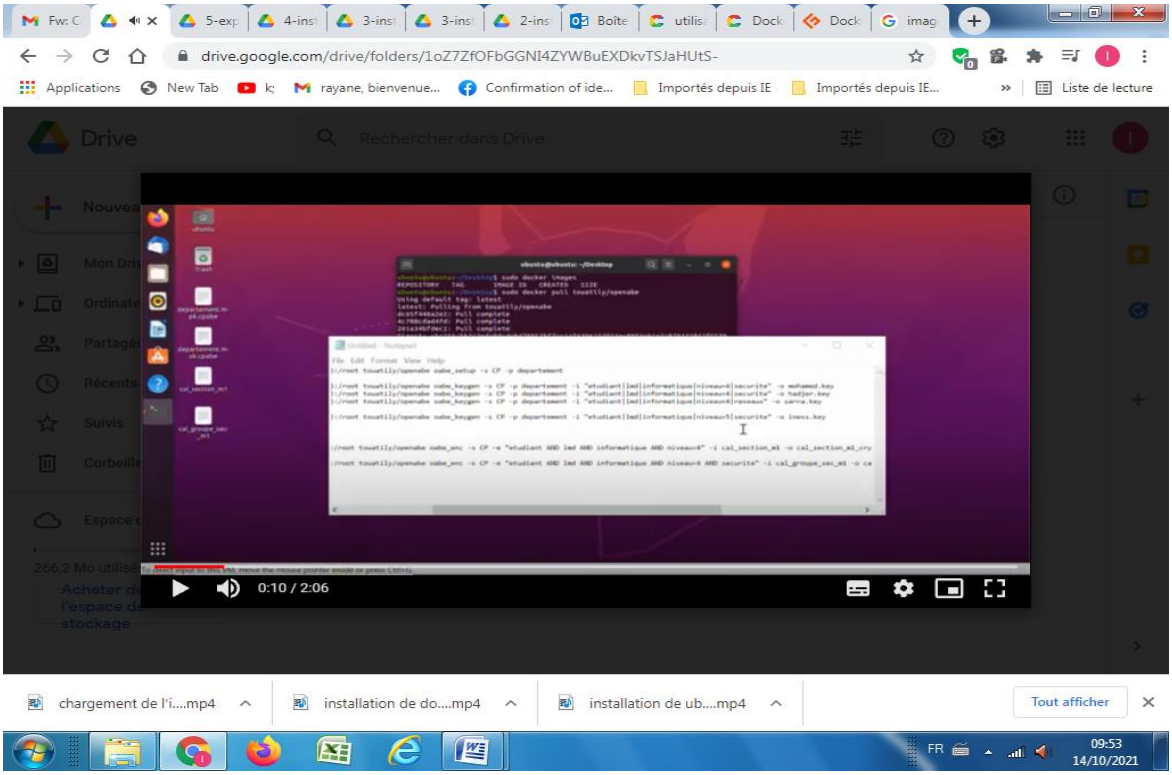

Figure 4.19 : *Le Scenario de la cryptographie des fichiers*

« Phase I initialisation des clés »

Après l'installation de l'image on va faire des conteneurs pour voir les différentes phases de l'algorithme CP-ABE

« sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_setup -s CP -p département »

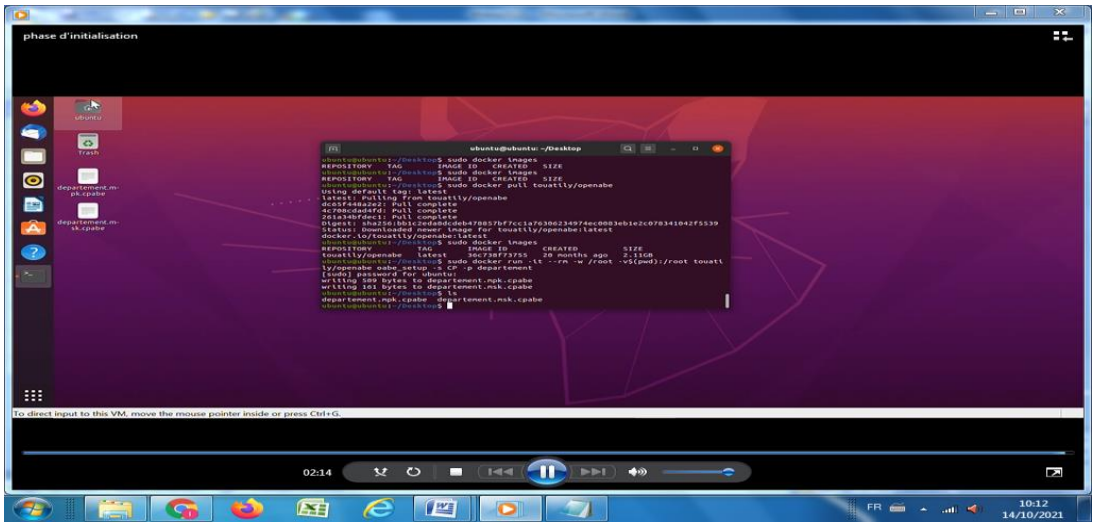

Figure 4.20 : *initialisation des clé*

« Phase II génération des clés » :

Etudiant 1 sudo docker run -it --rm -w /root -v$(pwd) :/root touatily/openabe oabe_keygen -s CP -p départements -i "etudiant|lmd|informatique|niveau=4|securite" -o mohamed.key

Etudiant 2sudo docker run -it --rm -w /root -v$(pwd) :/root touatily/openabe oabe_keygen -s CP -p départements -i "etudiant|lmd|informatique|niveau=4|securite" -o hadjer.key

Etudiant 3sudo docker run -it --rm -w /root -v$(pwd) :/root touatily/openabe oabe_keygen -s CP -p départements -i "etudiant|lmd|informatique|niveau=4|reseaux" -o sarra.key

Etudiant 4sudo docker run -it --rm -w /root -v$(pwd) :/root touatily/openabe oabe_keygen -s CP -p departement -i "etudiant|lmd|informatique|niveau=5|securite" -o iness.key

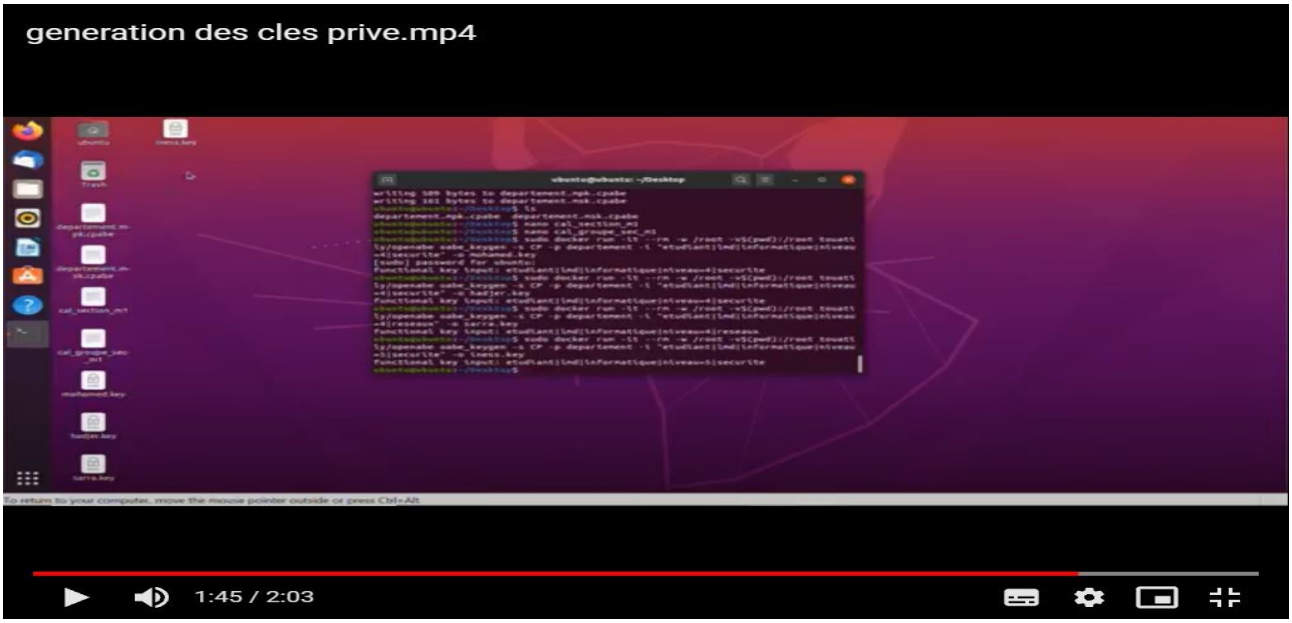

Figure 4.21 : génération des clés

« Phase III : le chiffrement »

Encrypt 1

sudo docker run -i --rm -w /root -v$(pwd) :/root touatily/openabe oabe_enc -s CP -e "étudiant AND lmd AND informatique AND niveau==4" -i cal_section_m1 -o cal_section_m1_crypter -p departement

Encrypt 2

Sudo docker run -i --rm -w /root -v$(pwd) :/root touatily/openabe oabe_enc -s CP -e "étudiant AND lmd AND informatique AND niveau==4 AND securite" -i cal_groupe_sec_m1 -o cal_groupe_sec_m1_crypter -p departement

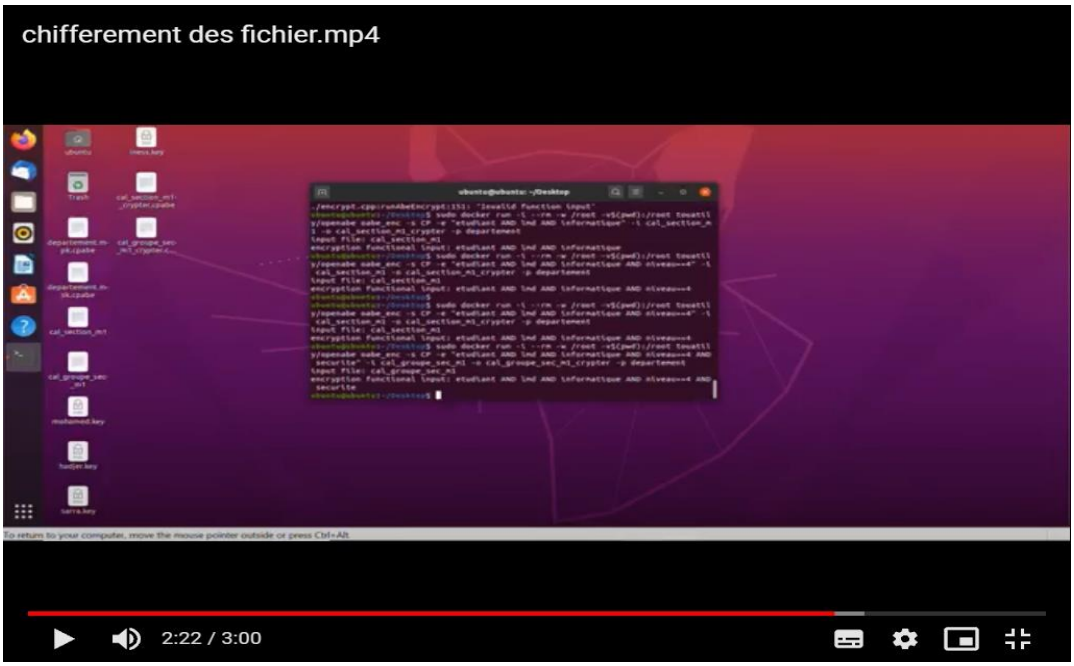

Figure 4.22: le chiffrement

« Phase IV le déchiffrement »

Decrpypt 1 sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k mohamed.key -i cal_section_m1_crypter.cpabe -o cal_section_m1_decrypter -p departement

Decrpypt 1sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k hadjer.key -i cal_section_m1_crypter.cpabe -o cal_section_m1_decrypter -p departement

Decrpypt 1sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k sarra.key -i cal_section_m1_crypter.cpabe -o cal_section_m1_decrypter -p departement

Decrpypt 1sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k iness.key -i cal_section_m1_crypter.cpabe -o cal_section_m1_decrypter -p departement

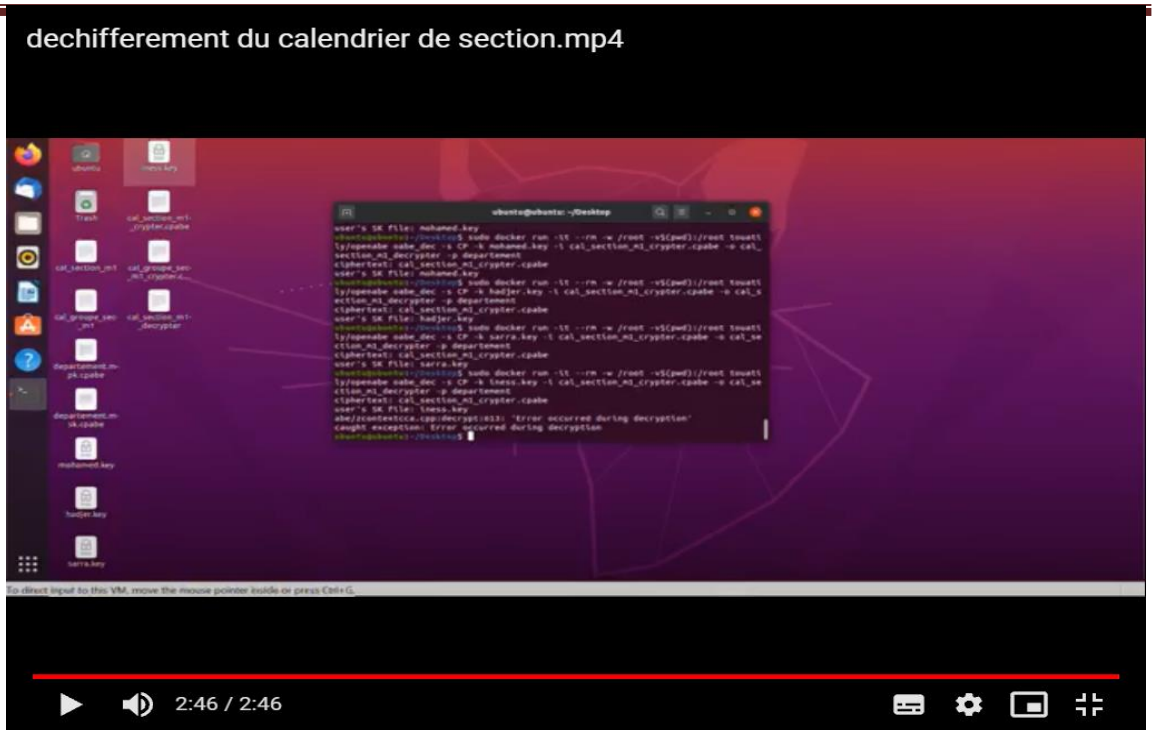

Figure 4.23 : le déchiffrement

« Phase IV le déchiffrement »

Decrpypt 2sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k mohamed.key -i cal_groupe_sec_m1_crypter.cpabe -o cal_groupe_sec_m1_decrypter -p departement

Decrpypt 2sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k hadjer.key -i cal_groupe_sec_m1_crypter.cpabe -o cal_groupe_sec_m1_decrypter -p departement

Decrpypt 2 sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k sarra.key -i cal_groupe_sec_m1_crypter.cpabe -o cal_groupe_sec_m1_decrypter -p departement

Decrpypt 2 sudo docker run -it --rm -w /root -v$(pwd):/root touatily/openabe oabe_dec -s CP -k iness.key -i cal_groupe_sec_m1_crypter.cpabe -o cal_groupe_sec_m1_decrypter -p departement

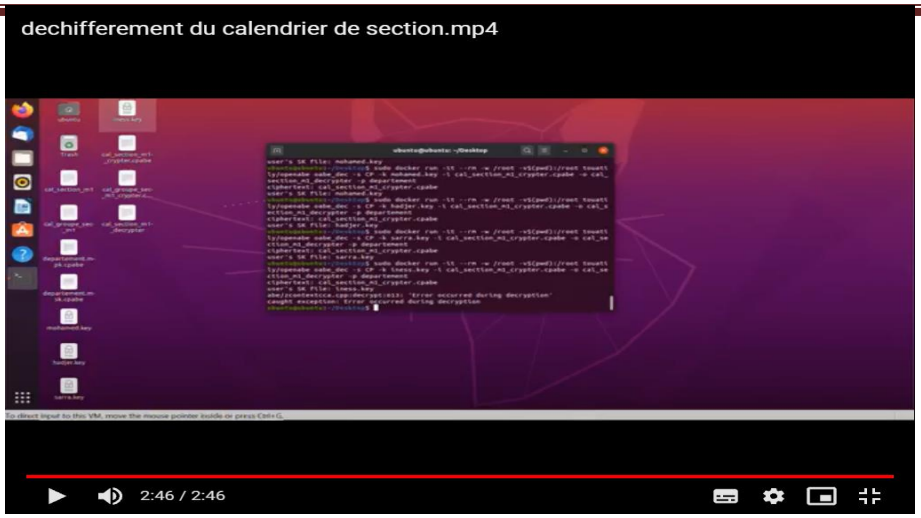

Figure 4.24 : le déchiffrement

Conclusion

À travers cet essaye, nous avons vu tous les aspects majeurs de Docker, rendant ainsi son utilisation très simple, intéressante et très puissante et sur tout sur l'environnement UBUNTU Docker apporte une valeur immédiate, augmentant ainsi rapidement sa productivité. Il vous permet de diffuser donnés plus rapidement tout en réduisant les coûts d'infrastructure et de maintenance, et il fournit de nouvelles expériences client allant des applications monolithiques traditionnelles aux applications Cloud natives.

Ainsi nous avons décrit le processus de réalisation de notre application, en spécifiant les outils de développement et les librairies utilisés. Nous avons présenté les différentes interfaces graphiques qui composent les espaces des utilisateurs de notre système tout en exposant les fonctions qui constituent chaque espace. Nous avons achevé l'implémentation de notre application tout en respectant la conception élaborée.

Conclusion Générale

Conclusion Générale

Il n'y a pas de sécurité absolue, mais des éléments de sécurité relatifs. La recherche d'algorithmes, Protocoles et architectures de sécurité doit donc viser à faire en sorte qu'il ne soit pas rentable par rapport à l'information convoitée.

Notre approche se repose sur le contrôle d'accès avec chiffrement CP-ABE qui offre une solution efficace pour garantir le contrôle des données et sur l'authentification anonyme qui permet de garantir non seulement la confidentialité mais aussi la confiance des utilisateurs.

Nous avons montré la faisabilité de l'adoption de l'ABE dans des systèmes représentatifs. Nous avons constaté que les recherches futures devraient se concentrer sur l'amélioration de l'efficacité de l'ABE, via une sélection rigoureuse des attributs et des optimisations logicielles et matérielles pour la bibliothèque cryptographique.

On constate donc qu'il y a nécessairement des solutions différentes selon la valeur de l'information à transmettre et le choix de l'algorithme est déterminant.

BIBLIOGRAPHIE

- [01] Eli Biham and Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In Burton S. Kaliski, Jr, editor, Advances in Cryptology – CRYPTO ’97, volume 1294 of Lecture Notes in Computer Science, pages 513–525. Springer-Verlag, 1997.
- [02] B. ONigA, S. H. FArr, A. MuNTEANu et V. DADArLAT : Iot infrastructure secured by tls level authentication and pki identity system. In 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pages 78–83, 2018. 44
- [03] Kim Thuat NguyEN, Nouha OuALhA et Maryline LAurENT : Securely outsourcing the ciphertext-policy attribute-based encryption. World Wide Web, 21(1):169–183,2018. ISSN 1573-1413. 49,
- [04] J. Bethencourt et al., Ciphertext-Policy Attribute-Based Encryption, IEEE S&P’07, 2007. URL https://www.google.dz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja& uact=8&ved=2ahUKEwixqOyL39rrAhUDGuwKHUJ9B2wQFjAAegQIAxAB&u rl=https%3A%2F%2Fwww.cs.utexas.edu%2F~bwaters%2Fpublications%2Fpaper s%2Fcp-abe.pdf&usg=AOvVaw0VSl3-q1Ss24vhkpdrwvnb
- [05] M. Ambrosin et al., On the feasibility of attribute-based encryption on smartphone devices, ACM MobiSys/IoT-Sys’15, 2015. URL https://www.google.dz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja& uact=8&ved=2ahUKEwi74M3W3trrAhWxMewKHUvIByoQFjAAegQIBBAB&u rl=https%3A%2F%2Fdl.acm.org%2Fdoi%2F10.1145%2F2753476.2753482&usg =AOvVaw2FCjiePIlLTLgg1-FTkQBf
- [06] H. Ma et al., Verifiable and exculpable outsourced Attribute-Based Encryption for access control in cloud computing,

- [07] M. Asim, M Petkovic et T Ignatenko: Attribute-based encryption with encryption and decryption outsourcing. URL http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1163&context=ism 18/6/2020
- [08] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi, A. M.Rahmani et P. Liljeberg : On the feasibility of attribute-based encryption on internet of things devices. URL https://ieeexplore.ieee.org/abstract/document/7765239/ 05/2020.
- [09]BirMohandAmokrane_DahmouniLyeshttps://www.google.dz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&
- [10] Article CESAR_2017_105_Ould-Yahia_ParadinaABE-report09ABEhttps://www.google.dz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&
- [11] Article cpabe Article TH2895 https://ieeexplore.ieee.org/abstract/document/7765239/ 05/2020.
- [12] http://dspace.univ-tlemcen.dz/bitstream/112/8255/1/Developpement-dune-application-pour-lechange-des-messages-securises.pdf
- [13] IBEZebiche_Tayeb_2004_memoire
- [14] [Cocks 2001] (en) Clifford Cocks, « An identity based encryption scheme based on quadratic residues », Cryptography and Coding, 2001 (lire en ligne [archive], consulté le 19 avril 2016).
- [15] [Neven et Joye 2008] (en) Gregory Neven et Marc Joye, Identity-Based Cryptography, Amsterdam, IOS Press, 2008, 263 p. (ISBN 978-1-58603-947-9, lire en ligne [archive])
- [16] [Shamir 1984] (en) Adi Shamir, « Identity-based cryptosystems and signature schemes », Crypto, 1984 (lire en ligne [archive], consulté le 19 avril 2016).
- [17] [Chase 2007] (en) Melissa Chase, « Multi-authority Attribute Based Encryption », Theory of Cryptography, Springer Berlin Heidelberg, lecture Notes in Computer Science, vol. 4392, 21 février 2007, p. 515–534 (ISBN 978-3-540-70935-0 et 978-3-540-70936-7, ISSN 0302-9743, DOI 10.1007/978-3-540-70936-7_28, lire en ligne [archive], consulté le 17 février 2016)

- [18] [Chase et Chow 2009] Melissa Chase et Sherman S.M. Chow, « Improving Privacy and
- [19] Mémoire fin d'étude La protection des données personnelles des systèmes basés sur l'internet des objets Promotion : 2019/2020
- [20] Security in Multi-authority Attribute-based Encryption », Proceedings of the 16th ACM Conference on Computer and Communications Security, ACM, cCS '09, 1er janvier 2009, p. 121–130 (ISBN 9781605588940, DOI 10.1145/1653662.1653678, lire en ligne [archive], consulté le 17 février 2016)
- [21] Gorbunov, Vaikuntanathan et Wee 2013] (en) Sergey Gorbunov, Vinod Vaikuntanathan et Hoeteck Wee, « Attribute-Based Encryption for Circuits », Dans les actes de Symposium on Theory of Computing, 2013 (lire en ligne [archive])
- [22] Müller, Katzenbeisser et Eckert 2009] (en) Sascha Müller, Stefan Katzenbeisser et Claudia Eckert, « On Multi-Authority Ciphertext-Policy : attribute-Based Encryption », Bulletin of the Korean Mathematical Society, The Korean Mathematical Society, vol. 46, no 4, 2009, p. 803–819 (DOI 0.4134/BKMS.2009.46.4.803, résumé [archive], lire en ligne [archive] [PDF])
- [23] Didier Roche, *Simple comme Ubuntu*, Cergy-Pontoise, InLibroVeritas, coll. « Framabook », 2010 (1re éd. 2007), 356 p. (ISBN 978-2-35922-032-2, lire en ligne [archive]) publié sous licence Creative Commons By-Sa.
- [24] Benoit Caccinolo, Lionel Dricot, Joyce Markoll, Ubuntu, la distribution Linux facile à installer, Eyrolles 2006. (ISBN 221211608X)
- [25] Michael Kofler (en), Ubuntu Linux, livre avec CD-ROM. CampusPress, 2005. (ISBN 274402029X)
- [26] Lionel Dricot, Ubuntu efficace, Eyrolles présentation en ligne [archive], 2009. (ISBN 2-212-12003-6 et 978-2-212-12003-5)