UNIVERSITE SAAD DAHLEB – BLIDA 1

Faculté des Sciences

Département d'Informatique

THESE DE DOCTORAT

Spécialité : Informatique Décisionnel

LA MODELISATION DES ASPECTS DE SECURITE DANS LE CLOUD

Par

Yasmina GHEBGHOUB

Devant le jury composé de:

M.OULD KHAOUA	Professeur ,	U. Blida 1	Président
N.BENBLIDIA	Professeur ,	U. Blida 1	Examinatrice
N.BOUSTIA	M.C.A ,	U. Blida 1	Examinatrice
O. KAZAR	Professeur,	U. Biskra	Examinateur
S. OUKID	M.C.A ,	U. Blida 1	Directrice de la thèse
O. BOUSSAID	Professeur,	U. Lyon2	Co Directeur de la thèse

BLIDA, Novembre 2017

SAAD DAHLEB UNIVERSITY – BLIDA1
Faculty of Sciences
Department of Computer Science

DOCTORAL THESIS

Specialty: Decision support system

DESIGN SECURITY ASPECTS ON CLOUD COMPUTING

by

Yasmina GHEBGHOUB

APPROVED:

M.OULDA KHAOUA	President
N.BENBLIDIA	Examiner
N.BOUSTIA	Examiner
O .KAZAR	Examiner
S. OUKID KHOUAS	Thesis Advisor
O. BOUSSAID	Thesis Advisor

Blida ,November 2017

Abstract

Cloud computing has been developed to deliver information technologies services on demand for organizations or as individual users.

Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. Organizations use the Cloud in a variety of different service models and deployment models (Private, Public, Hybrid, and Community). There are a number of security concerns associated with cloud computing. These issues fall into two broad categories: security issues faced by cloud providers and security issues faced by their customers. The responsibility is shared, however. The provider must ensure that their infrastructure is secure and that their clients' data and applications are protected while the user must take measures to fortify their application and use strong passwords and authentication measures.

In sequence, we think that a security is a concept that can be formalized and should be integrated in all layers of design from the early stages of development. Starting from this idea, we propose a security model based on organization role based access control (ORBAC) and encryption system. This model aims to give users a possibility to control security of their data. In this scheme, a secret key is partitioned using a Galois Field$\mathrm{GF}(2^{\mathrm{m}})$.Our proposal has been aligned with model driven architecture (MDA).

To evaluate our contributions, we have conducted an experimental study. The results computed by our evaluation measures show the importance of our approach of a security in cloud computing.

Keywords: cloud; data security; access control; organization role-based access control; ORBAC; encryption data; model driven architecture; MDA.

Résumé

L'informatique en nuage a été développée pour fournir des services de technologies de l'information à la demande d'organisations ou en tant qu'utilisateurs individuels.

Les solutions informatiques et de stockage en nuage offrent aux utilisateurs et aux entreprises diverses fonctionnalités pour stocker et traiter leurs données dans des centres de données tiers. Les organisations utilisent l'informatique en nuage dans différents modèles de services et modèles de déploiement (privé, public, hybride et communautaire).

Il existe un certain nombre de problèmes de sécurité liés à l'informatique en nuage. Ces problèmes se répartissent en deux grandes catégories: les problèmes de sécurité auxquels sont confrontés les fournisseurs de cloud et les problèmes de sécurité auxquels sont confrontés leurs clients. Toutefois, la responsabilité est partagée. Le fournisseur doit s'assurer que leur infrastructure est sécurisée et que les données et les applications de leurs clients sont protégées pendant que l'utilisateur doit prendre des mesures pour fortifier leur application et utiliser des mots de passe et des mesures d'authentification solides.

En séquence, nous pensons que la sécurité est un concept qui peut être formalisé et devrait être intégré dans toutes les couches de conception dès les premiers stades de développement.

À partir de cette idée, nous proposons un modèle de sécurité basé sur le contrôle d'accès basé sur le rôle organisationnel (ORBAC) et le système de cryptage. Ce modèle vise à donner aux utilisateurs la possibilité de contrôler la sécurité de leurs données. Dans ce schéma, une clé secrète est partitionnée à l'aide d'un Galois Field. Notre proposition a été alignée avec l'architecture pilotée par modèle (MDA).

Pour évaluer nos contributions, nous avons mené une étude expérimentale. Les résultats calculés par nos mesures d'évaluation montrent l'importance de notre approche de la sécurité dans le cloud computing.

Mots clés: cloud; sécurité des données; contrôle d'accès; ORBAC ; cryptage des données; architecture MDA.

الملخص

لقد تم تطوير الحوسبة السحابية لتقديم تكنولوجيا المعلوماتية على شكل خدمات عند الطلب للشركات أو المستخدمين الفرديين على سواء.

الحوسبة السحابية وحلول التخزين توفر للمستخدمين والشركات قدرات مختلفة لتخزين ومعالجة البيانات في مراكز البيانات و التي تعتبر الطرف الثالث .الشركات تستخدم الحوسبة السحابية في مجموعة متنوعة من الخدمات إلا إن هناك عدد من المخاوف الأمنية المرتبطة بهذه التكنولوجيا . وتندرج هذه القضايا إلى فئتين كبيرتين :القضايا الأمنية التي تواجه مقدمي تكنولوجيا السحابة او القضايا والأمن التي تواجه العملاء إذن المسؤولية مشتركة .، ولكن يجب على المزود ضمان أن أمان البنية التحتية و حماية البيانات والتطبيقات لعملائها في حين يجب على المستخدم اتخاذ التدابير اللازمة لتحصين حساباته كاستخدام كلمات مرور قوية .و من خلال ذلك ، نعتقد أن الأمن هو مفهوم يمكن صياغته و دمجه في جميع الطبقات من مراحل تصميم البرامج .

وانطلاقا من هذه الفكرة، فإننا نقترح برنامج قائم على أساس رقابة الولوج إلى المعطيات المخزنة ونظام التشفير .

و لقد قمنا بتصميمه بفضل نموذج)MDA(. و يهدف بحثنا إلى إعطاء المستخدمين إمكانية السيطرة على أمن البيانات الخاصة بهم .

لتقييم هذا العمل، أجرينا دراسة تجريبية ولقد أظهرت النتائج الجيدة أهمية نهجنا في دعم الأمن للحوسبة السحابية.

الدلالات : الحوسبة السحابية، نظام التشفير، حماية المعلومات ، رقابة الولوج، نموذج) MDA. (

ACKNOWLEDGMENTS

I would like to express my special appreciation and thanks to my advisor Professors Pr. Saliha Khouas Oukid and Pr. Omar Boussaid , you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research . Your advice on both research as well as on my career have been priceless. I would also like to thank my committee members, professor M.OULD KHAOUA, professor N.BENBLIDIA, professor N.BOUSTIA ,O. KAZAR for serving as my committee members even at hardship.

I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

A special thanks to my family. Words cannot express how grateful I am to my mother and father for all of the sacrifices that you've made on my behalf. Your prayer for me was what sustained me thus far. I would also like to thank all of my friends who supported me in writing, and incented me to strive towards my goal. At the end I would like express appreciation to my husband .

I also want to thank Ms Sana Aroussi for her help and all the members of LRDSI Research Laboratory for moments shared during this PhD, especially PhD students from different promotions.

I want to thank Department Headand all the staff of Computer Sciences Department and the Faculty of Science

Contents

LIST OF FIGURES

LIST OF TABLES

Introduction

1. Context

Cloud computing provides an extensible and powerful environment for growing amounts of services and data by means of on-demand self-service. It also relieves the client's burden of management and maintenance by providing a comparably low-cost, scalable, location-independent platform. However, cloud computing is also facing many challenges for data security as the users outsource their sensitive data to clouds, which are generally beyond the same trusted domain as data owners [**1**].

Additionally, security issue should be considered in the development of an approach because it is one of the most main design issues. Considering that the information managed by cloud computing is frequently highly sensitive, and sometimes refers to personal data (protected under the law in most countries), Data may be accessed by unauthorized users if it is not password protected. Consumers always want security of their data. If propagation of data is allowed, then it creates major problems.

From this, data stored on cloud should be protected from unauthorized accesses. In fact, for adequate security, we must use fine-grained flexible authorization models and access control mechanisms.

Therefore, rather than considering security once the system has been completely built, we believe that security and privacy measures should be integrated in all layers of design, from the early stages of its development as another relevant requirement, meaning that much more robust, secure and platform independent products will be produced [**2**].

2. Problem Statement

A cloud computing is a computing model that makes IT resources such as servers, middleware, and applications available over the Internet as services to business organizations in a self-service manner. Although cloud computing offers many advantages to the consumers, it also has several security problems.

Security in the cloud is a crucial issue. It is often generally reduced to an unauthorized access or malicious activities. Therefore, rather than considering security once the system has been completely built, we think that security is a concept that can be formalized and should be integrated in all layers of design from the early stages of development.

Therefore, the following questions appear:

- How to design security aspects on Cloud?
- How to secure a cloud against unauthorized access and malicious activity?
- How to increase a trust between costumers and cloud provider?

To answer these questions, it is necessary to find architecture to model the security aspects in our proposition.

3. Contributions

To address these issues posed before. We propose through this thesis to make the following contributions:

First, we propose a security model to control access to data on cloud, we use an access model based on Organization Role Based Access Control and we employ a encryption algorithm to secure data stored on cloud.

Next, In order to develop secure approach considering security issues in the whole development process, from an early development stage to the final implementation, our proposal has been aligned with Model Driven Architecture (MDA) in which security models are embedded and scattered throughout the high level system models, which are transformed until their final implementation

according to the MDA strategy. One important of architecture is to define a method with a set of models that can be used by designers.

Finally, to enforce a security of our approach, we propose to distribute the key encryption into several partitions where each partition is stored on a different server. The partitions themselves do not reveal any information and hence are implicitly secured. Figure 1 shows the architecture of ours contributions.

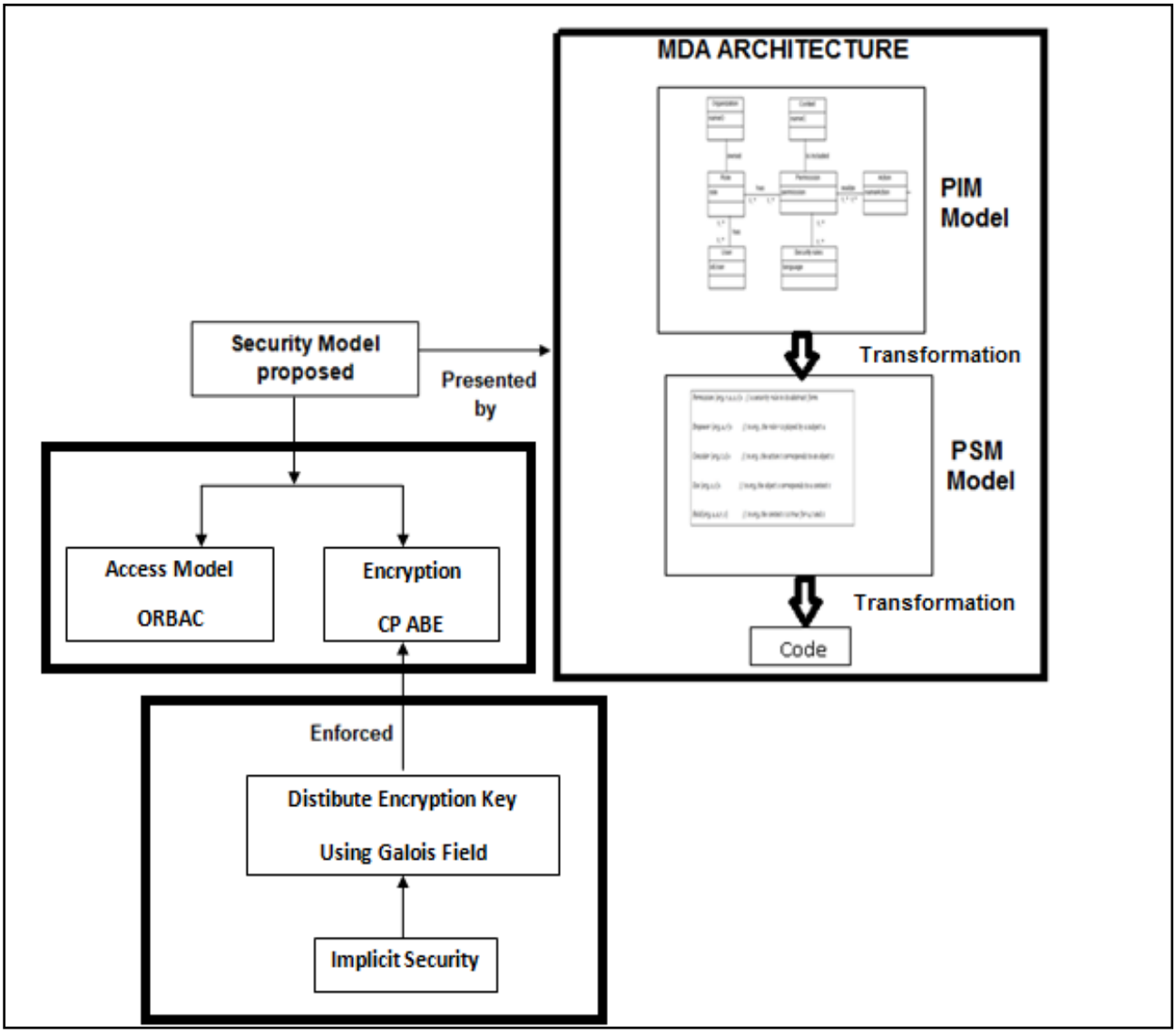

Figure 1: Scheme represents CP ORBAC MODEL

4. Organization

The remainder of this thesis is organized in two parts:

A part one surveys the state of the art of cloud computing and the security on cloud.

In the second part, the subsequent three chapters present the main contributions of this thesis.

- Chapter 3 , describes a proposed approach that can secure access to data .This approach associates for each user an access structure based on Organization Role based Access Control (ORBAC). This formula defines us that only authorized users are allowed to access the data resource after decryption of data because data will be encrypted by assigning for each resource a secret key.
- Chapter 4, we propose a methodological approach for the model driven development of secure data on cloud. This proposal is within a model driven methodology for the development of security access system based on the Model Driven Architecture (MDA) proposed by the Object Management Group (OMG).
- Chapter 5, to reinforce a security of our last proposition, we propose to partition encryption key used on encryption phase and send it randomly to the chosen servers. We use a mathematical arithmetic based on the Galois Field; it is particularly useful in translating computer data as they are represented in binary forms. No explicit encryption of data is required to secure each partition. The partitions themselves do not reveal any information and hence are implicitly secured. The thesis concludes with our future projects.

Chapter 1
Computing Security

1.1 Introduction

Computer security is the protection of information systems from theft or damage to the hardware, the software, and to the information on them, as well as from disruption or misdirection of the services they provide

In this chapter, we introduce the main concepts used in computer security. In addition we provide different technical and mechanism used to secure a system such as access control models and cryptographic mechanisms. However, we present a comparison between access models DAC, MAC, RBAC and ORBAC.

1.2 Computing Security

In the domain of computing, security can cover several meanings. The first corresponds to Trust. It is not a new research topic in computing science, spanning areas as diverse as security and access control in computer networks, reliability in distributed systems, game theory and agent systems, and policies for decision making under uncertainty [**3**]. Perhaps the most notable example was the development of the Trusted Computer System Evaluation Criteria (TCSEC) [**4**] in the late 70s and early 80s. Here, trust was used in the process of convincing observers that a system (model, design or implementation) was correct and secure [**5**].

The concept of trust, adjusted to the case of two parties involved in a transaction, can be described as follows: ‘‘An entity A is considered to trust another entity B when entity A believes that entity B will behave exactly as expected and required’’ [**6**].

The notion of trust in an organization could be defined as the customer’s certainty that the organization is capable of providing the required services accurately and infallibly. A certainty which also expresses the customer’s faith in its moral integrity, in the soundness of its operation, in the effectiveness of its security mechanisms, in its expertise and in its abidance by all regulations and laws, while at the same time, it also contains the acknowledgement of a minimum risk factor, by the relying party. The notion of security refers to a given situation where all possible risks are either eliminated or brought to an absolute minimum [**7**].

Trust in a cloud environment depends heavily on the selected deployment model, as governance of data and applications is outsourced and delegated out of the owner’s strict control.

The second corresponds to security. It is related to the important aspects of confidentiality, integrity and availability; they thus become building blocks to be used in designing secure systems. These important aspects of security, apply to the three broad categories of assets which are necessary to be secured, data, software and hardware resources.

1.2.1 Confidentiality and privacy

Confidentiality refers to only authorized parties or systems having the ability to access protected data. The threat of data compromise increases in the cloud, due to the increased number of parties, devices and applications involved, that leads to an increase in the number of points of access. Delegating data control to the cloud, inversely leads to an increase in the risk of data compromise, as the data becomes accessible to an augmented number of parties. A number of concerns emerge regarding the issues of multi tenancy, application security and privacy [**8**].

Data confidentiality in the cloud is correlated to user authentication. Protecting a user's account from theft is an instance of a larger problem of controlling access to objects, including memory, devices, software etc. Software confidentiality refers to trusting that specific applications or processes will maintain and handle the user's personal data in a secure manner.

1.2.2 Integrity

A key aspect of Information Security is integrity. Integrity means that assets can be modified only by authorized parties or in authorized ways and refers to data, software and hardware. Data Integrity refers to protecting data from unauthorized deletion, modification or fabrication. Managing an entity's admittance and rights to specific enterprise resources ensures that valuable data and services are not abused, misappropriated or stolen. By preventing unauthorized access, organizations can achieve greater confidence in data and system integrity.

1.2.3 Availability

Availability refers to the property of a system being accessible and usable upon demand by an authorized entity. System availability includes a systems ability to carry on operations even when some authorities misbehave. The system must have the ability to continue operations even in the possibility of a security breach. Availability refers to data, software but also hardware being available to authorized users upon demand [**9**].

1.3 Vulnerability, Threat and Attack

When discussing network security, the three common terms used are as follows [10]:

1.3.1 Vulnerability

- **A weakness:** that is inherent in every network and device. This includes routers, switches, desktops, servers, and even security devices themselves. Networks are typically plagued by one or all of three primary vulnerabilities or weaknesses:

- **Technology weaknesses:** Computer and network technologies have intrinsic security weaknesses. These include TCP/IP protocol weaknesses, operating system weaknesses, and network equipment weaknesses.
- **Configuration weaknesses:** Network administrators or network engineers need to learn what the configuration weaknesses are and correctly configure their computing and network devices to compensate.
- **Security policy weaknesses:** Security policy weaknesses can create unforeseen security threats. The network can pose security risks to the network if users do not follow the security policy.

1.3.2 Threats

The people eager, willing, and qualified to take advantage of each security weakness, and they continually search for new exploits and weaknesses [**10**]. There are four primary classes of threats to network security, as Figure 1.1 depicts.

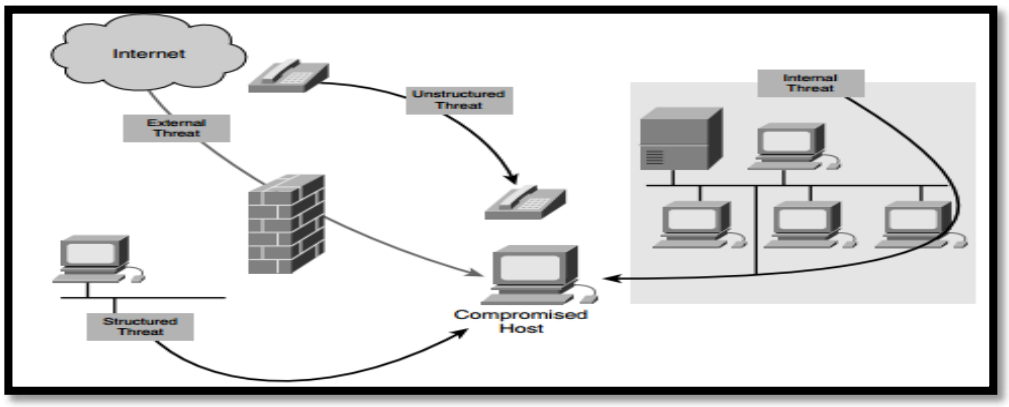

Figure 1.1: Primary Classes of Threats to Security Networks **[**10**]**

- **Unstructured threats:** consist of mostly inexperienced individuals using easily available hacking tools such as shell scripts and password crackers.

- **Structured threats:** come from hackers who are more highly motivated and technically competent. These people know system vulnerabilities and can understand and develop exploit code and scripts. They understand, develop, and use sophisticated hacking techniques to penetrate unsuspecting businesses.
- **External threats:** can arise from individuals or organizations working outside of a company. They do not have authorized access to the computer systems or network. They work their way into a network mainly from the Internet or dialup access servers.
- **Internal threats:** occur when someone has authorized access to the network with either an account on a server or physical access to the network.

1.3.3 Attacks

The threats use a variety of tools, scripts, and programs to launch attacks against networks and network devices. Typically, the network devices under attack are the endpoints, such as servers and desktops. Four primary classes of attacks exist:

- **Reconnaissance:** is the unauthorized discovery and mapping of systems, services, or vulnerabilities.
- **Access:** System access is the ability for an unauthorized intruder to gain access to a device for which the intruder does not have an account or a password.
- **Denial of service:** Denial of service implies that an attacker disables or corrupts networks, systems, or services with the intent to deny services to intended users. DoS attacks involve either crashing the system or slowing it down to the point that it is unusable
- **Worms, viruses, and Trojan horses:** Malicious software is inserted onto a host to damage a system; corrupt a system; replicate itself; or deny services or access to networks, systems or services. They can also allow sensitive information to be copied or echoed to other systems.

1.4 Solutions and mechanisms to secure a system

To eliminate vulnerabilities, against an attack, and ensure a high level of protection to a network or information system, we can use mechanisms, tools and procedures that are called, in general, solutions or security measures. For example, an identification and authentication service helps reduce the risk of intrusion into a system.

Access control models will be presented as a necessary means to enhance system security.

We also explain other against measures to strengthen security as cryptographic mechanisms.

1.4.1 Access Control

In today's information technology, authorization is concerned with the ways in which users can access resources in the computer system, or informally speaking, with "who can do what." Access control is arguably the most fundamental and most pervasive security mechanism in use today.

Access control is generally a policy or procedure that allows, denies or restricts access to a system. It may, as well, monitor and record all attempts made to access a system.

In this course, access Control may also identify users attempting to access a system in an unauthorized way.

It is a mechanism which is very much important for protection in computer security. Various access control models are in use, including the most common Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role Based Access Control (RBAC).

All these models are known as identity based access control models. In all these access control models, user (subjects) and resources (objects) are identified by unique names. Identification may be done directly or through roles assigned to the subjects.

These access control methods are effective in unchangeable distributed system, where there are only a set of Users with a known set of services.

The fundamental goal of any access control mechanism is to provide a verifiable system for guaranteeing the protection of information from unauthorized and inappropriate access as outlined in one or more security policies. In general, this translation from security policy to access control implementation depends on the nature of the policy but involves the inclusion of at least one of the following controls:

- **Confidentiality:** refers to the need to keep information secure and private. This category may include anything from state secrets to confidential memoranda, financial information, and security information such as passwords.
- **Integrity:** refers to the concept of protecting information from being improperly altered or modified by unauthorized users. For example, most users want to ensure that bank account numbers used by financial software cannot be changed by anyone else and that only the user or an authorized security administrator can change passwords.
- **Availability**: refers to the notion that information is available for use when needed. Attacks that attempt to overload corporate Web servers, widely reported in the popular press, are attacks on availability.
- The condition of confidentiality requires that only authorized users can read information and the condition of integrity requires that only authorized users can alter information in authorized ways. Access control is less obviously central to preserving availability.

1.4.1.1 Terminology

A reasonably consistent terminology has developed for describing access control models and systems. Almost any access control model can be stated formally using the notions of *users, subjects, objects, operations,* and *permissions,* and the relationships between these entities. It is important to understand these terms.

The term *user* refers to people who interface with the computer system.

In many design, it is possible for a single user to have multiple login IDs, and these IDs may be simultaneously active. Authentication mechanisms make it possible to match the multiple IDs to a single human user, however. An instance of a user's dialog with a system is called a *session*.
A computer process acting on behalf of a user is referred to as a *subject.*

Note that in reality, all of a user's actions on a computer system are performed through some program running on the computer. A user may have multiple subjects in operation, even if the user has only one login and one session. For example, an e-mail system may be operating in the background, fetching e-mail from a server periodically, while the user operates a Web browser. Each of the user's programs is a subject, and each program's accesses will be checked to ensure that they are permitted for the user who invoked the program.

An *object* can be any resource accessible on a computer system, including files, peripherals such as printers, databases, and fine-grained entities such as individual fields in database records. Objects are traditionally viewed as passive entities that contain or receive information, although even early access control models included the possibility of treating programs, printers, or other active entities as objects [**11**] .

An *operation* is an active process invoked by a subject. Early access control models that were concerned strictly with information flow (i.e., read-and-write access) applied the term subject to all active processes, but RBAC models require a distinction between subject and operation.

Permissions (or privileges) are authorizations to perform some action on the system. Generally, the term permission refers to some combination of object and operation. A particular operation used on two different objects represents two distinct permissions, and similarly, two different operations applied to a single object represent two distinct permissions.

1.4.2 Access Control models

Access control models have four flavors: Mandatory Access Control (MAC), Role Based Access Control (RBAC), Discretionary Access Control (DAC), and Rule Based Access Control (RBAC).

1.4.2.1 Discretionary Access Control (DAC)

Discretionary Access Control (DAC) is a user-centric access control model in the sense that a file owner determines the permissions that are assigned to other users requiring access to the file. There is no central control so this model is easy to implement in a distributed applications on the Web. Using a DAC mechanism allows users control over the access rights to their files without the necessity of complying with a set of pre-specified rules. When these rights are managed correctly, only those users specified by the file owner may have some combination of read, write, execute, etc. permissions (privileges) on the file. The figure 1.2 shows an illustration of DAC model.

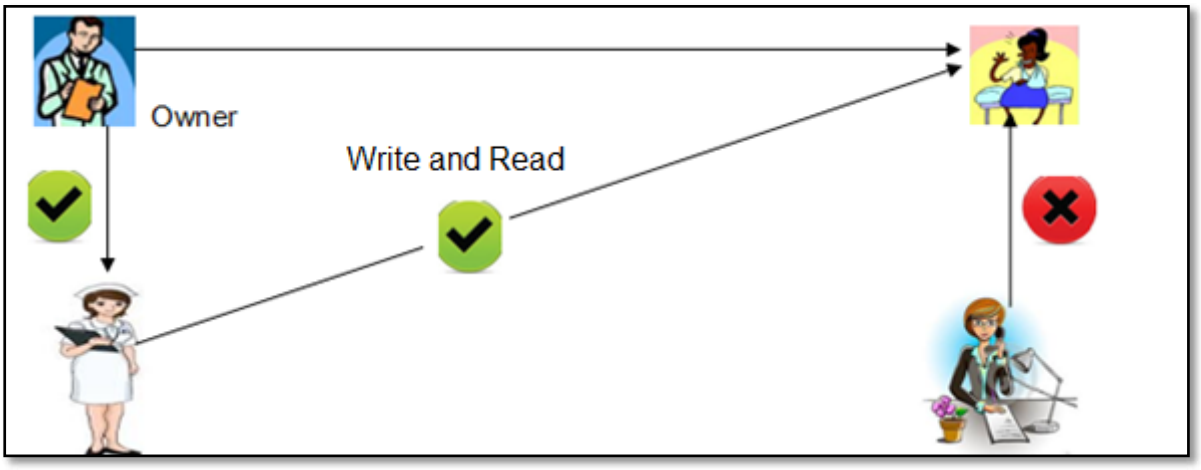

Figure 1.2: Illustration of DAC model [11]

In this model, permissions are represented by a matrix where each row represents a user and each column represents a resource. The figure 1.3 shows an example about an access matrix.

	File 1	File2	File3	Program
Nadji	own read write	read write		execute
Omar	read		read write	
Ali			read	execute read

Figure 1.3: Example of an Access Matrix

1.4.2.2 Mandatory Access Control (MAC)

Besides the confinement and information semantics problems that the access control matrix model faces, a key problem that the DAC model faces is vulnerability to Trojan Horse attacks. Trojan horse attacks are typically provoked by deceiving valid users into accepting to run code that then allows a malicious user to get access to information on the system [**11**].

Loosely defined as any access control model that enforces security policies independent of user operations, Mandatory Access Control is usually associated with the 1973 Bell- La Padula Model [**12**] of multi-level security.

In MAC users do not have the authority to override the policies and it totally controlled centrally by the security policy administrator. MAC is a system-wide policy which defines who is allowed to have access; individual user cannot change that access rules

It totally relies on the central system .MAC policies are defined by the system administrator. The basic principle of MAC is to control access according to the user's clearance and the object's classification. These classifications are divided into security levels (one can refer to MAC as a multilevel access control); the higher the level is, the more confidential the information is. For example, the common government classifications are unclassified, confidential, secret, and top-secret [**12**], as Figure 1.4 describes.

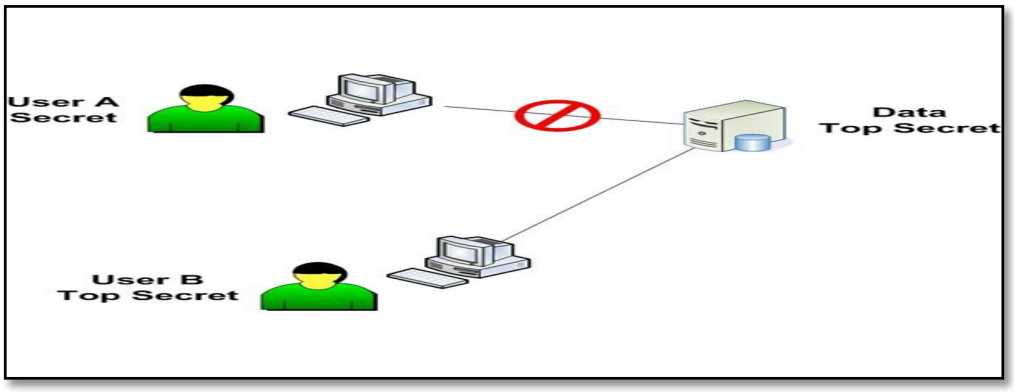

Figure 1.4: Illustration of an Example of Mac Model [12].

1.4.2.3 Role Based Access Control

RBAC has emerged as the primary alternative to MAC and DAC because it is much better suited to the needs of commercial users than these earlier models. This policy is very simple to use. RBAC is very useful method for controlling what type of information users can utilize on the computer, the programs that the users execute, and the changes that the users can make.

In RBAC roles for users are assigned statically, which is not used in dynamic environment. It is more difficult to change the access rights of the user without changing the specified roles of the user. RBAC is mostly preferable access control model for the local domain. The Figure shows the principle of RBAC model [**13**], as Figure 1.5 shows

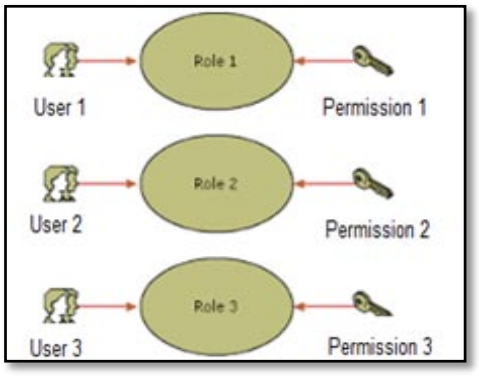

Figure 1.5: Illustration of Principle of RBAC [13]

Essentially, role based access control policies need to identify the roles in the system. A role can be defined as a set of responsibilities and actions associated with a particular working activity. In an Access control security model, a role is considered as a job related access rights which can be given to the authorized users within an organization. It allows authorized user to achieve its associated responsibilities.

1.4.2.4 Organization Based Access Control

The ORBAC model generalizes RBAC models and adds an Organizational dimension to politic where an organization is an entity that is responsible for managing a set of security rules (obligations, permissions, prohibitions) on which permit to control the access of user.

One of the main goals of the ORBAC model is to allow the policy designer to define a security policy independently of the implementation. The chosen method to fulfill this goal is the introduction of an abstract level where the role, activity and view concepts abstract the subject, action and object concepts.

In ORBAC model, the term user refers to people who interface with the computer system. In many designs, it is possible for a single user to have multiple login IDs, and these IDs may be simultaneously active. Authentication mechanisms make it possible to match the multiple IDs to a single human user.

An object can be any resource accessible on a computer system, including files, peripherals such as printers, databases, and fine-grained entities such as individual fields in database records. Objects are traditionally viewed as passive entities that contain or receive information, although even early access control models included the possibility of treating programs, printers, or other active entities as objects [**14**].

The following figure shows how those concepts are related:

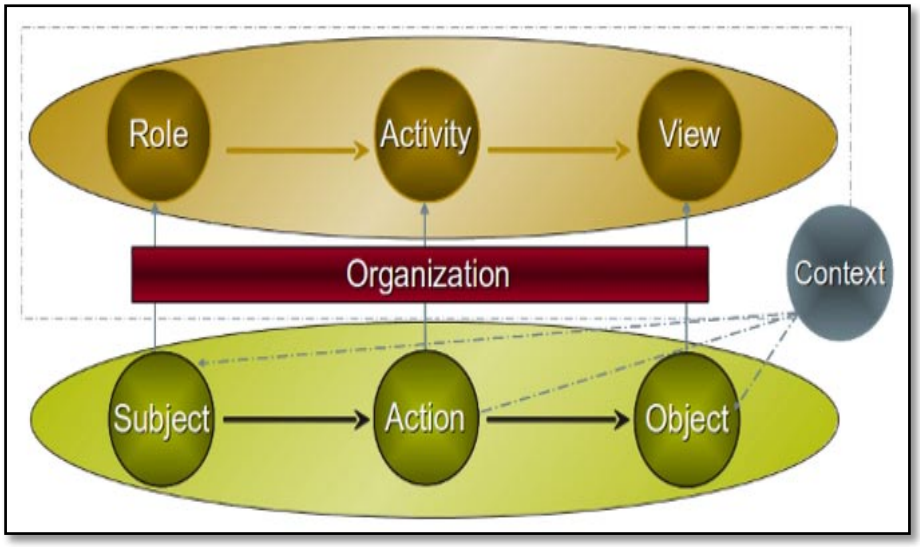

Figure 1.6: The ORBAC Model

There are different context grouped by type as following

- **Temporal context:** These contexts control the validity of privileges.
- **Spatial context:** It can be linked to belonging to a network, or position geographic or other spatial situation.
- **Supposed user context:** In exceptional cases, permissions can be given as they would be prohibited in a normal case.
- **Provisional context:** This context allows to give privileges based history. For an example, limit access to a document twice.

In some cases, the security policy based on access control may be incomplete. Therefore, it must reinforce security through other measures against such cryptographic mechanisms.

1.4.3 Comparative analysis of the Access Control Models

The next Table summarizes our discussion and comparative analysis of the DAC, MAC, RBAC and ORBAC:

	DAC	**MAC**	**RBAC**	**ORBAC**
Authentication	User	Server	Server	Server
Review of Access Rights	User	Server	Server	Server
Access Rights Propagation	User	Server	Server	Server
Access RightsRevocation	User	Server	Server	Server
Implementation	Access Control list	Latice Model	Latice Model	Latice Model

Table 1.1 : Comparaison DAC,MAC, RBAC,ORBAC

1.5 Cryptographic Mechanisms

Cryptography is the art or science of secret writing, or more exactly, of storing information (for a shorter or longer period of time) in a form which allows it to be revealed to those you wish to see it yet hide it from all others. A cryptosystem is a method to accomplish this. Cryptanalysis is the practice of defeating such attempts to hide information.

Cryptology includes both cryptography and cryptanalysis. The original information to be hidden is called "plaintext". The hidden information is called "ciphertext". Encryption is any procedure to convert plaintext into ciphertext. Decryption is any procedure to convert ciphertext into plaintext. An encryption algorithm is a method of transforming a message to add some cryptographic protection, such as confidentiality or integrity. Most encryption algorithms involve one or more keys, which are cryptographic variables, often unique to one user, that control the algorithm and provide security against attackers [**15**] , as Figure 1.7 depicts.

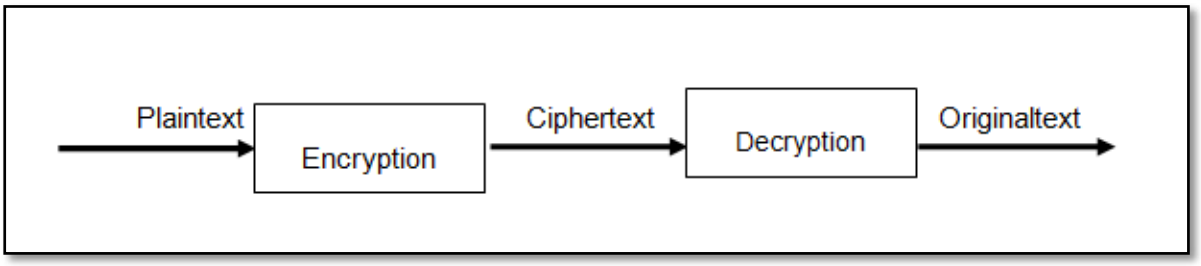

Figure 1.7: Encryption and Decryption

The introduction of such algorithms started at the 70's. The most robust and secure asymmetric algorithm was proposed by Rivest, Shamir and Adelmann (RSA) in 1977 and proved to become a standard, with a large basis of products and applications that are still in operation. At the same time, a symmetric crypto algorithm was adopted by the National Bureau of Standards, by evolving an IBM's earlier crypto-system known as LUCIFER.

The Data Encryption Standard (DES) was released in January 1977, and reviewed every five years. The standardization of the DES algorithm ended in 1998, with the announcement of the Advanced Encryption Standard contest [**16**]. There are two basic techniques for encrypting information: symmetric encryption (also called secret key encryption) and asymmetric encryption (also called public key encryption.).

There are two basic techniques for encrypting information: symmetric encryption (also called secret key encryption) and asymmetric encryption (also called public key encryption.)

1.5.1 Symmetric Encryption

The symmetric key encryption is a cryptography technique that uses a shared secret key to encrypt and decrypt the data. Symmetric encryption algorithms encrypt and decrypt messages with a key in such a way that it is difficult to decrypt without the key. Because the encryption and decryption keys in a secret-key cryptosystem are the same. There are two types of symmetric encryption algorithms: stream ciphers and block ciphers which provide bit-by-bit and block

encryption respectively. There are various symmetric key algorithms such as DES, TRIPLEDES, AES, and BLOWFISH, as Figure 1.8 illustrates.

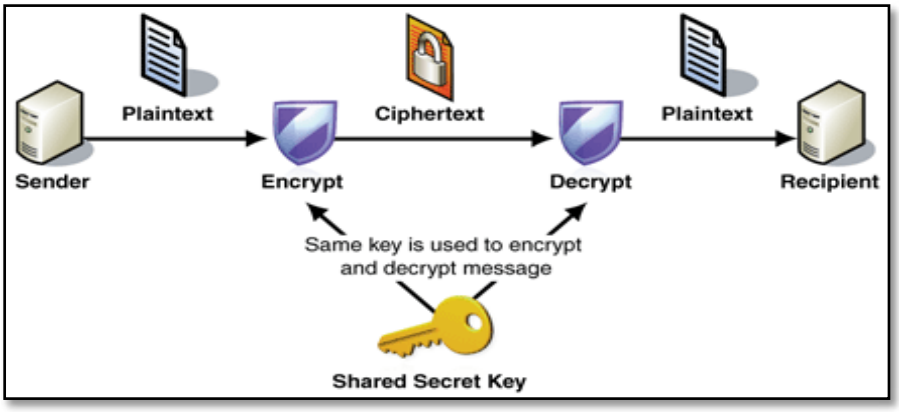

Figure 1.8 : Symmetric Encryption **[**17**]** .

1.5.2 Asymmetric encryption

The problem with secret keys is exchanging them over the Internet or a large network while preventing them from falling into the wrong hands. Anyone who knows the secret key can decrypt the message. The answer to this problem is asymmetric encryption, in which there are two related keys-a key pair.

A public key is made freely available to anyone who might want to send you a message. A second, private key is kept secret, so that only you know it. These algorithms encrypt and decrypt messages with two different keys in such a way that it is difficult to decrypt without the decryption key, as Figure 1.9 demonstrates.

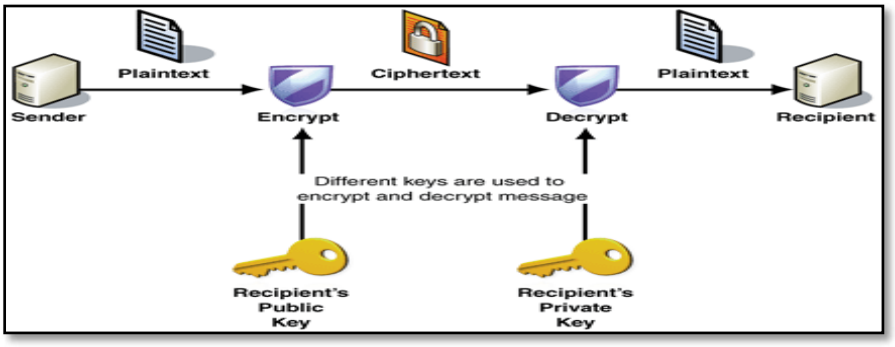

Figure 1.9: Asymmetric Encryption [**17**]

1.5.3 Attribute Based Encryption

In [**18**], they introduce the concept of Attributed-Based Encryption (ABE). In an ABE system, a user's keys and ciphertexts are labeled with sets of descriptive attributes and a particular key can decrypt a particular ciphertext only if there is a match between the attributes of the ciphertext and the user's key.

The cryptosystem of Sahai and Waters allowed for decryption when at least k attributes overlapped between a ciphertext and a private key. While this primitive was shown to be useful for error-tolerant encryption with biometrics, the lack of expressibility seems to limit its applicability to larger systems.

1.5.4 Intrusion Detection

Intrusion Detection is the art of detecting inappropriate or suspicious activity against computer or networks systems. Today, it is difficult to maintain computer systems or networks devices up to date, numerous breaches are published each day. IDS monitor the usage of such systems and detect the apparition of insecure states [**10**]. This insecure state can be either an attempt from internal users to abuse their privileges or outside users (attackers) to exploit security vulnerabilities, as Figure 1.10 shows.

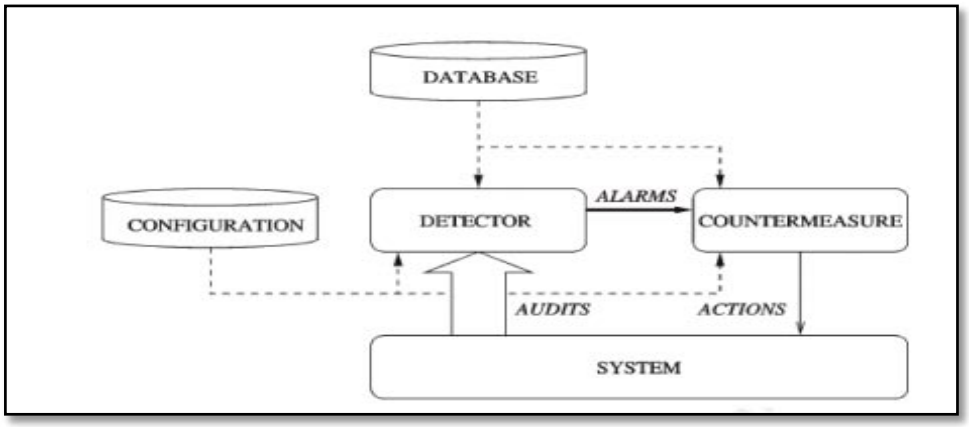

Figure 1.10 : A simple Intrusion detection system **[**10**]**

This detector uses three kinds of information:

- **Technique information:** used to detect intrusion (for example signature database)
- **Configuration information:** about the current state of system
- **Audit trail:** The detector eliminates all unnecessary information, determines if this action can be considered as a symptom of an intrusion, and takes an action (sends alerts for example).

1.6 Conclusion

In this chapter, we introduced the main concepts used in computer security. In addition we provided different technical and mechanism used to secure a system such as access control models and cryptographic mechanisms. However, we presented a comparison between access models DAC, MAC, RBAC and ORBAC. In the next, we propose a survey on security issues and the existing solutions to secure cloud computing.

Chapter 2:

A Survey on Security Issues and the Existing Solution in Cloud

2.1 Introduction

Cloud computing is changing the way industries and enterprises do their businesses in that dynamically scalable and virtualized resources are provided as a service over the Internet. This model creates a brand new opportunity for enterprises.

In this chapter, we focus on some papers that show different risks in the cloud and the different existing solutions that address these various problems. Our work is organized as follow: in the first and second sections, we will define cloud computing and its various models. In the third section, we related the services that inspired cloud computing, and in the forth section, we evocate the various advantages and disadvantages of this technology. Section five exposes some challenges facing the cloud and existing solutions. Finally, we conclude with a summary and a future proposal that will be our next work in this area.

2.2 Definition of Cloud Computing

The name cloud computing was inspired by the cloud symbol that is often used to represent the Internet in flow charts and diagrams. A distinct migration to the clouds has been taking place over recent years with end users, ‘‘bit by bit’’ maintaining a growing number of personal data, including bookmarks,

photographs, music files and much more, on remote servers accessible via a network. Cloud computing is empowered by virtualization technology; a technology that actually dates back to 1967, but for decades was available only on mainframe systems.

In its quintessence, a host computer runs an application known as a hypervisor; this creates one or more virtual machines, which simulate physical computers so faithfully, that the simulations can run any software, from operating systems, to end-user applications [**19**].

Cloud Computing is a computer model which provide services in the form of on demand services, accessible from anywhere, anytime and by anyone, including clouds referring to the internet and the web.

Cloud computing is a concept still young but not so new that. We find the beginnings in the 1990, when the predicted mass adoption of grid computing, beyond only scientific computing applications. Basically, the principle is the same: it is thanks to technology virtualization pool of computing resources geographically dispersed to form a common virtual resource use "on demand". Found in cloud services, applications, processes all types, at least more customizable and that you can subscribe for free, without the need for other hardware resources that terminal (PC, phone or game console...) connected to the internet.

The National Institute of Standards and Technology (NIST) [**20**] defined cloud computing as ‘‘a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

2.3 The Services of Cloud Computing

In Cloud computing, everything is treated as a service (i.e. XaaS), e.g. SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service). These services define a layered system structure for cloud computing Figure 3.1 shows different services.

2.3.1 Infrastructure As A Service

The infrastructure provides computing capacity and storage and networking connectivity. Servers, storage systems, switches, routers and other equipment are available to manage a workload required by applications. Infrastructure as a service, allows having a demand infrastructure that can host and run applications, services, or store data. Specifically, it is characterized by physical infrastructure often made available by a provide service. It will include a virtualization solution allowing the creation of "virtual data centers".

2.3.2 Platform As A Service

PaaS provides users with a high level of abstraction that allows them to focus on developing their applications and not worry about the underlying infrastructure. Just like the SaaS model, users do not have control or access to the underlying infrastructure being used to host their applications at the PaaS level. Among the solutions include Microsoft Windows Azure, Google AppEngine, and Force.com from Salesforce. Each Platform as a Service offers different development models. Google AppEngine is limited to Java and Python while working with Windows Azure languages.Net, Python, PHP, Ruby, and Java.

2.3.3 Software As A Service

Software as a Service allows users to simply make use of a web-browser to access software that others have developed and offer as a service over the web. At the SaaS level, users do not have control or access to the underlying infrastructure being used to host the software.

2.4 The Different Categories of Cloud Computing

The cloud can be deployed in three models . They are described in different ways. In general, it is described as below:

2.4.1 The Public Cloud Computing

The public cloud is one in which the services and infrastructure are provided off-site over the internet. That are shared which anyone can access using an internet connection and a credit card on a usage basis without subscription. So, these are virtualized infrastructures that are shared by several users. It is easily accessible and managed from a self-service portal.

2.4.2 The Private Cloud Computing

A private cloud is one in which the services and infrastructure are maintained on a private network. Nevertheless, private cloud computing is unlike public cloud, owned and managed privately, access can be limited to a single business or a part of it. The private cloud computing may well seem safer in terms of security, stability, privacy and data persistence.

2.4.3 The Hybrid Or Mixed Cloud

It combines the use for the same company, a private cloud and a public cloud.

2.4.4 The Community Cloud

Dedicated to a professional community that includes partners, subcontractors ... to work collaboratively on a project or government cloud dedicated state institutions.

2.5 The Benefits and Disadvantages of Cloud Computing

Cloud computing seems to promise a great future. Here, we show some of its benefits and disadvantages of cloud:

2.5.1 The Benefits of Computing Cloud

Cloud computing offers the possibility of extending the information system of an enterprise at the request of the latter, A number of key characteristics of cloud computing has been identified [**22**], [**23**] .

- **Flexibility/Elasticity:** users can rapidly provision computing resources, as needed, without human interaction. Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out or up.
- **Scalability of infrastructure**: new nodes can be added or dropped from the network as can physical servers, with limited modifications to infrastructure set up and software. Cloud architecture can scale horizontally or vertically, according to demand.
- **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous platforms (e.g., mobile phones, laptops).

2.5.2 The disadvantages of cloud computing

Many people or companies are against this notion. The problem that comes up most is security related.

- How to guarantee the security of information stored in the cloud?
- No there is no risk of intrusion, loss or damage to data?

2.6 Security on Cloud

Cloud computing is a model for enabling on-demand network access in order to share computing resources such as network bandwidth, storage, applications, etc The advantage of cloud is cost savings. The prime disadvantage is security. Cloud

computing is used by many software industries nowadays. Since the security is not provided in cloud, many companies adopt their unique security structure [**24**] for example Amazon has its own security.

Security is not a new issue and now it is recognized as one of the most complex problems, it has been an issue in an increasingly growing network connectivity, size and implementation of new information technologies [**25**]. Various malicious activities from illegal users have threatened this technology such as data misuse, inflexible access control and limited monitoring. The occurrence of these threats may result into damaging or illegal Problems of security on cloud. In the next, we classify security issues according to four categories of security, data security, logical security, physical security and administrative security. As Figure 2.1 illustrates.

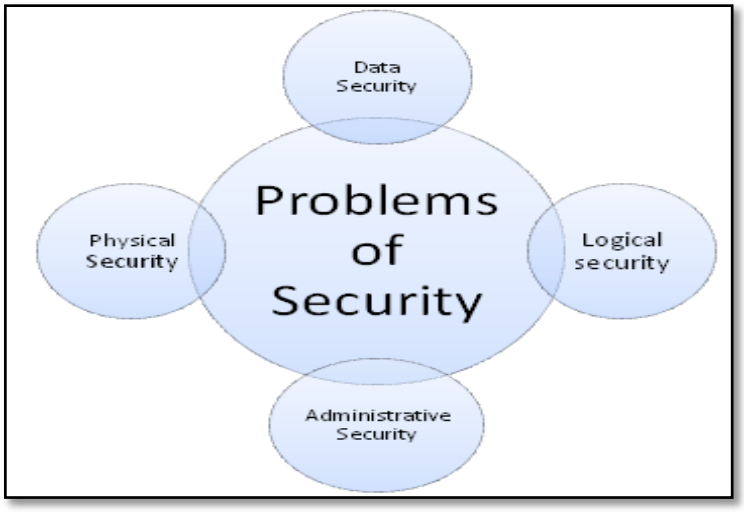

Figure 2.1: Different problems of Security

2.6.1 Data Security

Data security mainly refers to confidentiality, integrity and availability which are the major issue for the vendors. In a traditional on-premise application deployment model, the sensitive data of each enterprise continues to reside within the enterprise boundary and is subject to its physical, logical and personnel security and access control policies. However, in cloud, the enterprise data is stored

outside the enterprise boundary. Consequently, the cloud vendor must adopt additional security checks to ensure data security and prevent breaches.

The following key security elements should be carefully considered in order to ensure security of the enterprise:

- **Data Network Security**: In a cloud, sensitive data is obtained from the enterprises, processed by the SaaS application and stored at the SaaS vendor end. All data flow over the network needs to be secured in order to prevent leakage of sensitive information. This involves the use of strong network traffic encryption techniques such as Secure Socket Layer (SSL) and the Transport Layer Security (TLS) for security.

- **Data locality:** In a cloud environment, the consumers use the applications provided by the SaaS and process their business data. But in this scenario, the customer does not know where the data is getting stored. In many cases, this can be an issue. Due to compliance and data privacy laws in various countries, locality of data is of utmost importance in many enterprise architecture. For example, in many EU and South American countries, certain types of data cannot leave the country because it is considered potentially sensitive information

- **Data Integrity:** Data integrity is one of the most critical elements in any system. Data integrity is easily achieved in a standalone system with a single database. Next in the complexity chain are distributed systems. In a distributed system, there are multiple databases and multiple applications. In order to maintain data integrity in a distributed system, transactions across multiple data sources need to be handled correctly in a fail safe manner. This can be done using a central global transaction manger. Each application in the distributed system should be able to participate in the global transaction via a resource manager. The lack of integrity controls at the data level could result in profound problems. Architects and developers need to approach this danger cautiously.

- **Data access:** Data access issue is mainly related to security policies provided to the users while accessing the data. In a typical scenario, a small business organization can use a cloud provided by some other provider for carrying out its business processes. This organization will have its own security policies based on which each employee can have access to a particular set of data. The security policies may entitle some considerations wherein some of the employees are not given access to certain amount of data.

- **Data segregation:** Multi-tenancy is one of the major characteristics of cloud computing. As a result of multi-tenancy multiple users can store their data using the applications provided by SaaS. In such a situation, data of various users will reside at the same location. Intrusion of data of one user by another becomes possible in this environment. This intrusion can be done either by hacking through the loop holes in the application or by injecting client code into the SaaS system. A client can write a masked code and inject into the application. If the application executes this code without verification, then there is a high potential of intrusion into other's data.

We can find many works in this area like Arshad and al. [**26**], in their work showed various security issues, separating element by element. They began with problems related to data security from unauthorized access to data sources in an enterprise because the data is spread across different systems and they can be accessed by unauthorized persons. A.Parakh and al. [**27**] have shown that the traditional approach of security (explicit), whose data are stored on a single server and access to these data by a password, which is generally simple and memorable for most users, facilitated the attacks and intrusions on these data sources. However, Karkouda and al., [**28**] treated in their work the security of the data warehouses stored in the cloud. They showed that reliance on providers is difficult to build with the traditional architecture of the cloud based on a single provider. This architecture threatens the confidentiality of customer data since they are hosted by a single provider of external risk operate.

2.6.2 Logical Security

In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, operating systems, storage devices, and computer network resources.

Due to the cloud virtualization, cloud providers are residing the user's applications on virtual machines (VMs) within a shared infrastructure. The VMs are managed by hypervisor. As the hypervisor is main source of managing a virtualized cloud platform, hackers are targeting it to access the VMs and the physical hardware, because hypervisor resides between VMs and hardware, so attack on hypervisor can damage the VMs and hardware.

So, In order to maintain the security of users, providers are isolating the VMs from each other so if any of them is malicious, it will not affect the other VMs under the same provider. Several vendors such as Xen and KVM are providing strong security mechanisms of securing the cloud hypervisors, but still it is identified that sometimes security of VMs is compromised

Sajithabanu and al. [**29**], consider that virtualization is one of the main components of a cloud. But this poses major security risks ensuring that different instances running on the same physical machine are isolated from each other is a major task of virtualization which is not met completely in today's scenario. The other issue is the control of administrator on host and guest operating systems. Current VMMs (Virtual Machine Monitor) does not offer perfect isolation. Some vulnerability has been found in all virtualization software which can be exploited by malicious and local users to bypass certain security restrictions or gain privileges.

Bamiah and al. [**30**] speak about virtual machines (VMs) that are managed by hypervisor in order to provide virtual memory as well as CPU (central processing unit) scheduling Policies to virtual machines. As the main source of hypervisor is managing a virtualized cloud platform. Hackers are targeting it to access the virtual machine and the physical hardware, because hypervisor resides between virtual machine and hardware so attack on hypervisor can damage the VMs and hardware. In addition, co-location of multiple virtual machines increases the attack area and risk of virtual machine to compromised virtual machine. Intrusion

detection and prevention systems must be able to detect malicious activity at the level of virtual machines, regardless of the location of the virtual machine virtualized cloud within the environment.

2.6.3 Physical Security

Physical security is often overlooked (and its importance underestimated) in favor of more technical and dramatic issues such as hacking, viruses, Trojans, and spyware. However, breaches of physical security can be carried out with little or no technical knowledge on the part of an attacker. Moreover, accidents and natural disasters are a part of everyday life, and in the long term, are inevitable.

There are three main components to physical security. First, obstacles can be placed in the way of potential attackers and sites can be hardened against accidents and environmental disasters. Such measures can include multiple locks, fencing, walls, fireproof safes, and water sprinklers. Second, surveillance and notification systems can be put in place, such as lighting, heat sensors, smoke detectors, intrusion detectors, alarms, and cameras. Third, methods can be implemented to apprehend attackers and to recover quickly from accidents, fires, or natural disasters.

2.6.4 Administrative Security

We mean by administrative problems all cases that affect the type of provider and the type of contract, there are certain authors who have spoken on this kind of problems. Sajithabanu and al. [**29**], they think that there may be a case that some cloud providers are not the authorized provider. They may be duplication of a Web page that already exists in order to trick and entice users into giving private or financial particulars or their passwords.

2.7 Existing Solutions to Protect Cloud Computing

There are several works that propose solutions to some security issues that threaten cloud computing. These solutions are based on several techniques shown below.

We find solutions based on cryptography such as Bamiah and al. [**30**] who propose a method to build a trusted computing environment for cloud computing system by providing secure access cross platform into cloud computing system. The proposed Network consists of three backup sites for recovery after disaster. The backs up sites are located at remote location from the main server. If any one of the paths fails, it will use alternate path working. The encrypted file will be creating during back up sites and data's are compressed. The data will be decrypted during recovery operation. They proposed a cross-platform integration model by using secure communication via the internet and the utilization of a key for security.

However, Ruj and al. [**31**] propose a new model for data storage and access in clouds, their scheme avoids storing multiple encrypted copies of the same data. In this framework, cloud stores encrypted data (without being able to decrypt them) in order to secure data storage. The main novelty of this model is addition of key distribution centers (KDCs). They propose DACC (Distributed Access Control in Clouds) algorithm by employing attribute-based encryption, where one or more KDCs distribute keys to data owners and users.

Melvin and al. [**32**], survey different encryption schemes used in clouds. Many encryption schemes like Attribute Based Encryption (ABE),key-policy attribute based encryption (KP-ABE), Ciphertext-Policy ABE(CP-ABE), hierarchical identity based encryption (HIBE) are discussed in which all the schemes are concentrated on efficient access control.

Conversely, there are some works based on data partitioning scheme for example Parakh and al. [**27**] consider that the traditional approach to securing data is to store and back it up on a single server and allows the access upon the use of passwords that are needed to be frequently changed. However, there is a tendency among users to keep passwords simple and memorable leading to the possibility of brute force attacks. Therefore, they prefer to use an algorithm for online data storage and number theory.

The idea is to divide the data into K parts. This division is made with the separation algorithm to the data stored on servers later randomly. Data is

partitioned on different servers so they are implicitly secured and they do not need to be encrypted partitions because they do not show the same information.

Karkouda and al. [**28**] propose a way to protect data warehouses, to limit risks in cloud computing and to provide confidentiality and availability of data. The proposal is to split each data stored in the warehouse on several cloud providers through the sharing secret algorithm (Shamir 1979). Algorithm secret sharing shares the data tuples from several suppliers. The way of distributing data allows in one hand to store at each part of the provider information, they are then not understandable and not exploitable by a malicious user in the case of intrusion and secondly not to depend on one provider.

On the other hand, we find a number of works based on machine learning like Arshad and al. [**26**] focus on such challenge intrusion severity analysis. Particularly, they highlight the significance of intrusion severity analysis for the overall security of clouds. Additionally, they present a novel method to address this challenge in accordance with the specific requirements of clouds for intrusion severity analysis. They proposed to solve the severity problem by treating it as a problem of classification. Furthermore, machine learning techniques have been used to perform this classification. In this frame, the unsupervised learning techniques are usually more suitable for offline analysis as the classifications tend to change over the length of analysis data sets. The goal of a classifier is to build a model of class distribution in terms of the quantified characteristics of the constituent objects of the data set.

2.8 Conclusion

In this chapter, we present an overview of cloud computing which became used from individuals and institutions alike after we underscored the most important problems in cloud, the security. There are several threats tire protection of cloud and tracts the trust between the user and the provider including unauthorized access, data loss, and poor use of services provided by the cloud. To make this technology more secure, researchers have proposed several security solutions that are based on several ways as cryptography and other.

As we mentioned earlier, the cloud has several security problems particularly it is accessible by "anyone" who exposes it to several threats such as unauthorized access to data sources, and intrusion. In the next chapter, we propose our security model based on ORBAC model and cryptographic mechanism.

Chapter 3:

CP ORBAC to Secure Access Data on Cloud Computing

3.1 Introduction

Although Cloud computing has been developed to reduce Information Technology (IT) expenses, and to provide agile IT services to individual users as well as organizations. It moves computing and data away from desktop and laptop computer into large data centers. However, this technology gives the opportunity for more innovation in lightweight smart devices, and it forms an innovative method of performing business. Various malicious activities from illegal users threaten this technology such as inflexible access control which is generally a policy or procedure that allows, denies or restricts access to a system. Access control policies define the subjects and the permissions in a computer system to enforce the security of an organization.

In this chapter, we give an overview about some works in relation to access models and Ciphertext Policy Attribute-Based Encryption. After, we will describe an approach which proposes a security model bases on Organization Role Based Access Control (ORBAC) and encryption system. This model aims to give users a possibility to control security of their data.

Finally, we present an experimental study to show the importance of our proposition and we will provide a comparative study between our approach and some access control models.

3.2 Related Works

In this part we show some works that speak about the following topic. In the first, we talk about the integration of access models on some proposition. In the next, we illustrate a number of papers that use cryptographic mechanisms. More especially, the employ of ciphertext attribute based encryption.

3.2.1 Integration of an Access Model

In the work of Singh and al. [**33**], present a new extended architecture of RBAC which can resolve the security issues and data loss issues by using restriction policy on number of roles, number of users per role and number of transaction per day/hour/user. They create a new extended architecture of RBAC system which a kind of Ontology which can keep backup of the data which is getting send to the cloud server and to restrict the number of users per role .

Ruj and al. [**31**], propose a new model for data storage and access in clouds. Their scheme avoids storing multiple encrypted copies of same data. The main novelty of this model is addition of key distribution centers (KDCs).
They offer DACC (Distributed Access Control in Clouds) algorithm by employing attribute-based encryption, where one or more KDCs distribute keys to data owners and users. KDC may provide access to particular files in all records.
DACC is an access control mechanism, where owners decide on attributes that users should have and users receive decryption keys which enable them to access records which they are authorized to access thus, a single key replaces separate keys from owners. Owners and users are assigned certain set of attributes. Owner encrypts the data with the attributes it has and stores them in the cloud. The users with matching set of attributes can retrieve the data from the cloud.

Khan and al. [**34**], consider that the cloud environment is a large open distributed system and it is important to preserve the data, as well as, privacy of users.

They think that Access Control methods ensure that authorized users access the data and the system. In their paper, they discuss various features of attribute based access control (ABAC) mechanism, suitable for cloud computing environment. ABAC provides policies for sensitivity of credentials.

It allows organization to maintain their autonomy while collaborating efficiently. In addition, it provides an automated trust negotiation, which is auditable as and when that capability is required. It leads to the design of attribute based access control mechanism for cloud computing.

In the same way, Cao and **al**. [**35**], offer a temporal access control solution along with a proxy-based re-encryption mechanism for cloud computing.

The proposed scheme is originated from the needs of practical cloud applications, in which each outsource can be associated with an access policy on a set of temporal attributes, e.g., period-of-validity, opening hours, or hours of service. Each user can also be assigned a license with several privileges based on the comparative attributes.

To enforce the valid matches between access policies, user's privileges time. This design brings about several efficient benefits, such as flexibility, supervisory, and privacy protection, compared with prior works. This solution also addresses another practical issue to implement cryptographic integer comparisons and encryption mechanism.

3.2.2 Ciphertext Policy Attribute-Based Encryption (CP-ABE)

The identity-based encryption (IBE) was proposed by Shamir and al. [**36**]. In an IBE system, an authority distributes keys to users with associated identities, and messages are encrypted directly to identities. Then, the secure schemes in the standard model were selectively constructed by Shamir and al [**36**], Boneh and al. [**37**] Canetti and al. [**38**] give an IBE system and security proof that moved beyond the connection of the partitioning strategy, but at the cost of a large and complicated complexity assumption.

Hierarchical identity-based encryption (HIBE) , Gentry and al. [**39**] expand the functionality of IBE to include a hierarchical structure on identities where identities can delegate secret keys to their subordinate identities. Since the introduction of attribute-based encryption (ABE) in implementing fine-grained access control systems, a lot of works have been proposed to design flexible ABE schemes. There are two methods to realize the fine-grained access control based on ABE: key policy attribute-based encryption (KP-AB) and CP-ABE. They were both mentioned by Lyn and al. [**40**] .

In KP-ABE, each attribute private key is associated with an access structure that specifies which type of ciphertexts the key is able to decrypt, and ciphertext is labeled with sets of attributes.
CP-ABE is complementary where the attributes are used to describe user's credentials, and the formulas over them are attached to ciphertext by encrypting party. As the CP-ABE is conceptually closer to the traditional access control methods, a ciphertext policy attribute-based encryption (CP-ABE) system consists of four fundamental algorithms: Setup, Encrypt, KeyGen and Decrypt. Attributes set used to generate secret key.

However, Lewko and al. [**41**] present the Time proxy re-encryption (PRE) scheme to achieve fine-grained access control and scalable user revocation in a cloud environment. The scheme enables each user's access right to be effective in a pre-determined period of time, and enables the cloud service provider (CSP) tore-encrypt cipher texts automatically, based on its own time. Thus, the data owner can be offline in the Process of user revocations. Their approach is to provide Scalability, Fine-grained access, data confidentiality and cost efficiency.

3.3 Security Model Based CP ORBAC

This model implements a management system of access to data sources stored in a cloud based on Organization Role Base Access Control model (ORBAC) and Encryption.

The access control based on roles is insufficient to satisfy all our needs in terms of protection. One of the major problems of this model is the fact that all users associated with the same role necessarily have the same privileges. This reduces the flexibility of security policies.

In fact, the expression of contextual aspects related to permissions is not present in the RBAC model. There is confusion between the role and organization. We can only express permissions which it causes a complex exception management. ORBAC gives a possibility to specify a mix political. It allows controlling access to data, the use of this data specifies some permissions and it permits to manage different situations thank to the use of a Context.

To get better the security of data in this model, we include CP-ABE. This solution allows to control access of users due to access structure proposed by data owner. This formula defines that the only authorized users are allowed to access the data sources after decryption of data because data will be encrypted by assigning for each resource a secret key.

In this work, we provide the first construction of a ciphertext-policy attribute-based encryption (CP-ABE). In our system, a data owner's private key will be associated with a number of attributes based on ORBAC. On the other hand, when a party encrypts a message in our system, they specify an associated access structure over attributes. A user will only be able to decrypt a ciphertext if that user's attributes pass through the ciphertext's access structure. At a mathematical level, access structures in our system are described by a $\mathrm{tuple}(r, a, s, c)$, where r is the role, a is the authorized actions, s is a resource and c is a context.The figure 3.1 shows our security model which consists of the following system entities:

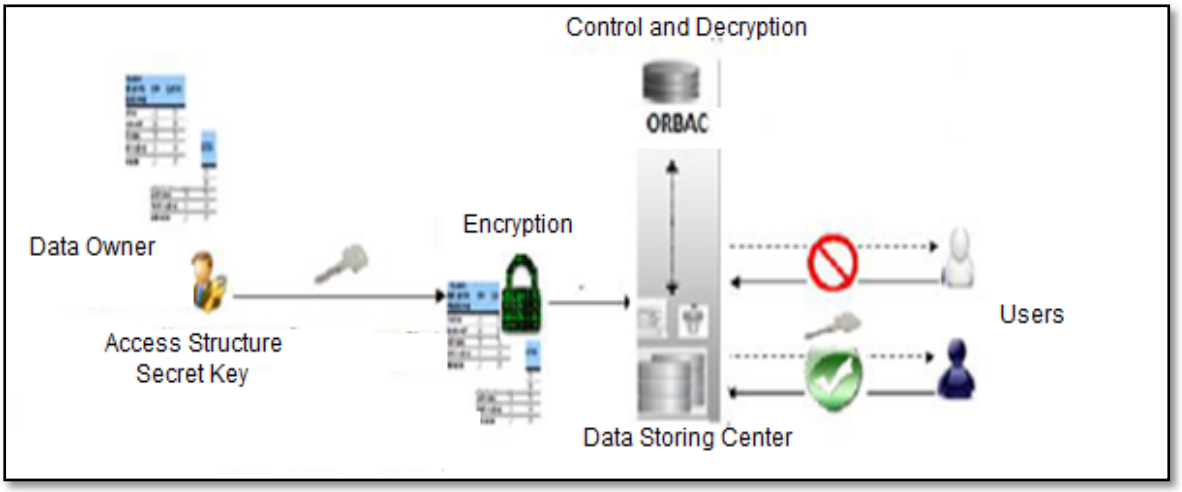

Figure 3.1: Security model based CP-ORBAC Encryption

- **Data owner:** proposes an access structure to define authorized users where an Access structure is a $\text{tuple}(r, a, s, c), r$ is the role, a is the authorized actions,s is a resource and c is the request contextual information after he encrypts their data and stores it in data center storage to share on the cloud and they decrypt it. Moreover, each data consumer is administrated by a Control System.

- **Data storing center:** It is an entity that provides a data sharing service.

- **Control and encryption system:** It is a responsible entity to control access to data and encryption. First it gives an authorisation response as a tuple $\text{is\_permitted}(r, a, s)$ where r is a role, a is an action and s is resource. After the system encrypts data and generates a secret key to data owner and user.It use the following algorithms :

A ciphertext-policy attribute based encryption scheme consists of three fundamental algorithms: Setup, Encrypt, and Decrypt. The next figure 3.2 shows different steps of our algorithm , as shown in Figure 3.2.

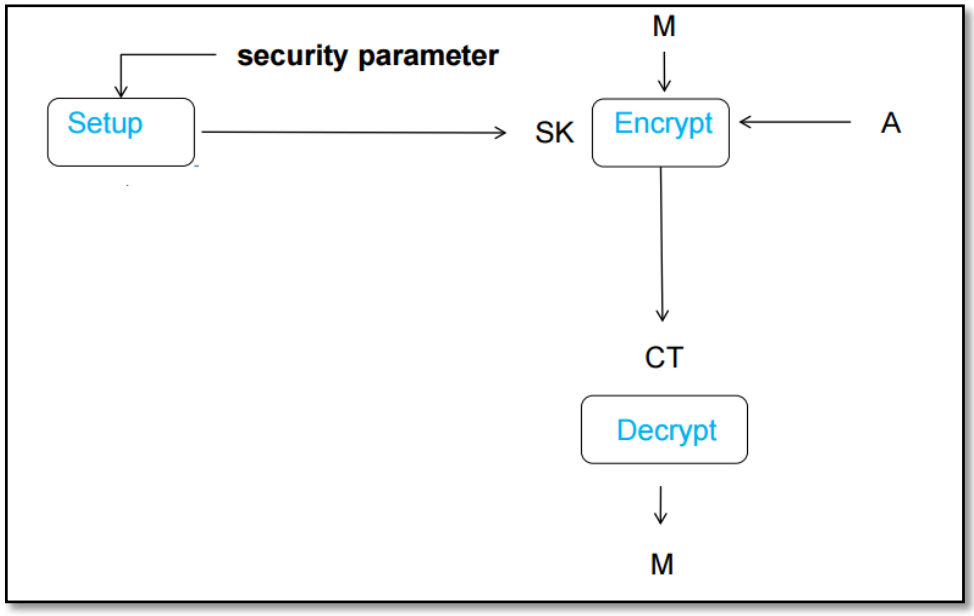

Figure 3.2: An Illustration represents an Encryption System

- $\boldsymbol{Setup}$: this is a randomized algorithm that takes security parameters.
- $\boldsymbol{Encrypt}\ (\boldsymbol{SK}, \boldsymbol{A}, \boldsymbol{M})$ **:** this is a randomized algorithm that takes as input a message M, a set of attributes and a secret key SK proposed by data owner. It outputs the ciphertext CT .
  The algorithm will encrypt M and produce a ciphertext CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message.
- $\boldsymbol{Decrypt}\ (\boldsymbol{CT}, \boldsymbol{SK})$: The decryption algorithm takes as input a ciphertext CT that was encrypted under the set of attributes A, and a private key SK, which is a private key for a set A of attributes. If the set A of attributes satisfies the access structure then the algorithm will decrypt the ciphertext and return a message$M$.

3.4 Experiments

To validate our approach, we have developed a prototype in Java and conducted an experimental study. Our prototype is installed on virtual network.

3.4.1 Experimental Setup

To evaluate our proposal based on ORBAC, we propose to compare it with an application based on RBAC which is usually used but it does not introduce the concept of context. We decided to use both models evaluation measures recall and precision.

3.4.2 Test Data

We applied our model on an organization = “hospital” and a collection of about 200 documents which are medical files. However, we consider a query where a doctor wants to access to medical files in a case of emergency System returns a list of 100 that contains 30 documents with context=“emergency” and 70 documents with context = “treating” and we will assign actions (read, modify, delete, print, download) to the following roles (emergency doctor, treating doctor).

On the other hand, we applied both models to other roles (nurse, administrative agent) which belong to the hospital organization that have no access to resources and the result was the same that “don’t access”.

3.4.3 Comparison between ORBAC and RBAC

We propose to compare it with an application based on RBAC which is usually used but it does not introduce the concept of context.

Recall and precision measures will define the capacity of the access selection of both models to data sources as follows:

In our case, the recall is defined by the number of authorized access to resources in context c in terms of number of relevant resources in context c that owns the database. This means when a user queries the database, system must show all documents in context c this signifies that our system does not allow user to access to all documents proposed by a database.

However, the precision is the number of authorized access to resources in context “c" , found in relation to number of total resources proposed by data base. The principle is: when a user queries a database, he wants access to the documents

proposed in response to his context c. If precision is high, it means that some unnecessary documents are proposed by system and it signifies some unauthorized accesses are realized on our resources.

The next formulas show our evaluation measures:

$$Recall = \frac{number\ of\ authorized\ accessto\ resources\ in\ context\ c}{number\ of\ resources\ in\ context\ c} \ldots\ldots\ldots\ldots (3.1)$$

$$Precision = \frac{number\ \ of\ \ aut\,horized\ \ accessto\ \ resources\ \ in\ context\ \ c}{number\ \ of\ resources} \ldots\ldots\ldots . (3.2)$$

$$FMeasure = \frac{2 * Recall\ Precision}{Recall + Precision} \ldots\ldots\ldots\ldots\ldots\ldots . . \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . \ldots (3.3)$$

3.5 Results

The result is explored in next tables:

Files	Recall ORBAC	Recall RBAC	Precision ORBAC	Precision RBAC	F-Meas. ORBAC
5	0.167	0.05	0.025	0.150	0.075
10	0.333	0.100	0.050	0.300	0.150
15	0.500	0.150	0.075	0.450	0.225
20	0.667	0.200	0.100	0.600	0.300
25	0.833	0.250	0.125	0.750	0.375
30	1.000	0.300	0.150	0.900	0.450

Table 3.1: Results of Recall, Precision and F measure to ORBAC and RBAC models where c=”Emergency”

Files	Recall ORBAC	Recall RBAC	Precision ORBAC	Precision RBAC	F-Meas. ORBAC
5	0.071	0.050	0.025	0.150	0.075
10	0.143	0.100	0.050	0.300	0.150
15	0.214	0.150	0.075	0.450	0.225
20	0.286	0.200	0.100	0.600	0.300
25	0.357	0.250	0.125	0.750	0.375
30	0.429	0.300	0.150	0.900	0.450

Table 3.2: Results of Recall, Precision and F measure to ORBAC and RBAC models where c=”Treating”

The next histograms make clear our table’s results:

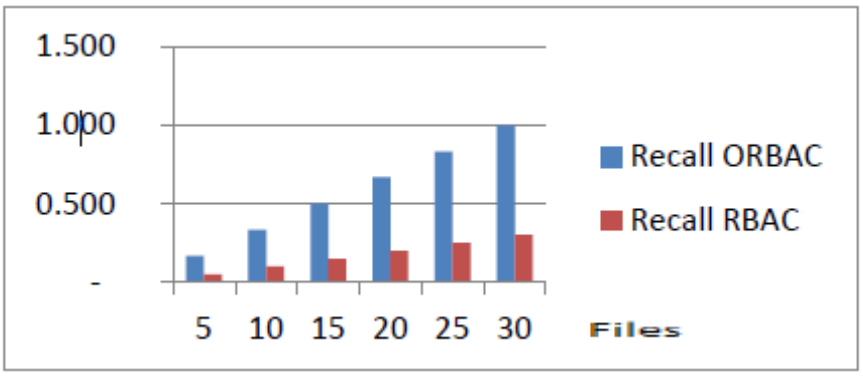

Figure 3.3: Histogram represents different between Recall ORBAC and Recall RBAC where context=”Emergency”

When comparing results of ORBAC model with those of RBAC model, we note that the use of the model ORBAC increases a protection of resources against random use by categorizing them in context and as a result the number of authorized access is increasing against RBAC model. It requests more resources

because it does not take into account the context of the role which will allow these resources to be exploited by unauthorized roles.

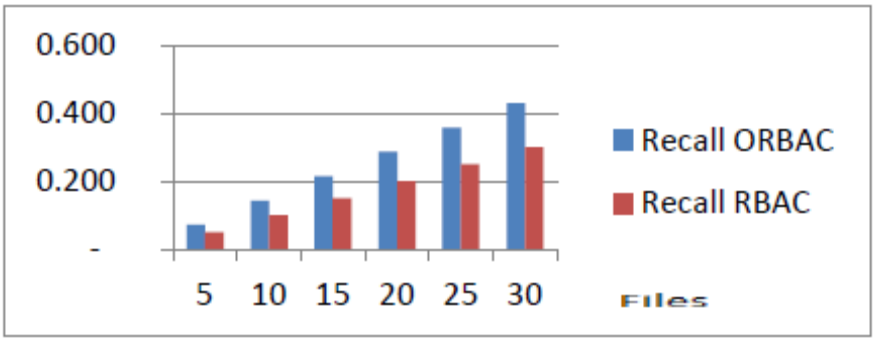

Figure 3.4: Histogram represents different between Recall ORBAC and Recall RBAC where context=”Treating”

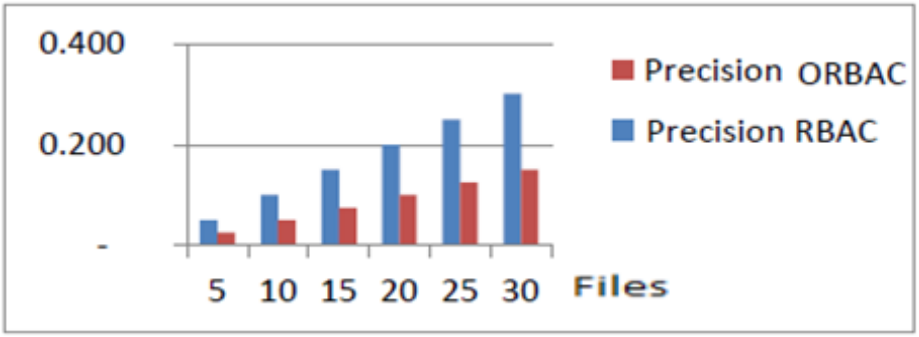

Figure 3.5: Histogram represents different between Precision ORBAC and Precision RBAC where context=”emergency”

We notice that the recall of system based on RBAC model is higher compared to the system based on ORBAC model which signifies that our system selects access more correctly and precisely by eliminating more unauthorized access.

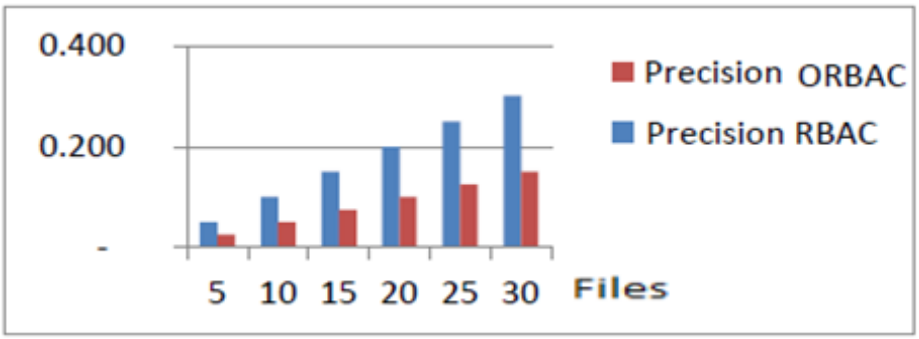

Figure 3.6: Histogram represents different between Precision ORBAC and Precision RBAC where context=”Treating”

We notice that in the two contexts, the precision of system based on the model RBAC is higher compared with the system based on the model ORBAC which signifies that the model based on the model RBAC selects fewer documents to users and allows to reduce unauthorized accesses to documents, thus it allows to increase the confidentiality of documents not concerned by the given contexts.

3.6 Comparison

The Table 3.3 summarizes a comparative analysis of the RBAC, ORBAC, and CP ORBAC. This table shows that our proposal includes the advantages of ORBAC model and the advantages of CP ABE :

- It permits to mange different situations thanks to the notion of Context.
- It allows to data owner to define Access Rights (propagation, revocation) due to an Access Structure.
- It provides a possibility to encrypt data and to increase a trust between data owner and provider.
-

	RBAC	ORBAC	CP ORBAC
Authentication	Server	Server	Server
Review of Access Rights	Server	Server	User/Server
Access Rights Propagation	Server	Server	User/Server
Access Rights Revocation	Server	Server	User/Server
Dynamic	No	Yes	Yes
Implementation	Latice Model	Latice Model	Latice Model

Table 3.3: Comparison between RBAC, ORBA and CP ORBAC .

3.7 Conclusion

In this chapter, we proposed an approach that aims to protect data stored in the cloud by offering a new layer between different users and the cloud. This layer will allow the property owner to manage data access and security of these data but we note that our approach does not give the details about security rules. To fill this gap, we think to use a model driven architecture (MDA) architecture in which security models are embedded in and scattered throughout the high level system models, which are transformed until their final implementation according to MDA strategy.

Chapter 4:

An MDA Approach to Design Security of Access to Data on Cloud

4.1. Introduction

In this chapter, we propose a methodological approach for the model driven development of secure data on cloud. This proposal is within a model driven methodology for the development of security access system based on the Model Driven Architecture (MDA) proposed by the Object Management Group (OMG).
Therefore, rather than considering security once the system has been completely built, we believe that security and privacy measures should be integrated in all layers of design, from the early stages of its development as another relevant requirement, meaning that much more robust, secure and platform independent products will be produced [**35**].

In order to develop our model considering confidentiality issues in the whole development process, from an early development stage to the final implementation, our proposal has been aligned with a model driven architecture (MDA) in which security models are embedded in and scattered throughout the high level system models, which are transformed until their final implementation according to the MDA strategy.
We use the data conceptual model as a Platform Independent Model (PIM) and the security rules as a Platform Specific Model PSM, such models will be modified,

so as to be able to add security aspects if the stored information is considered as critical. On the one hand, the use of a UML to incorporate security aspects at the conceptual level (PIM) is proposed; on the other, security rules will be presented using a logical structure that helps to reason about permissions, prohibitions and obligations.

To get a better security of data in this model, we include CP-ABE. This solution allows to control access of users due to access structure proposed by data owner. This formula defines that the only authorized users are allowed to access the data sources after decryption of data because data will be encrypted by assigning for each resource a secret key.

4.2. Model Driven Archiecture (MDA)

MDA specifies three default models of a system corresponding to the three MDA viewpoints. These models can perhaps more accurately be described as layers of abstraction, since within each of these three layers a set of models can be constructed, each one corresponding to a more focused viewpoint of the system (user interface, information, engineering, architecture, etc.). In the picture 4.1 below we can see how different models interact:

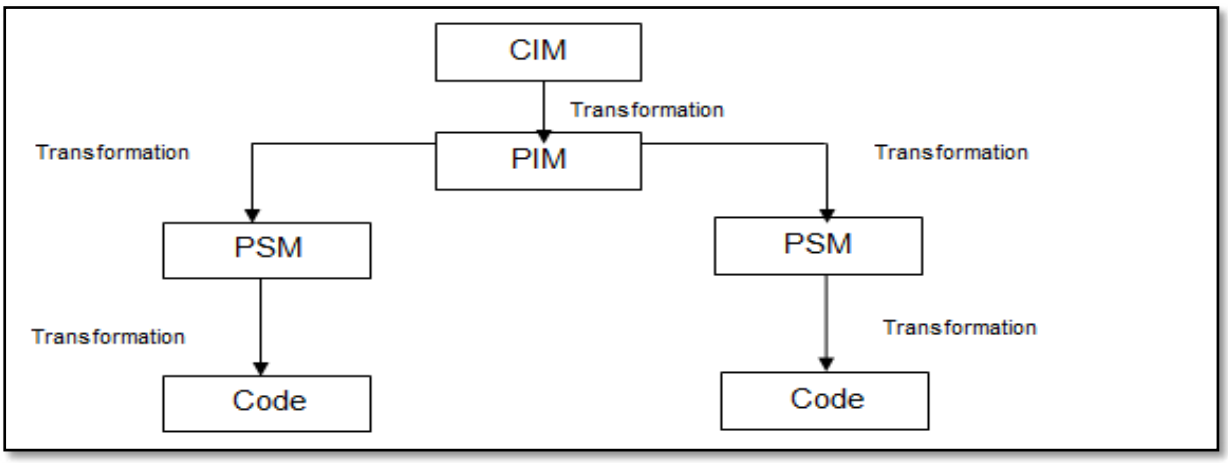

Figure 4.1: A Scheme represents MDA architecture.

4.2.1 Computation Independent Model (CIM)

A CIM is also often referred to as a business or domain model. It presents exactly what the system is expected to do, but hides all information technology related specifications to remain independent of how that system will be (or currently is) implemented.

The CIM plays an important role in bridging the gap which typically exists between these domain experts and the information technologists responsible for implementing the system.

In an MDA specification the CIM requirements should be traceable to the PIM and PSM constructs that implement them (and vice-versa).

4.2.2 Platform Independent Model (PIM)

A PIM exhibits a sufficient degree of independence so as to enable its mapping to one or more platforms. This is commonly achieved by defining a set of services in a way that abstracts out technical details. Other models then specify a realization of these services in a platform specific manner [**42**].

4.2.3 Platform Specific Model (PSM)

A PSM combines the specifications in the PIM with the details required to stipulate how a system uses a particular type of platform. If the PSM does not include all of the details necessary to produce an implementation of that platform it is considered abstract.

4.3. Related Works

In this section, we show some works where the authors think that security is a concept that can be formalized.

In the work of Véla and al. [**2**], propose a methodological approach for the model driven development of secure XML databases (DB). This proposal is within the framework of MIDAS, a model driven methodology for the development of Web Information Systems based on the Model Driven Architecture (MDA) proposed by the Object Management Group (OMG).

The XML DB development process in MIDAS proposes using the data conceptual model as a Platform Independent Model (PIM) and the XML Schema model as a Platform Specific Model (PSM), with both of these represented in UML. In this work, such models will be modified; so as to be able to add security aspects if the stored information is considered as critical. On the one hand, the use of a UML extension to incorporate security aspects at the conceptual level of secure DB development (PIM) is proposed; on the other, the previously-defined XML schema profile will be modified, the purpose being to incorporate security aspects at the logical level of the secure XML DB development (PSM).

.

in the same way, Véla and al. [**43**], offer an approach for the model driven development of Secure XML DWs because authors have remarked that security issues have been considered during the whole development process of traditional Data warehouses (DWs), current research lacks approaches with which to consider security when the target platform is based on the Web and XML technologies.

In order to tackle this situation, they propose a methodological approach based on the Model Driven Architecture (MDA) for the development of Secure XML DWs.

They therefore specify a set of transformation rules that are able to automatically generate not only the corresponding XML structure of the DW from secure conceptual DW models, but also the security rules specified within the DW XML structure, thus allowing us to implement both aspects simultaneously. A case study is provided at the end of the paper to show the benefits of this approach.

By the same token, Véla and al. [**44**], provide another approach to the model driven development of Secure XML Data Warehouse (DW). This approach begins by defining the secure conceptual MD model (PIM) represented by means of the secure UML profile, independently of the target logical MD model.

This PIM is transformed into a secure XML DW, as a logical model (PSM), by applying Model to Model (M2M) Transformations. They have first defined the rationale behind these transformation rules and how they have been developed in natural language, and they have then established them clearly and formally by using the Query/View/Transformation (QVT) language. QVT is a standard language consists of two parts: The declarative and the imperative parts.
The declarative part provides mechanisms to define relations that must hold between the model elements of a set of candidate models (source andtarget models). A set of these relations (or transformation rules) defines a transformation between models. The imperative part: defines operational mappings that extend the declarative part with imperative implementations when it is difficult to provide a purely declarative specification of a relation.

Farhan and al. [**45**], use the Model Driven Architecture (MDA) approach to represent logical model requirements for secure Temporal Data Warehouses (TDW). They employ the Platform Independent Model (PIM) which does not include information about specific platforms and technologies. They consider that the most crucial issue in MDA is the transformation between a PIM and Platform Specific Models PSM. Thus, OMG defines use the Query/View/Transformation (QVT) language, an approach for expressing these MDA transformations. This paper proposes a set of rules to transform PIM model for secure temporal data warehouse (TDW) to PSM model.

4.4. A Secure Computation Independent Model (SCIM)

The CIM level presents exactly what the system is expected to do, but hides all it related to specifications to remain independent of how that the system will be. The requirements should be traceable to the PIM and platform specific model (PSM) constructs that implement them [**42**].

The next figure 4.2 shows our CIM model which consists of the following system entities:

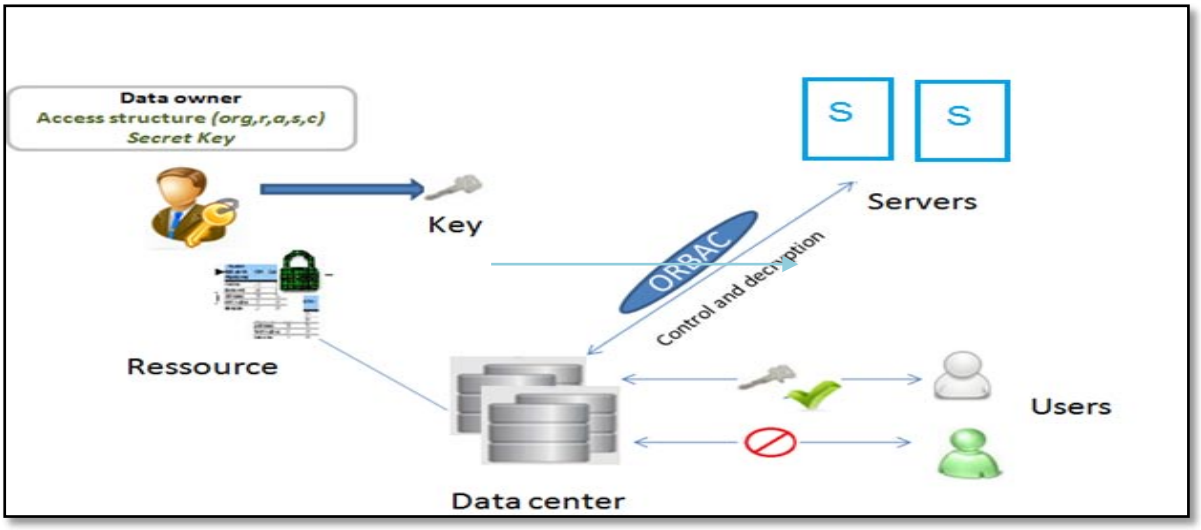

Figure 4.2: An illustration represents a CIM model

- **Data owner** proposes an access structure to define authorized users where an access structure is a tuple **(r, a, s, c), r** is the role, a is the authorized actions, s is a resource and c is the request contextual information after he encrypts their data and stores it in data centre storage to share on the cloud and they decrypt it. Moreover, each data consumer is administrated by a control system.

Our access structure is defined as following [**46**]:

Definition 5.1: Let $\{P_{1,\dots\dots}P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P_{1,\dots}Pn\}}$.. is monotone if $\forall B, C$: if $B \in A$ and $B \subseteq C$, ,then $C \in A$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection). A of non empty subsets of $\{P_{1,\dots\dots}P_n\}$. i.e., $A \subseteq 2^{\{P_{1,\dots}Pn\}} \setminus \{\ \}$. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets.

In our case, attributes will play the role of parties and we will only deal with monotone access structure.

- **User:** It is an entity who wants to access the data. If a user has a set of roles satisfying the access policy of the encrypted data, and it is not revoked in any valid attribute groups, then he will be able to decrypt the ciphertext and obtain the data.
- **Data storing center**: It is an entity that provides a data sharing service.

- **Control and encryption system**: It is a responsible entity to control access to data and encryption. First it gives an authorization response as a tuple $is\_permitted\ (r, a, s)$ where $\boldsymbol{r}$ is a role, a is an action and s is resource. After encryption, system encrypts data and generates a secret key to data owner and user. In this step, the secret key is partitioned into two or more pieces and stored at randomly chosen places on the network that are known only to the data owner. Our system retrieves the partitions to decrypt data if and only if the rule is checked.

The encryption scheme used by our system consists of three algorithms:

– $\boldsymbol{Setup}$**:** this is a randomized algorithm that takes no input other than the implicit security parameter.

- $\boldsymbol{Encryption}$**:** this is a randomized algorithm that takes as input a message M , a set of attributes and a secret key SK . It outputs the ciphertext T .
- $\boldsymbol{Decryption}$: this algorithm takes as input the ciphertext E that was encrypted under the set of attributes, the decryption key SK for access control structure. It outputs the message$M$.

4.5. A Secure Platform Independent Model (SPIM)

To develop a secure data PIM, a secure UML class diagram has been developed. At the PIM level, the secure data conceptual model is carried out without considering the selected technology, since this model is platform independent. This secure data PIM is represented through an extended UML class diagram, so as to be able to represent security aspects.

In its place of defining a new modeling language, we propose the use of UML, a widely accepted Object Oriented (OO) modeling that unifies the methods most used around the world. UML combines elements from the three major OO design methods: Rumbaugh's Object Modeling Technique (OMT) modeling **[**47**]**; Booch's OO Analysis and Design **[**48**]**, and Jacobson's Objectory **[**49**]**.

We think that the use of UML as the modeling language of our approach is the best choice. This option can be justified by the side of three considerations **[**50**]**.

- **UML** follows the **OO** paradigm, which has been proved to be semantically richer than other paradigms because **OO** models are closer to the user conception **[**51**]**.

- **UML** is a standard of the object Management Group (OMG) and unifies many years of effort in **OO** analysis and design **[**52**]** .

- **UML** has been widely accepted by the scientific and industrial communication.

By using our class diagram (Figure 4.3) and the relationships between them, we can define security policies applied to a particular organization.

A security policy regulates access to sources through permissions, prohibitions and obligations. We deal only with permission considering that the same reasoning applies to prohibitions and obligations. Permission class allows an organization to specify the permissions granted to a role in a context. More precisely, if an organization noted org, a role noted r, an action noted a, a resource noted s and a context noted c then $Permission\ (org, r, a, s, c)$ means that the organization org grants to the role r a permission to realize an action a on a resource s in context c.

In the UML class diagram, we add the ORBAC classes; we consider that each system model corresponds to one organization; hence all the classes will belong to this “organization”.

We use Role, Action and Resource classes to represent roles, activities and views respectively. These elements correspond to the ORBAC model. Moreover, for the ORBAC concrete classes, we can use the class user and Data Owner to represent subjects.

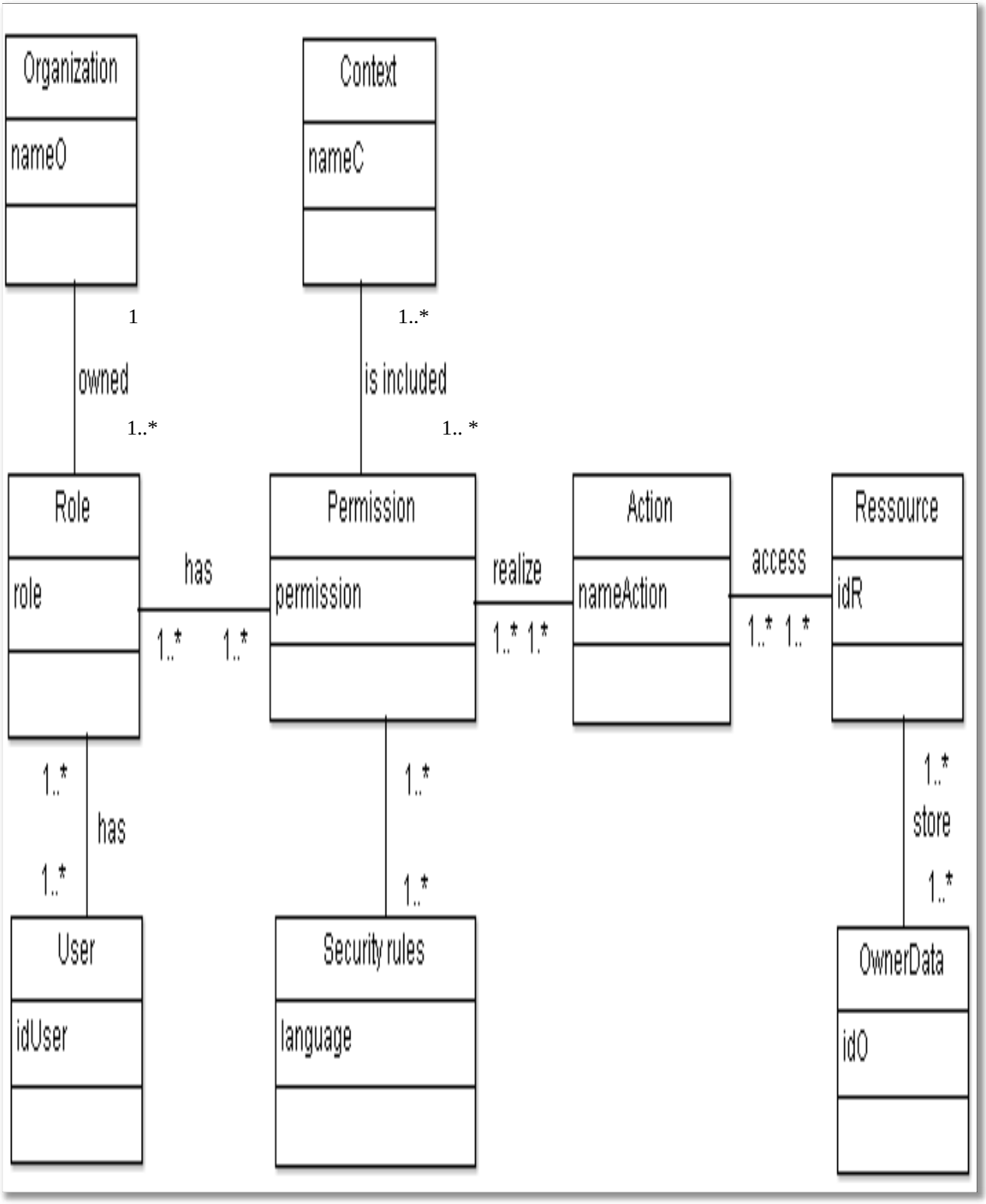

Figure 4.3: Class diagram represents a SPIM

Related to the ORBAC model, contexts are attached to permissions for us i.e. permissions are authorizations to perform some action on the system in a particular context. To represent contexts, we add the class **Context** and the relation **is included** to associate context to class **Permission.** For example, in a medical system, the permission of a doctor is to care patients and the context attached to it is that the patient is under the responsibility of the doctor.

On the other hand, we create class **Security rules**, we add relation **has** attached to each context a one or a set of permissions according to roles that the play within an organization. This scenario is for a usual ambiance (context).

However, the scenario in unusual contexts can also be part of the organization's duty.

For example, in the medical system, assistants can be in charge of managing patients' appointments in a usual context. Assistants can also be in charge of consulting the stock of medicine in an unusual context such emergency**.**

We give you an author example defined in [**53**] the security policy of the hospital $H1$ may include the following:

$\text{Permission}\,(\text{H1}, \text{doctor}, \text{medical file}, \text{emergency})$ means that "The hospital $H1$ gives doctors a permission to use any medical file in the context of "emergency" and $\text{Permission}\ (\text{H1}, \text{doctor}, \text{medical file}, \text{treating doctor})$ which means that "The hospital $H1$ gives doctors permission to access medical files of their patients" [**54**].

4.6. A Secure Platform specific model (SPSM)

A PSM combines the specifications in the PIM with the details required to stipulate how a system uses a particular type of platform.

4.6.1 A logical structure

The Unified Modeling Language, UML, has gained increased popularity in recent years. The success of UML can to a large extent be attributed to two factors.

First, UML has received extensive industry support from IT suppliers as well as users and has been effectively standardized.

Secondly, UML makes use of intuitive and visual modeling constructs as the main components of the language, which facilitates its adoption among large user groups. However, this reliance on graphical constructs poses problems when it comes to precise and unambiguous semantics. The constructs of UML are typically informally defined, which leaves room for ambiguities, loose interpretations and misunderstandings.

The purpose is to present a logical structure that helps to reason about permissions, prohibitions and obligations.

The objective is to combine a first-order language with a class diagram described previously. Axioms of a first-order theory are usually divided into logical axioms and own axioms. The logical axioms provide the basis for proving all the theorems of classical first-order logic while own axioms match special rules.

ORBAC presents tow levels:

- **Abstract level:** the security administrator defines security rules through abstract entities (roles, activities) without worrying about how each organization implements these entities.
- **Concrete level:** when a user requests an access, authorizations are granted to him according to the concerned rules, the organization, the role currently played by the user, the requested action on the resource and the current context. The derivation of permission can be formally expressed as follows:

$\forall$ **org** $\in$ **Organization**, $\forall$ **u** $\in$ **Subject**, $\forall$**a** $\in$ **Action**, $\forall$ **a** $\in$ **Action**, $\forall$ **r** $\in$ **Role**, $\forall$**c** $\in$ **Context** , $\forall$**s** $\in$ **resource**

$Permission\ (org, r, a, s, c) \wedge$ // *a security rule in its abstract form*

$Empower\ (org, u, r) \wedge$ // *in org , the role r is played by a subject u*

$Consider\ (org, t, s) \wedge$ // *in org , the action t corresponds to an object s*

$use\ (org, s, c) \wedge$ // *in org, the object s corresponds to a context c*

$Hold(org, u, a, r, c)$ // *in org, the context c is true for u, t and s*

$\rightarrow is\ permitted\ (u, t, s)$ // *runtime decision allowing u carrying out t on s*

This rule means:

"if the organization org, in the context c, grants permission to the role r to realize the action a on the resource s, if org empowers subject u in role r, if org uses the resource s in the context c, if within the organization org context c is true between r, a and s, then the role has a permission to perform the action on a resource. Based on defined rules, a decision is inferred for $\boldsymbol{is\_permitted}\,(r, a, s)$. As we have indicated previously, in the case that the user is specified by the role r is authorized to access to the data source s.

4.7. Experiments

To validate our approach, we have developed a prototype in Java and conducted an experimental study. Our prototype is installed on virtual network.
We use a last test data presented on chapter 4 but we increase a number of documents to 1000 documents.

We apply our model on an $organization = 'hospital'$ and a collection of about 1000 documents which are medical files. However, we consider a query where a doctor wants to access to the medical files in a case of emergency system returns a list of 600 documents contains 200 documents with $context = 'emergency'$ and 400 documents with context $= 'treating'$ and we will assign $actions\ (read, modify, delete, print, download)$
to the following roles $(emergency\ doctor, treating\ doctor)$.
On the other hand, we apply both models to other $roles\ (nurse, administrative\ agent)$ which belongs to the hospital organization that have no access to resources and the result was the same that $'don't\ access'$.
Recall and precision measures define the efficiency of our proposition and we use formulas (3.1) and (3.2).
For example:

"If a doctor wants 30 medical files of a database with the context $c = "Emergency"$, the Result is:

- ORBAC model :

$$Recall = \frac{30}{200} = 0.15 \qquad Precision = \frac{30}{600} = 0.05$$

- RBAC model:

$$Recall = \frac{30}{600} = 0.15 \qquad Precision = \frac{30}{1000} = 0.03$$

The Results are presented on the next table:

Files	Recall ORBAC	Recall RBAC	Precision ORBAC	Precision RBAC	F-Meas. ORBAC	F-meas. RBAC
10	0.050	0.017	0.017	0.050	0.050	0.030
15	0.075	0.025	0.025	0.075	0.075	0.045
20	0.100	0.033	0.033	0.100	0.100	0.060
25	0.125	0.042	0.042	0.125	0.125	0.075
30	0.150	0.050	0.050	0.150	0.150	0.090

Table 4.1: Results of recall, precision and F measure to ORBAC and RBAC models where context =" emergency"

The next histograms make clear our table's results:

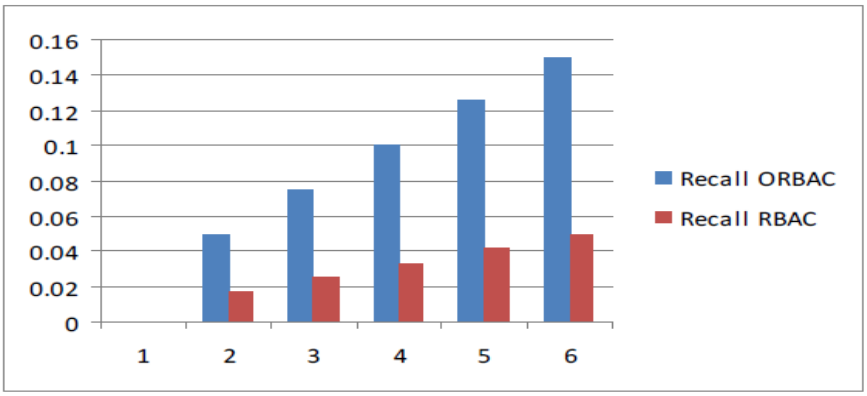

Figure 4.4: Histogram represents a difference between Precision ORBAC and Precision RBAC where context = 'emergency'.

When comparing results of ORBAC model with those of RBAC model (Figure 4.4), we note that the use of the model ORBAC increases a protection of resources against a random use by categorizing them in context and as a result the number of authorized access is increasing against RBAC model. We notice that the recall of system based on ORBAC model is higher compared with the system based on RBAC model which signifies that our system selects access more correctly and precisely by eliminating more unauthorized access.

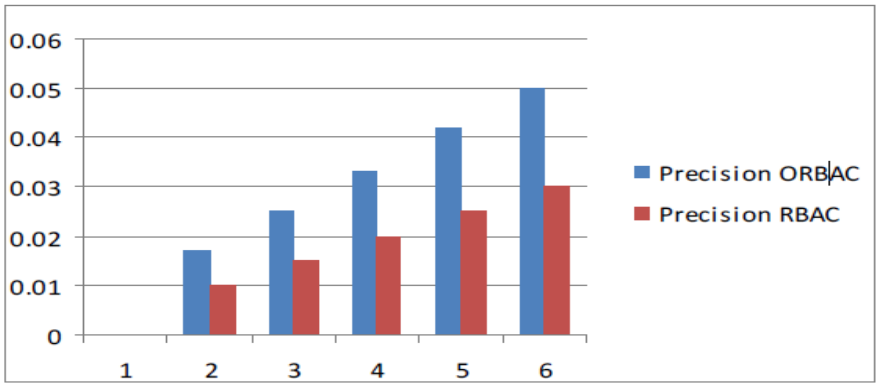

Figure 4.5: Histogram represents a difference between Precision ORBAC and Precision RBAC where context = 'emergency'

We notice that the precision of system based on the model ORBAC is higher compared with the system based on RBAC model (Figure 4.5) which signifies that the model based on the model ORBAC selects fewer documents to users and allows to reduce unauthorized accesses to documents, thus it allows to increase the confidentiality of documents not concerned by the given contexts.

4.8. Discussion

This step allows us to extract the benefits of our approach shown in:

- **Security:** When the number of resources on which the system must control the security access is restricted, there have fewer risks and vulnerabilities

with reference to the situation or the system must pay attention to the whole resource

- **Trust between provider of cloud and data owner:** data owner encrypted data before it stores these resources in a cloud.
- **Performance:** Since access to the model ORBAC is selective, the access resources became faster
- **Resource uses:** the selective access allows optimizing resources for reading, downloading, since the number of documents in a given context is much reduced.
- **Technology obsolescence**: new implementation infrastructure can be more easily integrated and supported by existing designs.
- **Portability:** existing functionality can be more rapidly migrated into new environments and platforms as dictated by the business
- **Quality**: the formal separation of concerns implied by this approach plus the consistency and reliability of the artifacts produced all contribute to the enhanced quality of the overall system.
- **Maintenance**: the availability of the design in a machine-readable form gives analysts, developers and testers direct access to the specification of the system, simplifying their maintenance chores.

4.9. Conclusion

In this chapter, we proposed an approach that aims to protect data stored in the cloud by offering a new layer between different users and the cloud. This layer will allow the property owner to manage data access and security of these data. Our proposal has been aligned with an MDA where security aspects are developed from an early development stage to the final implementation.

However, due to MDA existing functionality can be more rapidly migrated into new environments and the formal separation of concerns implied by this approach plus the consistency and reliability of the artifacts produced all contribute to the enhanced quality of the overall system

Chapter 5

CP ORBAC MODEL USING IMPLICIT SECURITY

5.1 Introduction

Securing data stored on distributed servers is of fundamental importance in cloud computing. The traditional (explicit) approach to securing data is to store and back it up on a single server and allow access upon the use of passwords that are needed to be frequently changed. But there is a tendency among users to keep passwords simple and memorable (28) leading to the possibility of brute force attacks. Furthermore since the data on the Web is archived, keys that provide adequate encryption today are likely to be broken in the future. This model implements a robust management system of access to data sources stored in a cloud based on ORBAC model and encryption. ORBAC is an access control model in which authorization is given to users depending on their role in an organization in a given context.

To enhance the security of data in this model, we add CP-ABE. This solution allows controlling access of users due to access structure proposed by data owner. This formula defines that the only authorized users are allowed to access the data sources after decryption of data because data will be encrypted by assigning for each resource a secret key.

However, in our case there is a large amount of data, the user may find it inefficient to create data partitions and distribute them over the network. For this, we propose in our approach to distribute the secret key into two or more pieces and stored at randomly chosen places on the network that are known only to the system. Our system retrieves the partitions to decrypt data if and only if the rule is checked. Our key partitioning scheme uses polynomials in Galois field $\mathrm{GF}(2^{\mathrm{m}})$.

5.2 Implicit Security

Past researches done in the area of cloud security has mainly focused on securing the communication of the information. But securing the storage of the information has been overlooked. Traditionally explicit techniques are used to secure data on servers. A data is stored and backed up on a single server. This technique it's never a good option to store a complete data at a server because if in any case the attacker managed to surpass the imposed security mechanisms then complete data will be easily accessed.

In such scenario, implicit security techniques may be proved to be beneficial .The data is partitioned into pieces and the partitions are stored over multiple servers. The main advantage of this technique is that there is no need to encrypt the data because these partitioned pieces will not provide any information to the attacker.

Implicit techniques are mainly used with two variations that depend on the size of the data to be stored online. If the size of data to be stored on server is small, then the data may be simply partitioned into n pieces and then stored over multiple servers, as Figure 5.1 shows.

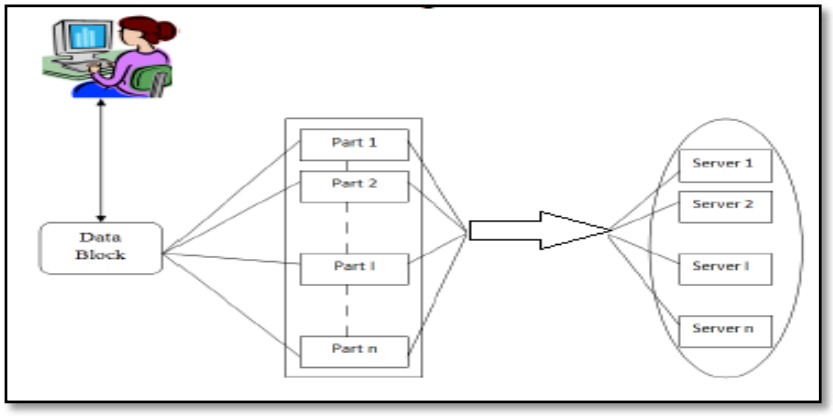

Figure 5.1: Scheme represents Implicit Security (small size)

If the size of data to be stored on server is large, and it becomes quite tough to deal with large partitions then the data may be first encrypted and the key is

partitioned into n pieces and then stored over multiple server, as Figure 5.2 represents.

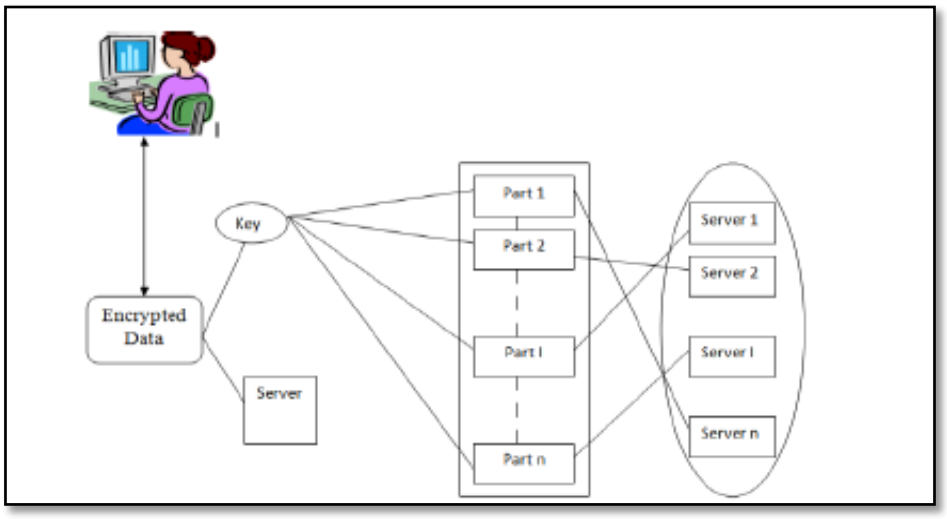

Figure 5.2: Scheme represents Implicit Security (large size)

5.3 Related works

In this section, we provide the state of the art related to the topic Implicit Security then we contrast them to our proposed approach.

Parakh and al. [**27**] tell about the use of a data partitioning scheme for implementing such security involving the roots of a polynomial in finite field. The partitions are stored on randomly chosen servers on the network and they need to be retrieved to recreate the original data. Data reconstruction requires access to each server, login password and the knowledge of the servers on which the partitions are stored. This scheme may also be used for data security in sensor networks and internet voting protocols.

The authors describe an implicit security architecture suited for the application of online storage. In this scheme data is partitioned in such a way that each partition is implicitly secure and does not need to be encrypted. These partitions are stored on different servers on the network which are known only to the user. Reconstruction of the data requires access to each server and the knowledge as to which servers data partitions are stored. Several variations of this scheme are

described, which include the implicit storage of encryption keys rather than the data, and where a subset of the partitions may be brought together to recreate the data.

We find the work of Karkouba and al. [**28**], they propose a way to protect data warehouses, to limit risks in cloud computing, and to provide confidentiality and availability of data. The proposal is to split each data stored in the warehouse on several cloud providers through the sharing secret algorithm. Algorithm secret sharing shares the data tuples from several suppliers. The way the data are distributed allows to store each data part, they are then not understandable and not exploitable by a malicious user in case of intrusion and secondly not to depend on one provider.

Parakh and al. [**55**], propose another scheme for dividing secret into shares and reconstructing the secret back from its shares is explained. In this scheme, additional information is added in the shares of the secret. This additional information is a message and the message is retrieved along with file (secret) on reconstructing the file (secret). Reconstruction of correct message shows the data is same as it was submitted which insures integrity of data (secret file). It is a $k-out-of-$n recursive secret sharing scheme which is based on n-ary tree data structure. In this scheme the user encodes extra secrets in the partitions of secret without increasing the size of original partition hence decreasing the effective size of partitions per secret hence increases the space efficiency of sharing the secret. This space efficiency is achieved with a trade off in security. This scheme is having application in area of secure distributed storage and Information dispersal algorithms.

Krawczyk and al. [**56**] offer a scheme based on distributed fingerprints. It uses Rabin's Information dispersal algorithm for fragmentation of data into pieces. In this scheme distributed fingerprints are used to find the fingerprints of information in distributed environment. These are known as public fingerprints which ensure data integrity. These fingerprints of data are stored without encryption keys. Using public fingerprints everyone in system can calculate fingerprinted information and if same is altered it is being noticed. To illustrate this technique they assume there is a secure storage in which information is stored for later use and same is not modified here. This safe space is used for integrity validation by fingerprinting the whole information by the help of short string and store the fingerprint result in this

space. The information here can be modified and without being noticed by replacing information with different piece of data whose fingerprint is the same. This type of fingerprints functions are hard to find and are called as collision free or one-way hash functions. In particular, no encryption keys are involved for handling them, and the same function can be used to fingerprint information of different sizes.

5.4 CP ORBAC model using implicit security

This part shows our third contribution which aims to strengthen our security model. We propose to break a secret key in several parties and distribute them on different servers to fortify the encryption phase.
Before continuing to detail our approach, we propose to give an overview from the mathematical model used.

5.4.1 Galois field

Galois Field, named after Evariste Galois, also known as finite field, refers to a field in which there exist finitely many elements. It is particularly useful in translating computer data as they are represented in binary forms. The most popular and widely used application of Galois Field is in Cryptography. Since each byte of data are represented as a vector in a finite field, encryption and decryption using mathematical arithmetic is very straight forward and is easily manipulable . In the 1970's, IBM developed Data Encryption Standard (DES). However, given that DES uses humble 56-bit key and technology advances rapidly, a supercomputer was able to break the key in less than 24 hours thus a more sophisticated algorithm was necessary. In 2001, Vincent Rijmen and John

On the other hand, computer data consist of combination of two numbers, 0 and 1, which are the components in Galois field whose number of elements is two. In the binary numeral system or base 2 number system, each value is represented with 0 and 1. To convert a decimal numeral system or base 10 number system into

binary system, to represent a decimal in terms of sums of an $a_n 2^n$. That is, if x is the said decimal number then we wish to have [**57**]: [**42**] [**42**]

$$x = \sum_{n \in N} a_n \, 2^n \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.1)$$

The coefficients an a_r is then written in descending order of n and all leading zeros are then omitted. The final result becomes the binary representation of the decimal x . Ultimately, binary system offers an alternative way of representing the elements of a Galois Field

Unlike working in the Euclidean space, addition (and subtraction) and multiplication in Galois Field requires additional steps

An addition in Galois Field is pretty straightforward. Suppose $f(p)$ and $g(p)$ are polynomials in $g\,f\,(p\,n\,)$.

Let $A = a_n - a_{n-1} - a_{n-2} \ldots a_1 a_0, B = b_n - b_{n-1} \ldots b_1 b_0 and c = c_n - 1 c_{n-1} - \ldots c_1 c_0$

be the coefficients of $f(p), g(p), \text{and } h(p) = f(p) + g(p)$ respectively. If $ak, bk,$ and ck are the coefficients of p^k in $f(p), g(p),$ and $h(p)$ respectively then

$$ck = ak + bk \,(\text{mod } p) \ldots\ldots\ldots\ldots\ldots\ldots (5.2)$$

Similarly, if $h(p) = f(p) - g(p) \text{then}$

$$ck = ak - bk \,(\text{mod } p) \ldots\ldots\ldots\ldots\ldots\ldots (5.3)$$

$\text{where } 0 \leq k \leq n - 1$. Since computer works in $gf(2^8)$, if ak and bk refer to the K^{th} bit in the bytes we wish to add then ck, the K^{th} bit in the resulting byte, is given by

$$ck = ak + bk \,(\text{mod } 2) \ldots\ldots\ldots\ldots\ldots\ldots (5.4)$$

Multiplication and Multiplicative Inverse Multiplication in Galois Field, however, requires more tedious work.

Suppose $f(p)$ and $g(p)$ are polynomials in $g\,f(p\,n\,)$ and let $m(p)$ be an irreducible polynomial (or a polynomial that cannot be factored) of degree at least n in $gf(p\,n\,)$. We want $m(p)$ to be a polynomial of degree at least n so that the product of two $f(p)$ and $g(p)$ does not exceed $11111111 = 255$ as the product needs to be stored as a byte. If $h(p)$ denotes the resulting product then

$$h(p) = \big(f(p) \cdot g(p)\big)\big(\mathrm{mod}\ m(p)\big) \text{..... (5.5)}$$

5.4.2 Proposed Model

The goal of our proposal is to increase a security and a confidentiality of our CP ORBAC approach. PC ORBAC uses ORBAC model and cryptography but in an explicit architecture. A data is stored and backed up on cloud servers but encryption key is stored on one server. In our case, the size of data to be stored on server is large, and it becomes quite hard to deal with large partitions then the data may be first encrypted and the key is partitioned into k pieces and then stored over multiple servers. In such situation, implicit security techniques may be proved to be beneficial .The Secret key is partitioned into pieces and the partitions are stored over multiple servers. The main advantage of this technique is that there is no need to encrypt the data because these partitioned pieces will not provide any information to the attacker. Original secret key can be reconstructed from these partitioned pieces or subset of these pieces whenever user wants to access complete data. The proposed model is as follows:

- **Data owner:** proposes an access structure to define authorized users where an access structure is a $\mathrm{tuple}(r, a, s, c)$, r is the role, a is the authorized actions, s is a resource and c is the request contextual information after he encrypts their data and stores it in data centre storage to share on the cloud and they decrypt it. Moreover, each data consumer is administrated by a control system.
- **User :** It is an entity that wants to access the data. If a user has a set of roles satisfying the access policy of the encrypted data, and it is not revoked in any valid attribute groups, then he will be able to decrypt the ciphertext and obtain the data.
- **Data storing centre:** It is an entity that provides a data sharing service.
- **Control and encryption system**: It is a responsible entity to control access to data and encryption. First it gives an authorization respons- e as a tuple is permitted (r, a, s) where r is a role, a is an action and s is resource. After encryption, system encrypts data and generates a secret key to data owner

and user. In this step, the secret key is partitioned into two or more pieces and stored at randomly chosen places on the network that are known only to the data owner.

Our system retrieves the partitions to decrypt data if and only if the rule is checked. Our key partitioning scheme uses polynomials in Galois field $GF(2m)$.

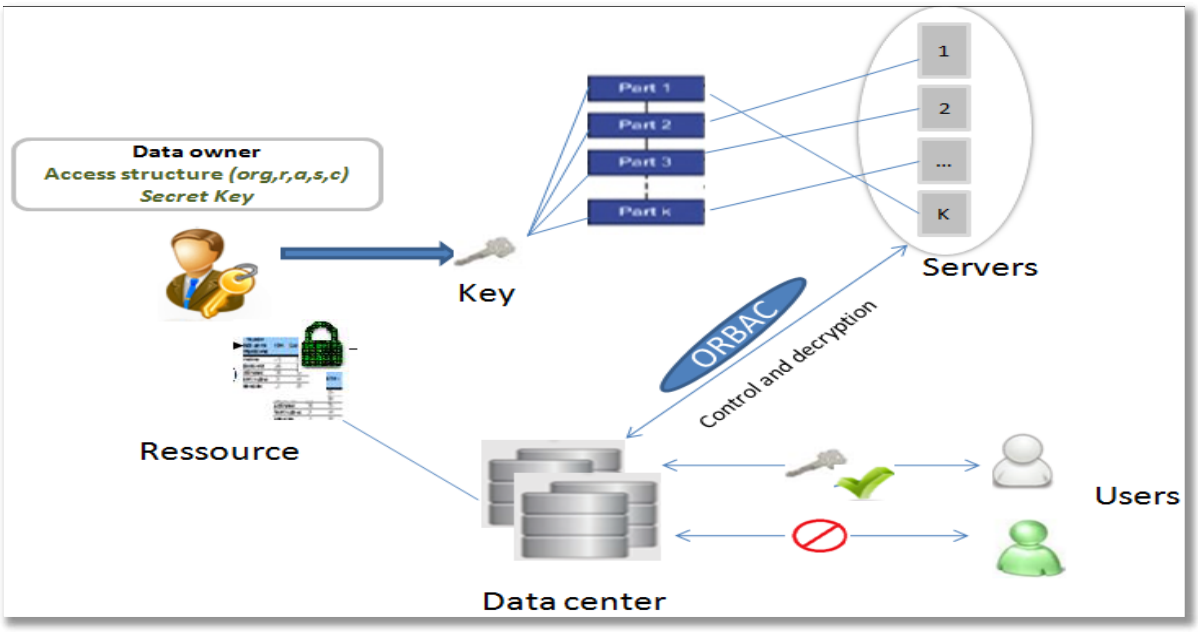

Figure 5.3: CP ORBAC using implicit security

Policy attribute-based encryption using key partitioning in Galois field is said to be irreducible if it cannot be factored into two or more polynomials each one with coefficients in $GF(2^m)$ and each degree less than m.

Therefore, a data owner can generate a random key and our system partitions it into Kpartitions using the following procedure. It generates K random polynomials of any random degree and computes their product modulo the irreducible polynomial $g(x)$.

The resultant polynomial of degree $m - 1$ is taken as the required key in its binary representation and the randomly generated polynomials are the partitions, so that we can think of the polynomials as bit strings corresponding to the coefficients that can only be 0 or 1, and each power of x represents a specific position in a bit string.

Example 6.1: In order to generate a random 8 bit key and create three partitions of it, the system may proceed as follows. Let we will consider all polynomials defined over modulo the irreducible polynomial $g(x) = x^3 + x + 1$. It chooses two polynomials of degree m – 1 (at random).

Such as $p1 = x^2 + x + 1$, $p2 = x^2 + 1$ and computes there product. $P1(x)p2(x) \equiv K(x) \bmod g(x)$,

where $K(x)$ is the key polynomial and the coefficients are the binary representation of the key. Figure 5.4 illustrates different steps to partition the encryption key.

$$(x^2 + x + 1) * (x^2 + 1) \bmod (x^3 + x + 1)$$
$$= (x^4 + x^3 + x + 1) \bmod (x^3 + x + 1)$$
$$= -x^2 - x$$
$$= x^2 + x$$

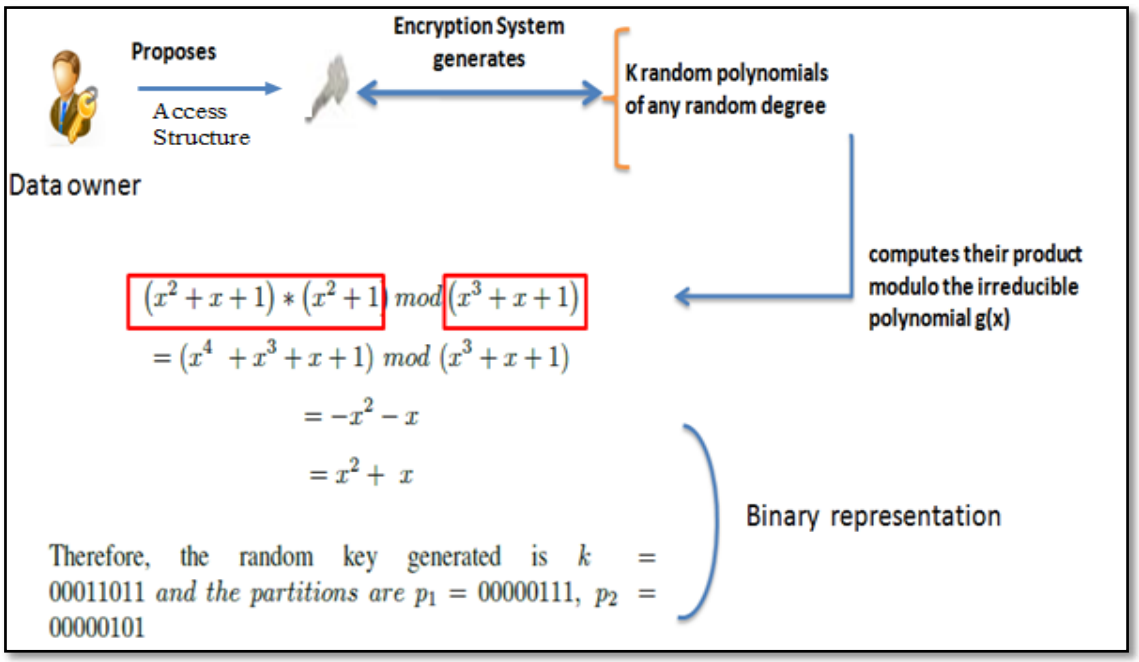

Figure 5.4 : An illustration represents an example using partitioning scheme.

5.4.3 Proposed algorithm CP ORBAC Based Encryption:

In this section, we show our proposed algorithm based on ABE algorithm and in the same time, we integrate the principle of implicit security. The algorithm consists of four steps. Figure 5.5 illustrates the CP ORBAC BE algorithm that was developed for distribute and encrypt data.

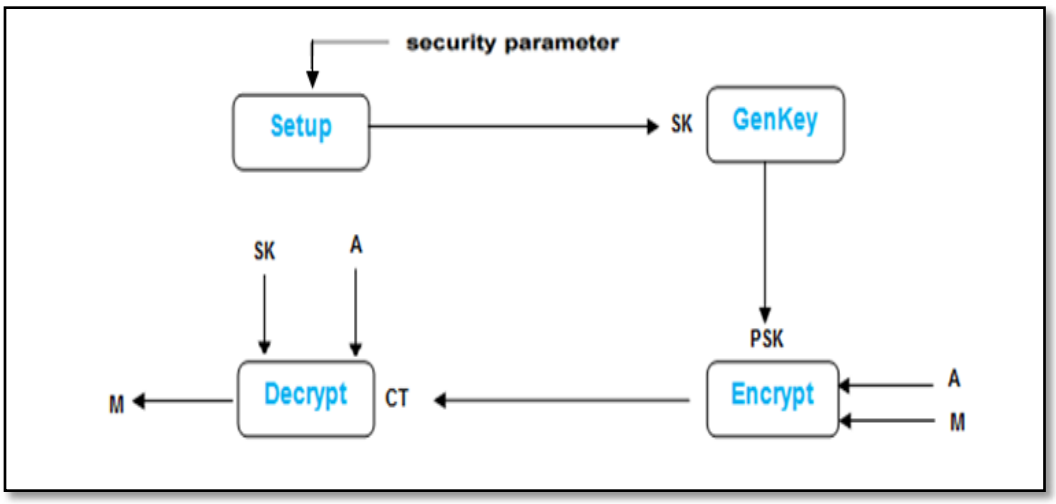

Figure 5.5: Scheme represents CP ORBAC BE algorithm

Setup: This algorithm takes as input the security parameters

- A set of attributes proposed by data owner on access structure
- A secret key proposed by data owner.

KeyGen: This algorithm takes as input :

- The secret key proposed by data owner and it generates random partitions of Secret key (PSK) using the following procedure:
  - Generating K random polynomials of any random degree in Galois field $\mathbf{GF}(\mathbf{2^m})$.
  - Computing their product modulo the irreducible polynomial g(x).

- Taking the resultant polynomial of degree $m-1$ is taken as the required key in its binary representation and the randomly generated polynomials are the partitions.

Encrypt $(\mathbf{PSK}, \mathbf{M}, \mathbf{A})$**:** This is a randomized algorithm that takes as input:

- A message M
- A set of attributes A
- A secret key PSK.It outputs the ciphertext CT.

Decrypt $(\mathbf{SK}, \mathbf{A}, \mathbf{CT})$: This algorithm takes as input:

The ciphertext CT that was decrypted under SK and a set of attributes$A$. It outputs M.

5.5 Experimental setup

In this part, we propose to evaluate our approach by comparing the implicit and the explicit approaches.

Using the explicit and the implicit architecture, we store a secret key on a number of servers in which some servers will be attacked. We decide to use both models evaluation measures recall and precision.

$$Recall = \frac{Number\ \ of\ Attacked\ \ Servers\ \ includes\ \ Partitions\ \ of\ SK}{Number\ \ of\ Servers\ \ includes\ \ partitions\ \ of\ SK} \quad \text{..... (5.1)}$$

$$Precision = \frac{Number\ \ of\ Attacked\ \ Servers\ \ includes\ \ Partitions\ \ of\ SK}{Number\ \ of\ Servers} \quad \text{..... (5.2)}$$

The results are represented on the next table:

Cases	Servers	Servers/ Partitions	Servers/ Attacked	Recall (implicit)	Recall (explicit)	Precision (Implicit)	Precision (explicit)
1	10	3	2	66%	100%	20%	100%
2	50	10	5	50%	100%	10%	100%
3	100	15	10	66%	100%	10%	100%
4	150	50	25	60%	100%	16%	100%

Table 5.1: Results of recall, precision in implicit and explicit approaches

The next histograms shows our table's results:

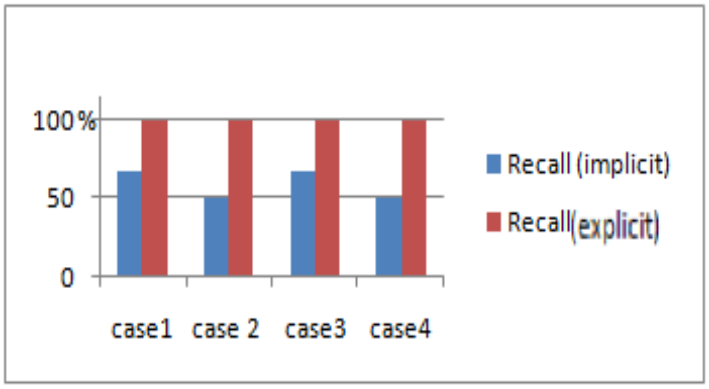

Figure 5.6: Histogram represents difference between Recall (implicit) and Recall (explicit).

We notice that the recall of our model used implicit approach (Figure 6.5) is lower compared with explicit approach which signifies that a number of servers includes partitions of our secret key is more protected.

For example, in the first case, in implicit architecture, we note that the attacker was not able to detect only 66% of a secret key and he could not decrypt and access to resource.

By against in explicit architecture, attacker has got 100% of a secret key and he has a possibility to decrypt data

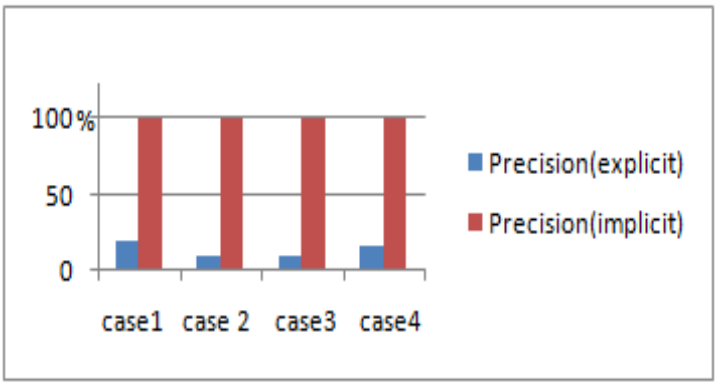

Figure 5.7: Histogram represents difference between Precision (implicit) and Precision (explicit)

We notice that the precision of our model used implicit approach (Figure 6.6) is lower compared with explicit approach which signifies that the distribution of partitions of a secret key into different servers increases the confidentiality and a security of a secret key and data.

5.6 Conclusion

We have described an implicit security architecture suited for the application of cloud. In this scheme encryption key is partitioned in such a way that each partition is implicitly secure and does not need to be encrypted. These partitions are stored on different servers on the network which are known only to the data owner and user due to the encryption key. Reconstruction of the key requires access to each server where partitions are stored. The advantage of this scheme is that the partitioned data pieces cannot reveal any user information but in case user forgets in which server the partitioned data is stored, it will become difficult to reconstruct original data.

CONCLUSION

Cloud computing is a way to increase the capacity or add capabilities dynamically without investing in new infrastructure, training new personnel, or licensing new software. Despite of all the hype surrounding the cloud, enterprise customers are still reluctant to deploy their business in the cloud. Security is one of the major issues which reduces the growth of cloud computing.

We have described an approach to protect access to data on cloud computing based on ORBAC which authorization is given to users depending on their role in an organization. In a given context then we use the Model Driven Architecture (MDA) to define security in our proposition. We concretely define security specifications in the Conceptual Data Model (PIM), independently of the target logical Data model.

CIM is also often referred to as a business or domain model. It presents exactly what the system is expected to do, but hides all information technology related to specifications to remain independent of how that the system will be.

At the **PIM level**, the secure data model is carried out without considering the selected technology, since this model is independent of the platform. This PIM is represented through an extended UML class diagram which furthermore permits the specification of security constraints on the model.

At the **PSM level**, the data logical design is performed, taking into account the selected target platform in which the Security model will be implemented.

We have presented a process $\mathrm{GF}(2^{\mathrm{m}})$ which allows encrypting data using an access structure proposed by data owner. In this scheme a secret key is partitioned using a $\mathrm{GF}(2^{\mathrm{m}})$ such a way that each partition is implicitly secure. These partitions are stored on different servers on the network. We propose a process which allows encrypting data using an access structure proposed by data owner. In this scheme, a secret key is partitioned using a Galois field. These

partitions are stored on different servers on the network. We tested our approach, it gives a constructive results.

Our future work is to develop an approach based on digitally signed documents, or certificates, that convey identity, authorization, and attributes.

Data owner wants to be able to control access to the resources they offer. We can summarize the requirements for an access control system in cloud as follow:

It must support ORBAC model. It must permit fine-grained access control on data, where the sources of authority (SOA) that may issue permissions are individual users owning the data. It must provide the traceability of all accesses to data. Our proposition supports two concepts role and context used on ORBAC model. ORBAC is an access control model in which authorization is given to users depending on their role in an organization in a given context. In first step Data owner SOA installs or stores encrypted resource on a resource server.

He contacts the provider of cloud and registers himself as source of authority for this resource in the metadata base of Policy Decision Point. A Policy Decision Point (PDP) evaluates participant requests against relevant policies/contracts and attributes to render an authorization, eligibility or validation decision or provide calculated results. It is typically associated with one or more Policy Enforcement Point's (PEP's) However, for each resource, the PDP stores only its SOA identifier. To grant access to a resource consumer (step2) the SOA issues rules that allows access to the resource. In contrast, the resource server needs two different modules. The policy Decision Point that produces an access control decision based on the request provided by the user and Policy Enforcement Point (PEP). However, PEP is a network device on which policy decisions are carried out or enforced.

A. List of symbols

IT	Information Technology
MAC	Common Mandatory Access Control
DAC	Discretionary Access Control
RBAC	Role Based Access Control
ORBAC	Organization Role Based Access Control
MDA	Model Driven Architecture
OMG	Object Management Group
NIST	The National Institute of Standards and Technologies
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
TCSEC	Trusted Computer System Evaluation Criteria
VM	Virtuel Machine
KDC	Key Data Center
DACC	Distribute Access Control
ABE	Attribute Based Encryption
KP ABE	Key Policy Attribute Encryption
CP ABE	Ciphertext Policy Attribute Encryption
HIBE	Hierarchical Identity Based Encryption
PRE	Time Proxy re encryption
CSP	Cloud Service Provider
M	Message
E	Ciphertext
CIM	Computation Independent Model
PIM	Platform Independent Model

PSM	Platform Specific Model
UML	The Unified Modeling Language
GF	Galois Field
SOA	Source Of Authority
PDP	Policy Decision Point
PEP	Policy Enforcement Point

B. List of Publications

- International Journals:

- Yasmina Ghebghoub, Saliha Oukid, Omar Boussaid , An MDA approach to secure access to data on cloud using implicit security, International Journal of Information and Computer Security 8(2):107 · January 2016 IJICS 8(2): 107-120 (2016).
- Yasmina Ghebghoub, Saliha Oukid, Omar Boussaid, A Survey on Security Issues and the Existing Solutions in Cloud Computing, International Journal of Computer and Electrical Engineering, Vol. 5, No. 6, December 2013

- **International Conferences:**

- Yasmina Ghebghoub, Saliha Oukid, Omar Boussaid , Security model based encryption to protect data on cloud, ISDOC 2014: 50-55, March 2014.

- Yasmina Ghebghoub, Saliha Oukid, Omar Boussaid, CP ORBAC to secure access to data on cloud using implicit security, 2nd International Conference on Networking and Advanced Systems, ,May 2015.

Bibliography

[1] Y. Zhu, H. Huy, G. Ahny, D. Huang, and S. Wang, "Towards Temporal Access Control in Cloud Computing," *Huang US National Science Foundation (NSF-IIS-0900970 and NSFCNS-0831360) and Department of Energy (DE-SC0004308). D. Huang'sresearch is sponsored by Office*, 2008.

[2] B.Véla, C.Blanco, E.Medina, and E.Marcos, "Model Driven Development of Secure XML DataWarehouses: A Case Study," *EDBT 2010*, 2010.

[3] Y. Gil, D. Artz, "A survey of trust in computer science and the semantic web," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2007.

[4] DoD Computer Security Center, "Trusted computer system evaluation criteria, " *DoD 5200.28-STD,* 1985.

[5] V. Varadharajan A. Nagarajan, "Dynamic trust enhanced security model for trusted platform based," *Future Generation Computer Systems*, 2010.

[6] International Telecommunication Union, "Public-key and attribute certificate frameworks," *ITU, X-Series*, 2001.

[7] A. Giddens, "The Consequences of Modernity," *Polity Press, UK*, 1991.

[8] Cloud Security Alliance,". Top threats to cloud computing," 2010.

[9] D. Lekkas D. Zissis, "Addressing cloud computing security issues," *University of the Aegean, Greece.*, 2012.

[10] Philippe Bunel, "An introduction to Intrusion Detection Systems," *Security Essentials Certification (GSEC) Practical Assignment*, 2004.

[11] R.Sandhu , Q.Munawer S.Osborn, "Configuring role-based access control to enforce mandatory and discretionary access control policies," *ACM Transactions on Information and System Security*, vol. 3, no. 2, pp. 85-106, 2000.

[12] D. Bell and L. LaPadula, "Secure computer system:Unified exposition and multics interpretation.," vol. TRM74-244, 1976.

[13] U.Swapnaja, S. Apte Sulabha, G. Modani Dattatray D. Bokefode Jayant, , vol. 104, no. 5, 2014.

[14] D.Richard, R.Chandramouli D Ferraiolo, "Role Based Access Control," *ARTECH HOUSE, INC,2007DoD, Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28-STD.*, no. , ISBN 13: 978-1-59693-113-8

pp. 1-5.

[15] B Kalisk, "A Survey of Encryption Standards," RSA Laboratories , 02721732/93/1200-0074, IEEE ,1993.

[16] E. Chaniotakis N. Moshopoulos, "A Survey of Cryptography Algorithms – Trends and Products" National Technical University of Athens Electrical & Computer Engineering Department, IJCS, 2010.

[17] Microsoft, "Services Development Web Service Security ," vol. Chapter 2: Message Protection Patterns, pp. 77-78, December 2005.

[18] A.Sahai and B. Waters, "Fuzzy Identity Based Encryption," *Advances in Cryptology – Eurocrypt*, vol. 3494 of the series Lecture Notes in Computer Science, pp. 457-473, 2005.

[19] E. Naone, Technology overview, conjuring clouds, MIT Technology Review, July–August, 2009.

[20] H. ZAKI D. Jamil, "SECURITY ISSUES IN CLOUD COMPUTING AND COUNTERMEASURES," *International Journal of Engineering Science and Technology (IJEST)*, vol. 3 , no. ISSN : 0975-5462, 2011.

[21] G. Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, in: Theory in Practice," 2009.

[22] C. Yeo, S. Venugopal, S. Malpan B. Rajkumar, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing," *5th utility, Future Generation Computer Systems* , 2009.

[23] C. Wang, K. Ren, W., Jin Li Q. Wang, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE network , 2010.

[24] R. Anderson, "In Security engineering: a guide to building dependable distributed systems.," *John Wiley & Sons Inc.*, 2001.

[25] N. Padia and M. Parekh, "Cloud Computing Security Issues, in Enterprise Architecture and Its Solutions," *International Journal of Computer Application*, vol. vol.2, issue 1, pp. 149-155,December 2011, 2011.

[26] P. Townend, and J. Arshad, "A Novel Intrusion Severity AnalysisApproach for Clouds, School of Computing," pp. 1-13, 2011.

[27] S. Kak. A.Parakh, ""Online data storage using implicit security," *Information Sciences Journal*, vol. 179, pp. 3323-3331, 2009.

[28] N. Harbi, J. Darmont, and G. Gavin K. Karkouda, "Confidentialité et disponibilté des donnéesentreposéesdans les nuages," pp. 1-14, 2012.

[29] P. Raj S. Sajithabanu, "Data Storage Security in Cloud," vol.. 2, pp. 437-440, 2011.

[30] S. Nawaz Brohi M. Adib Bamiah, "Seven Deadly Threats and Vulnerabilities in Cloud Computing," vol. 09, pp. 087-090, 2011.

[31] A. Nayak, and V. Stojmenovic S. Ruj, ""DACC: Distributed Access Controlin Clouds," pp. 91-98, 2011.

[32] N. Antony and A. A. R. Melvin, "A Survey on Encryption Schemes in the Clouds for Access Control," no. 135-1139, 2012.

[33] S.Singh P.Singh, "RBAC to Enhance the Security in Cloud Computing," vol. 3, no. 2277 128X, 2013.

[34] A.Khan, "Access control in cloud computing environment," *ARPN Journal of Engineering and Applied Sciences*, vol. 7, no. 5, 2012.

[35] Liu and Z. Cao, "On efficiently Transferring the Linear Secret-Sharing Scheme Matrix in Ciphertext-Policy Attribute Based Encryption," Report 2010/374,Version 20100702:085423, 2010.

[36] A.Shamir, "Identity-based cryptosystems and signature schemes," *CRYPTO*, pp. 47–53, 1984.

[37] A. Boyen D. Boneh, "Efficient selective-id secure identity based encryption without random oracles," *EUROCRYPT*, pp. 223–238, 2004.

[38] S. Halevi,J. Katz R.Canetti, "A forward-secure public-key encryption scheme," pp. 255–271, 2003.

[39] C. Gentry, "Practical identity-based encryption without random oracles," *EUROCRYPT*, pp. 445–464, 2006.

[40] J.Horwitz B. Lyn, "Toward hierarchical identity based encryption," pp. 466–481, 2002.

[41] T. Okamoto, A. Sahai, K.i Takashima, B. Waters A. Lewko, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," *Advances in Cryptology – EUROCRYPT 2010*, vol. 6110, pp. 62-91, 2010.

[42] F.Truyen, "The Fast Guide to Model Driven Architecture,The Basics of Model Driven Architecture (MDA)," pp. 1-3, January 2006.

[43] C.Blanco, E.Medina B.Véla, "Model Driven Development of Secure XML Data Warehouses," *EDBT 2010.*, 2010.

[44] C.Blanco, J.N. Mazón, E.Medina, E.Marcos, J. Trujillo B.Véla, "Development of Secure XML Data Warehouses with QVT," *Journal of Information and Software Technology*, 2013.

[45] E. Marie S.Farhan, "Transforming Conceptual Model into Logical Model for Temporal Data Warehouse Security: A Case Study," vol. 3, no. 3, 2012.

[46] A. Beimel, "Secure Schemes for Secret Sharing and Key Distribution," *PhD Thesis, Israel Institute of Technology*, 1996.

[47] G.Booch, "Object Oriented Analysis and Design with Applications," *Addison Wesley*, vol. 2, 1994.

[48] M.Blaha, W.Permerlani, F.Eddy , W.Lorensen J. Rumbaug, "Object Oriented Modeling and Design," 1992.

[49] M. Christerson, P.Jonsson and G.Overgaard I. Jacobson, "Object Oriented Software Engineering: A Use Case Driven Approach," *Addison- Wesley*, 1992.

[50] Object Management Group (OMG), "Unified Modeling Language (UML) Specification 1.5. Internet:http://www.omg.org ," 2003.

[51] S.Mora, "Data Warehouses Design with UML, Department of Spftware and Computing Systems," pp. 4-6, 2005.

[52] J.Samos, F.Saltor A.Abello, "Benefits of an Object Oriented Multidimentional Data Model," vol. 1944, pp. 141-152.

[53] R.El Baida, P.Balbiani,S. BenferhatF. Cuppens, Y. Deswarte,A. Miege A.El Kalam, "Or-BAC: un modele de controle d'acces base sur les organisations," vol. 2, pp. 30-43.

[54] S. Mudumbai, G Hoo, Keith Jackson, A. Abdelilah W.Johnston, "Certificate- -based Access Control for Widely Distributed Resources," *8th UsenixSecurity Symposium*, 1999.

[55] S. Kak A.Parakh, "A tree based recursive information hiding scheme," *Communications (ICC), 2010 IEEE International Conference*, pp. 1-5, 2010.

[56] H.Krawczyk, "Distributed fingerprints and secure information dispersal," *symposium on Principles of distributed computing*, pp. 207-218, 1993.

[57] J. Benvenuto, Galois Field in Cryptography, Chapter 2, pp. 4-5, May 2012.