2-1-2014

Data security in cloud storage services

Mai Mansour Dahshan

THE AMERICAN UNIVERSITY IN CAIRO

SCHOOL OF SCIENCES AND ENGINEERING

DATA SECURITY IN CLOUD STORAGE SERVICES

A THESIS SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

THE DEGREE OF THE MASTER OF SCIENCE

BY:

MAI MANSOUR DAHSHAN

SUPERVISOR

DR. SHERIF El-KASSAS

FALL 2013

To Jana

ACKNOWLEDGMENT

In the name of Allah the most gracious and most merciful

First and foremost I would like to thank my great supervisor, Dr. Sherif Elkassass, who always helped me, welcomed my questions and gave me a lot of recommendations and suggestions. I would not have reached this phase, if it were not for his permanent support, advice, and guidance.

I also would like to express my gratitude to my committee members, Dr. Aly Fahmy, Dr. Amr ElKadi and Dr. Sherif Aly for their support, guidance and helpful feedback.

I would also thank all my colleagues at the department of computer science. Thanks go to Amira Shoukry and Shaimaa Hegazy for being the best officemates and friend one could wish for.

I am greatly indebted to my parents. They are always very understanding and supportive on my choices. They love me more than themselves and have sacrificed so much to support me. I am very indebted to my grandfather and grandmother, who passed away during my study.

Last but not least, I would like to express my eternal gratitude sisters and friends for their permanent support, appreciation and patience. I am also grateful for my husband and my daughter, who have given me a lot of help and support during doing my research. I would like to dedicate this thesis to them all.

ABSTRACT

Cloud Computing is considered to be the next-generation architecture for ICT where it moves the application software and databases to the centralized large data centers. It aims to offer elastic IT services where clients can benefit from significant cost savings of the pay-per-use model and can easily scale up or down, and do not have to make large investments in new hardware. However, the management of the data and services in this cloud model is under the control of the provider. Consequently, the cloud clients have less control over their outsourced data and they have to trust cloud service provider to protect their data and infrastructure from both external and internal attacks.

This is especially true with cloud storage services. Nowadays, users rely on cloud storage as it offers cheap and unlimited data storage that is available for use by multiple devices (e.g. smart phones, tablets, notebooks, etc.). Besides famous cloud storage providers, such as Amazon, Google, and Microsoft, more and more third-party cloud storage service providers are emerging. These services are dedicated to offering more accessible and user friendly storage services to cloud customers. Examples of these services include Dropbox, Box.net, Sparkleshare, UbuntuOne or JungleDisk. These cloud storage services deliver a very simple interface on top of the cloud storage provided by storage service providers. File and folder synchronization between different machines, sharing files and folders with other users, file versioning as well as automated backups are the key functionalities of these emerging cloud storage services.

Cloud storage services have changed the way users manage and interact with data outsourced to public providers. With these services, multiple subscribers can collaboratively work and share data without concerns about their data consistency, availability and reliability. Although these cloud storage services offer attractive features, many customers have not adopted

these services. Since data stored in these services is under the control of service providers resulting in confidentiality and security concerns and risks. Therefore, using cloud storage services for storing valuable data depends mainly on whether the service provider can offer sufficient security and assurance to meet client requirements. From the way most cloud storage services are constructed, we can notice that these storage services do not provide users with sufficient levels of security leading to an inherent risk on users' data from external and internal attacks. These attacks take the form of: data exposure (lack of data confidentiality); data tampering (lack of data integrity); and denial of data (lack of data availability) by third parties on the cloud or by the cloud provider himself. Therefore, the cloud storage services should ensure the data confidentiality in the following state: data in motion (while transmitting over networks), data at rest (when stored at provider's disks).

To address the above concerns, confidentiality and access controllability of outsourced data with strong cryptographic guarantee should be maintained. To ensure data confidentiality in public cloud storage services, data should be encrypted data before it is outsourced to these services. Although, users can rely on client side cloud storage services or software encryption tools for encrypting user's data; however, many of these services fail to achieve data confidentiality. Box, for example, does not encrypt user files via SSL and within Box servers. Client side cloud storage services can intentionally/unintentionally disclose user decryption keys to its provider. In addition, some cloud storage services support convergent encryption for encrypting users' data exposing it to “confirmation of a file attack.” On the other hand, software encryption tools use full-disk encryption (FDE) which is not feasible for cloud-based file sharing services, because it encrypts the data as virtual hard disks. Although encryption can ensure data confidentiality; however, it fails to achieve fine-grained access control over outsourced data.

Since, public cloud storage services are managed by un-trusted cloud service provider, secure and efficient fine-grained access control cannot be realized through these services as these policies are managed by storage services that have full control over the sharing process. Therefore, there is not any guarantee that they will provide good means for efficient and secure sharing and they can also deduce confidential information about the outsourced data and users' personal information.

In this work, we would like to improve the currently employed security measures for securing data in cloud store services. To achieve better data confidentiality for data stored in the cloud without relying on cloud service providers (CSPs) or putting any burden on users, in this thesis, we designed a secure cloud storage system framework that simultaneously achieves data confidentiality, fine-grained access control on encrypted data and scalable user revocation. This framework is built on a third part trusted (TTP) service that can be employed either locally on users' machine or premises, or remotely on top of cloud storage services. This service shall encrypts users data before uploading it to the cloud and decrypts it after downloading from the cloud; therefore, it remove the burden of storing, managing and maintaining encryption/decryption keys from data owner's. In addition, this service only retains user's secret key(s) not data. Moreover, to ensure high security for these keys, it stores them on hardware device. Furthermore, this service combines multi-authority ciphertext policy attribute-based encryption (CP-ABE) and attribute-based Signature (ABS) for achieving many-read-many-write fine-grained data access control on storage services. Moreover, it efficiently revokes users' privileges without relying on the data owner for re-encrypting massive amounts of data and re-distributing the new keys to the authorized users. It removes the heavy computation of re-encryption from users and delegates this task to the cloud service provider (CSP) proxy servers.

These proxy servers achieve flexible and efficient re-encryption without revealing underlying data to the cloud.

In our designed architecture, we addressed the problem of ensuring data confidentiality against cloud and against accesses beyond authorized rights. To resolve these issues, we designed a trusted third party (TTP) service that is in charge of storing data in an encrypted format in the cloud. To improve the efficiency of the designed architecture, the service allows the users to choose the level of severity of the data and according to this level different encryption algorithms are employed. To achieve many-read-many-write fine grained access control, we merge two algorithms (multi-authority ciphertext policy attribute-based encryption (MA- CP-ABE) and attribute-based Signature (ABS)). Moreover, we support two levels of revocation: user and attribute revocation so that we can comply with the collaborative environment. Last but not least, we validate the effectiveness of our design by carrying out a detailed security analysis. This analysis shall prove the correctness of our design in terms of data confidentiality each stage of user interaction with the cloud.

Table of Contents

List of Figures

1 Introduction

Cloud computing has been envisioned as the next generation computing model that comes from grid computing, distributed computing, parallel computing, virtualization technology, utility computing and other computer technologies. This combination of computing models provide cloud computing with more advantages such as large scale computation and data storage, virtualization, high expansibility, high reliability and low price service. The cloud computing model is mainly based on the network and has the format of service for the consumers. These services can take the form of application, software, and infrastructure and it can be accessed by users from anywhere and at any time. In addition, these services can be shared among a large number of users. For example, the cloud storage can be shared by multiple users and each user can increase or decrease his resources of storage based upon his needs. Since the cloud services are accessible from anywhere in the world, the cloud appear as if it is a single point of access for all the computing needs of consumers. According to the U.S. National Institute of Standards and Technology (NIST) cloud computing can be defined as: "Cloud computing is a model for enabling 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. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models"[1].

Cloud storage is a newly developed concept in the field of cloud computation. It can be defined as a system that is composed of the cluster, grid and distributed file systems

that using application software coordinates a variety of different type's storage devices together to provide data storage and access service. In other words, cloud storage allows users to outsource their data that has been managed internally within the organization or by individual users.

The outsourcing of this data eliminates the concerns associated with the installation of the complex underlying hardware, saves increasing high cost in data center management and alleviates the responsibilities of its maintenance. Therefore, a large number of organizations and individuals are adopting these storage services by placing their data in their cloud storage. However, there are security concerns associated with cloud storage [2].

Recently, data security has been regarded as one of the main obstacles that block the development of cloud storage service. A study [3] surveyed more than 500 CTO and IT managers in 17 countries, showed that despite the potential benefits of cloud storage, organizations and individuals do not trust the existing cloud storage service providers because the fear of the security threats associated with them. The cloud system in general can be divided into several types according to the users and range of cloud [3].The same cloud system may serve different types of users ranging from customers, enterprises, individuals. The security issues and risk of each type is different. In cloud environment, data is stored in a public storage provider. This data is stored in the provider's hard drive and no one except the provider has control over it and knows where exactly it is saved.

When individual users and organizations outsource their data to multi-tenant environment as the cloud, they expect to have the same level of data security as they would have in their own premises [4, 5]; However, this not the case in cloud. In 1999,

Scott Mc-Nealys, the CEO of Sun Microsystems, shocked the media by undermining privacy expectations in the digital world with the statement: “You have zero privacy anyway. Get over it” [6]. Ten years later, Eric Schmidt, the CEO of Google, said that “If you have something that you do not want anyone to know, maybe you should not be doing it in the first place” [7].Therefore, users cannot trust cloud for their data confidentially. Data confidentiality ensures that CSP and unauthorized subscribers cannot learn any information about the outsourced data. In addition, the access control is not of same level as on premises. Access control is more than just controlling which users can access which resource. Access control manages users, files and other resources. It controls user’s privileges to files or resources. As a consequence, unauthorized parties must be prevented from gaining access to sensitive data so that data loss and leakage that can prevent [8].

All the above issues rises questions as who has access to the data, who encrypts the data, where are the encryption keys stored, who manages the access to the data, what is left behind when you scale down a service, and how is data protected.

This idea of securing data in cloud storage services has attracted many researchers to work in this field with the aim of constructing a trusted control model of cloud storage. In this study, we are interested in providing a secure framework for data security in the cloud storage services such as dropbox, that offer more functionalities such as file sharing and synchronization besides cloud storage basic functionalities. After studying all the previous and the current research done in this area, we would like to provide data confidentiality against cloud and data confidentiality against accesses beyond authorized rights. Therefore, we design a dynamic collaboration environment utilizing the benefits of cloud storage

while ensuring strong data security and fine-grained data access.

1.1 Problem Definition

Many cloud service providers provide storage as a form of service; data is transferred from user machine to be stored on large data centers. Although cloud storage providers often state that they offer safe environment for stored data, there have been cases discovered where users' data has been modified or lost due to some security breach or some human error.

A recent security flaw in the Dropbox authentication mechanism [9] begins the debate about whether cloud storage services are sufficiently secure to store sensitive data. Moreover, a recent research [10] about dropbox has shown that it suffers from three types of attacks which are hash value manipulation attack, stolen host id attack and direct download attack. In these attacks, the attacker is able to upload and link arbitrary files to the victim's Dropbox account once he have the host id. Moreover, Dropbox itself announced that it enables government agents to access customers' data. This means that there is a back door mechanism to access data which might be exploited. Moreover, another cloud storage service as Box may not encrypt user files via SSL during transfer to/from Box and may not encrypt data within Box servers [11]. Even in the more secure storage service, SpiderOak, user's data is encrypted with his own private encryption key and his password which can make it inaccessible in case of password loss [12].Furthermore, [13] evaluated four cloud storage systems: Mozy, Carbonite, Dropbox, and CrashPlan. After the evaluation, it was found out that none of these systems can provide any guarantees for data integrity, availability, or even confidentiality. For all

these reasons, a lot of customers are not rushing to use cloud storage services in spite of their appealing features from backup, file sharing, and synchronization.

From the above, we can notice that securing data in cloud storage services is an important aspect of quality of service. Since various cloud service providers use different methodologies to guarantee the safety of the data stored in their cloud. This raises the question of whether or not the methods used by these storage providers really secure the stored data. Because of the virtualized nature of cloud storage traditional mechanisms of handling data security will not be suitable in the cloud model [14]. Moreover, till this point little focus have been given to research addressing the issue of securing data in the advanced features of cloud storage services from file sharing and synchronization.

Most of the research done in this field has focused on the following: (1) Using cryptographic primitives from different encryptions techniques for the purpose of data confidentiality. The most famous technique for providing data storage security is utilizing the homomorphic token with distributed verification of erasure-coded data [15]; and (2) Verifying correctness of data storage by using data integrity techniques as in [16]. However, these techniques are useful for ensure the storage correctness without having users possessing data; they cannot address all data storage security threats because they do not take in consideration dynamic data operations.

Cloud storage is considered to be an important service that allows data owners to host their data in the cloud and access it at any time from any place. Therefore, data access control is an effective way to ensure data security. However, cloud storage service separates the roles of the data owner from the data service provider, and the data owner does not interact with the user directly for providing data access service, which makes the

data access control a challenging issue in cloud storage systems. Since the cloud storage service server is not a trusted entity that data owner can rely on for ensuring efficient data access control and traditional server-based access control methods are no longer applicable to cloud storage systems. Therefore, we need to define a cloud-based file sharing service that ensures data security in terms of data confidentiality against cloud and provider threats and data confidentiality against users' unauthorized access by implementing a fine-grained access control mechanism without relying on the cloud storage service for providing access control.

Most of the research done in this field has focused on providing efficient data access control mechanisms between data owners and data users and cloud storage. The data owners encrypts the data and access control policies locally and upload the data to the cloud and provide secret keys to users it want to share with and leave to cloud the task of managing the access control without have access to any keys. However, this model of access control is not feasible in cloud-based file sharing service where there is no direct interaction between the data owners and the data users. This means that most of the research has focused on data security in terms of access control in cloud storage models where the data owners can directly interact with data users. On the other hand, small amount of research is done about ensuring the data security cloud-based file sharing service where there is no direct interaction between data owners and data users [17].

As mentioned above, the current cloud storage services suffer from a number of data security limitations. Motivated by their limitations, in this thesis we pose the following research question:

Research Question: How to construct cryptographic scheme that can enforce data confidentiality and distributed data access control efficiently in dynamic environments?

In other words, we want construct a cloud-based file sharing service that ensures data security in terms of data confidentiality against cloud and provider threats and data confidentiality against users' unauthorized access by implementing a fine-grained access control mechanism.

1.2 Motivation and Objective

Cloud storage services have its advantages and disadvantages. The main advantages of storage services are capital cost savings, because users do not need to invest their money to own storage servers nor do they have to maintain these servers, and scalable; since users can easily increase or decrease their storage capacity based on their needs. In addition, more features are added to cloud storage services such as file sharing and synchronization which make it more appealing for users to use. However, the main disadvantages that pushes users away from adopting cloud storage is usually whether it is sufficiently secure or not.

As a large amount of electronic data is being generated, there is a need for dynamic storage systems that could be able to can hold that data. Moreover, users usually store their data on multiple devices and need to access the recent version of this data from any place. Therefore, storage systems that support synchronization and file sharing are required. The requirement is not just storing, sharing, and synchronizing the data but

performing these operations securely, i.e., the confidentiality, availability and integrity of the data should be maintained. The question of confidentiality, availability and integrity of data comes into the picture when the user's data is being stored in third party storage systems like the cloud storage services.

The primary motivation for this thesis is the increasing usage of cloud-based systems. Several large companies use highly scalable cloud storage and computation solutions internally, and some also offer their cloud services for commercial use and as a product to individuals. This results in the cloud being of significant importance in modern society with regards to how information is stored and communicated. Therefore, we believe that the use and importance of cloud-based solutions will be increasing significantly in years to come. Therefore, providing a secure service which handles sensitive data in cloud storage services is an interesting and new problem domain. The main objective of the thesis is to design cloud-based file sharing services that maintain data confidentiality against cloud and by implementing a fine-grained access control mechanism as well as ensuring data integrity. In other words, we want to design a secure file sharing service in which the data owners and data user can share data securely without direct interaction between them. In addition, we want to delegate the computational tasks to the cloud storage service without allowing him to have access to either the plaintext data or the access control keys used. Achieving this security is the obvious objective, but this has to be done in the context of maintaining compliance with the customer's security policies and meeting various regulatory and legislative requirements.

Our main contributions in this thesis are: 1) to design trusted third party service that enables users to share data over any web-based cloud storage platform while data security

is preserved. This service protects the confidentiality of the communicated data and it can be employed locally or remotely; 2) to construct a new multi-authority CP-ABE scheme that achieve fine grained access control. Based on multi-authority CP-ABE [18], we realize efficient fine grained access control. Different from [18], we support many-write-many-read for users(which means after the owner creates one encrypted file on the storage server, other users with appropriate attributes can also update the encrypted file at a later time without any help from files' original owners) instead of 1-write-many-read; 3) to propose an efficient revocation approach for the proposed multi-authority CP-ABE scheme. Basing on the revocation method [19], we realize efficiently immediate user/attribute-level revocation while achieving both the backward and forward secrecy. In addition, we delegate the re-encryption right to cloud proxy servers.

1.3 Organization of the thesis

The rest of this proposal is organized as follows: chapter 2 explains the basics of cloud computing and surveys the threats to data in the cloud model. Moreover, it explains the cloud storage services in more detailed and provides a detailed analysis about data security threats in this model. Chapter 3 explains previous work done for securing the data. While chapter 4 discusses the proposed approach for securing the data in the cloud - based data sharing services. Finally, in chapter 5, we will conclude the thesis and list some directions for future work. There is an appendix at the end of the document. Appendix A shows CP-ABE background.

2 Background

2.1 Cloud Basics

Nowadays cloud computing has become a significant technology trend either in the industrial or the academic field, and most of the experts expect that cloud computing will reshape -information technology (IT) processes 'and the IT market place. In Cloud Computing, users connect to the ’Cloud’, which appears as if it is a single entity as opposed to multiple servers. In this model, users can remotely store their data so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources [1].Although this pay-per-use model of the cloud services brings significant savings for users and offers flexibility and scalability in terms of capacity and performance, it involves giving the cloud service provider (CSP) some form of control over the user's data.

In spite of the wide spread of cloud computing, different people evoke different perceptions about it. To some, it refers to accessing software and storing data in the “cloud” representation of the Internet or a network and using associated services. To others, it is seen as nothing new, but just a modernization of the time-sharing model that was widely employed in the 1960s before the advent of relatively lower-cost computing platforms. These developments eventually evolved to the client/server model and to the personal computer, which placed large amounts of computing power at people’s desktops and spelled the demise of time-sharing systems

To formally describe cloud computing, the definition by the National Institute of Standards and Technology (NIST) is as follows:

“Cloud computing is a model for enabling 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." [1]

From the definition, we can conclude that the primary idea in cloud computing is that organizations no longer manage or own their data, but have it delivered as a service by a CSP. Over the last years, there is a trend to outsource more and more of data to external parties.

The rest of the chapter is organized as follows: clouds characteristics are presented in subsection one. In subsection two, cloud service models are described, and description of cloud deployment models is carried out in subsection three. While in subsection four, a detailed explanation of cloud architecture is offered and an illustration of cloud storage is presented in subsection five. Finally, subsection six presents benefits and drawbacks of cloud.

2.1.1 Cloud key characteristics

According to NIST definition of cloud computing, the cloud model is composed of five essential characteristics. These characteristics are explored in the following lines.

a. On-Demand Self-Service: Cloud customer can make use of cloud resources without any human interaction between them and the cloud service provider (CSP).In addition; they can schedule, manage and deploy any of cloud services such as computation and storage when needed. This leads to reduction in the personnel overhead of the cloud provider, cut in costs of the offered services [20].

b. Broad Network Access: Cloud services are accessible over the network via standardized interfaces which enables users to access the services not only by complex devices such as personal computers, but also by light weight devices such as smart

phones. In addition, the lowered cost of high-bandwidth network communication to the cloud provides access to a larger pool of IT resources that sustain a high level of utilization [20].

c. Location-Independent Resource Pooling: The cloud must be able to meet consumer's needs from resources. To do so, the cloud use a technique called "virtualization", which enables the cloud provider to pool his computing resources. This resource pool enables the sharing of virtual and physical resources by multiple consumers. As stated by NIST, "There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter)."[20].

d. Rapid Elasticity: It is the ability of the cloud to allocate and release resources quickly and efficiently in order to meet the requirements of the self-service characteristic of cloud computing. This automated process decreases the procurement time for new computing capabilities when the need is there, while preventing an abundance of unused computing power when the need has subsided [20].

e. Measured Service: Cloud computing can dynamically and automatically measure the used resources by cloud customers. These measurements can be used to bill the customer and provide them with a payment model based on "pay-per-use." The NIST view of measured service is "Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource

usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service."[20].

2.1.2 Cloud service models

One of the main principles of Cloud Computing is the `as-a-Service' paradigm in which some services are offered by a Cloud Service Provider (CSP) to customers for use. These offered services are often categorized using the SPI Service Model. This model represents the different layers/levels of service that can be offered to users by cloud service providers over the different application domains and types of cloud available. Clouds can be used to provide as-a-Service: software to use, a platform to develop on, or an infrastructure to utilize [21]. Figure 2.1[21] summarizes the SPI Service Model.

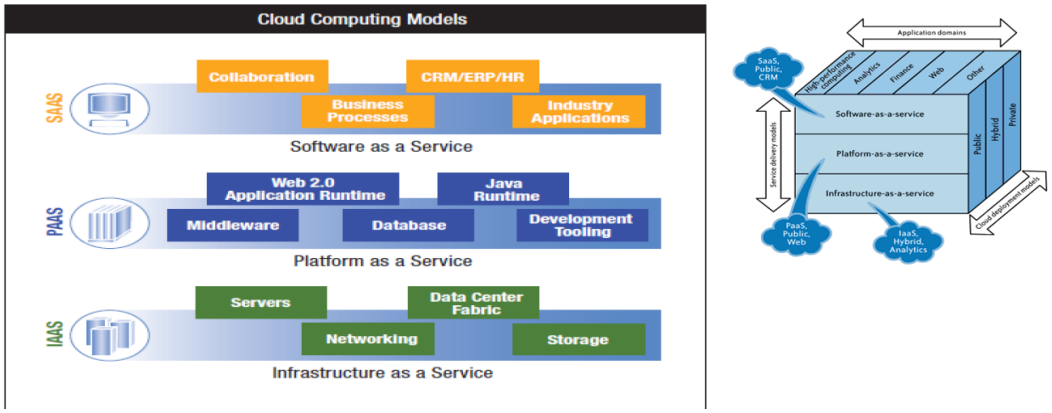

Figure 2-1 SPI Service Model [21]

2.1.2.1 Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) is a service that can provide the functionalities of a whole infrastructure including storage, networks, any platform and any number of desktops. The customers can make use of this service by configuring a virtual machine on the infrastructure, on which an operating system is installed. They deploy the middleware for communication with other applications, and install the CRM software. There is no need to buy extra servers, when the application needs more resources, extra CPUs and

storage can be assigned via a web interface or via the CSP, the customers only pay for the used computing power and data storage [4].

According to NIST, Infrastructure as a Service is defined as: “The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems; storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).”[20]. Examples of IaaS are Amazon Elastic Compute Cloud and Terremark Enterprise Cloud [20].

2.1.2.2 Platform as a Service (PaaS)

In the Platform as a Service (PaaS) model, the CSP offers a development platform on top of the services delivered with IaaS. The CSP offers a development platform, on which applications can be built. In other words, software developers can develop their application through virtual development platform, accessible via a Web browser, without the need to install the software building tools on their own computer. This helps the developers to later distribute or deploy their apps to the cloud easily. In order to avoid confusion of this service with SaaS, it is good to imagine it as a cloud OS. The providers of the service enable its users to install their applications on a platform, which can provide any operating system or even emulate various types of hardware [20].

According to NIST, PaaS is described as follows: “The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations."[20]. Examples of PaaS platforms are Amazon Elastic Beanstalk, Microsoft Azure Platform, Force.com and Google App Engine [20].

2.2.3 Software as a Service (SaaS)

SaaS is a very popular service in which cloud service providers deliver software applications over the Web. A SaaS provider deploys their software, which is hosted on their own server infrastructure or use another vendor's hardware, on user's demand .This operation is usually done using a licensing model where applications may be licensed directly to an organization, group of users or, a user or, or through a third party that manages multiple licenses between user organizations, such as an ASP. The user then can be able to access the applications through any well defined and Internet device, which is most probably a Web browser.

According to NIST, Cloud Software as a Service (SaaS) is defined as follows: "The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based e-mail).The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings."

The pioneer of providing SaaS was a company called Salesforce.com. Another common example of a SaaS provider is Google with its email and office tools like word

processor, spreadsheet or calendar. It could be any type of application and users can even create their own, hosting them on the provider's servers [20].

Benefits of the SaaS Model:

1- It reduces the cost licensing, management hardware, and other resources required to internally host the application by outsourcing the application hosting to an independent software vendor (ISV).

2- It increase the control over the use of the software — by limiting the distribution of unlicensed copies and allowing the software vendor greater upgrade and patch management control.

3-It enables the provider to control and create multiple revenue streams with a one-to-many model leading to reduction in the duplication of software packages and overhead [20].

Table 2-1[20] shows the three primary SPI framework services, paired with an example of the service the vendor supplies for that layer. While figure 2.2[22] shows the cloud computing stack.

SPI FRAMEWORK SERVICE	DESCRIPTION	VENDOR EXAMPLE
IaaS	Shared Internet infrastructure, such as servers and storage	Amazon EC2 and S3, Sun Microsystems Cloud Services, Terremark, Dropbox
PaaS	Application platform that provides developers with quick deployment	Google App Engine, force .com (from salesforce.com), Microsoft Azure
SaaS	Stateless cloud-enabled multiple-instance applications on a pay-per-use pricing model	Zoho Suite, Apple's MobileMe, Google Docs

Table 1[20]: SPI Services Delivery Vendors

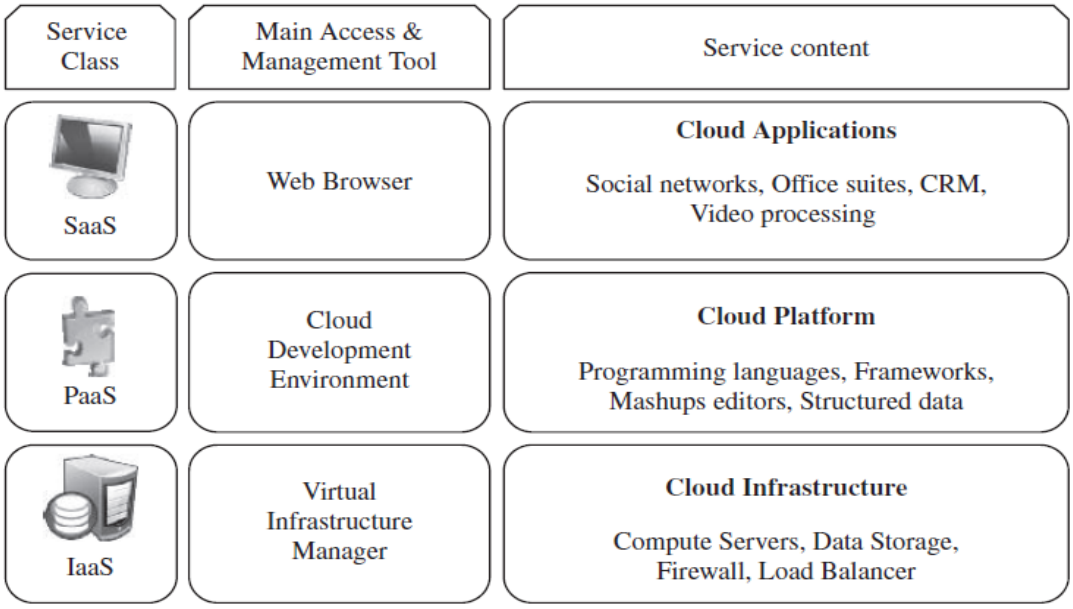

Figure 2-2[22]: Cloud Computing Stack

2.1.3 Cloud deployment models

Cloud computing offers four basic deployment models. These deployment models are the public, hybrid, community and private cloud. Each of these models has its own characteristics. These characteristics can be described as are who owns the infrastructure, who manages the infrastructure, where the infrastructure located is, and who accesses the cloud services. Figure 2.3[22] shows types of clouds based on deployment models.

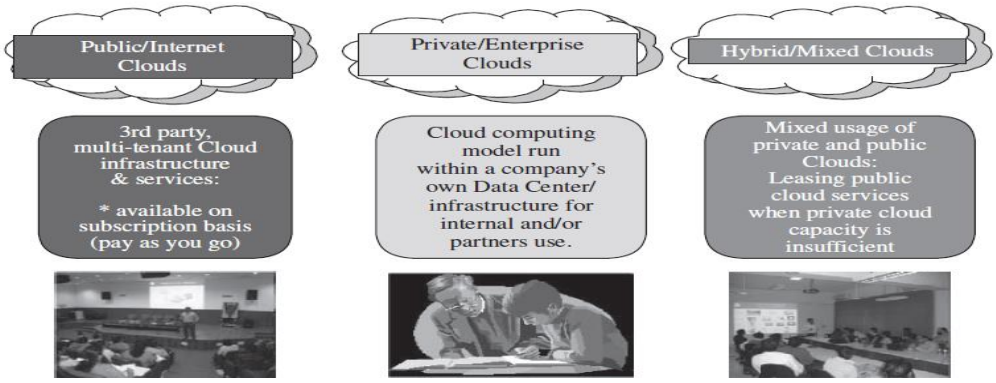

Figure 2-3[22]: Types of clouds based on deployment models

2.1.3.1 Private cloud

It is the cloud infrastructure in which the services are completely dedicated to one customer/organization. In other words, resources are not shared by other entities; it is only dedicated to the one customer. Moreover, it can be referred to as internal cloud or cloud computing on private networks, which are built for the exclusive use of one customer, providing full control over data, security, and quality of service. Furthermore, in this type of cloud, the users are considered as trusted by the organization, in which they are either employees, or have contractual agreements with the organization. [22]

In addition, according to NIST, private cloud is described as follows as:" a cloud infrastructure operated solely for an organization, managed by the organization or a third party and existing either on premise or off-premise. The private cloud is typically hosted within the boundaries of the owner organization."[20].Figure 2.4[23] shows an example of private cloud.

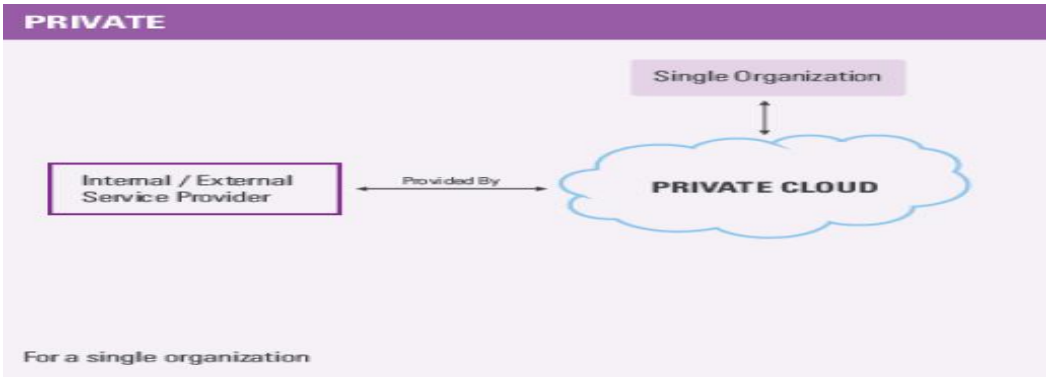

Figure 2-4[23]: Private Cloud

2.1.3.2 Public cloud

It is the cloud infrastructure in which the services are offered to the general public or a large industry group and is owned by an organization selling cloud services. In other words, resources are shared among all customers. This means that cloud users are

considered to be un-trusted, where they are not tied to the organization as employees and that the user has no contractual agreements with the provider.

In addition, according to NIST, public cloud is described as follows as:" The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider."[20]. Figure 2.5[23] shows a public cloud.

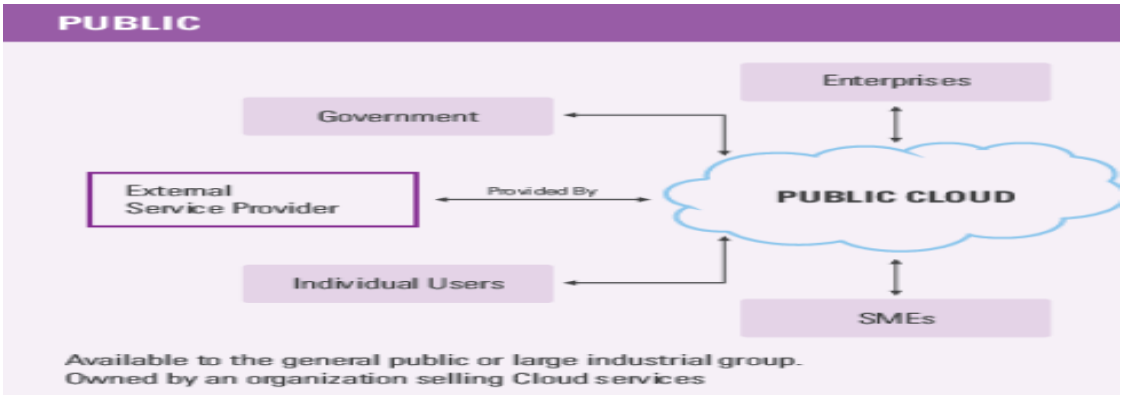

Figure 2-5[23]: Public Cloud

2.1.3.3 Hybrid cloud

Hybrid cloud is a combination of public, private, and community clouds. It leverages the capabilities of each cloud deployment model. In addition, each part of a hybrid cloud is connected to the other by a gateway, controlling the applications and data that flow from each part to the other. Moreover, it allows the organizations to manage some supporting resources in-house and has others provided externally. Furthermore, the users of hybrid clouds can be considered as trusted and un-trusted. Un-trusted users are prevented to access the resources of the private and community parts of the hybrid cloud [22].

In addition, according to NIST, hybrid cloud is defined as “a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).”[20]. Figure 2.6[23] shows a hybrid cloud.

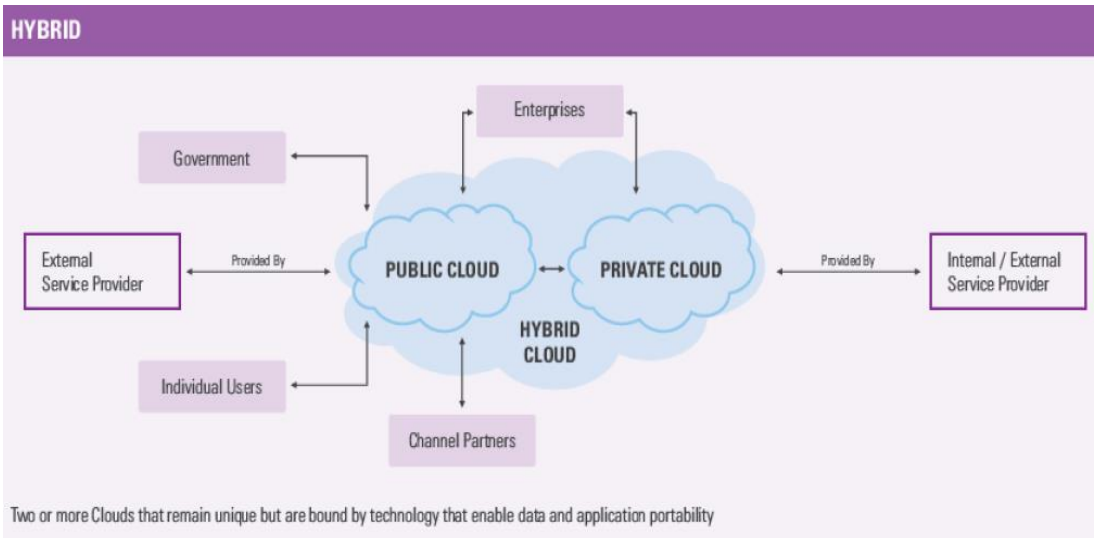

Figure 2-6[23]: Hybrid Cloud

2.1.3.4 Community cloud

It is the cloud infrastructure that is shared by several organizations and supported by multiple companies. Moreover, the shared cloud may reside on any member’s premises, or even on a third-party site, and managed either by the organizations or a third party. Furthermore, community users are also considered as trusted by the organizations that are part of the community. [20]

In addition, according to NIST, community cloud is defined as:"The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one

or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises."[20]. Figure 2.7[23] shows a community cloud.

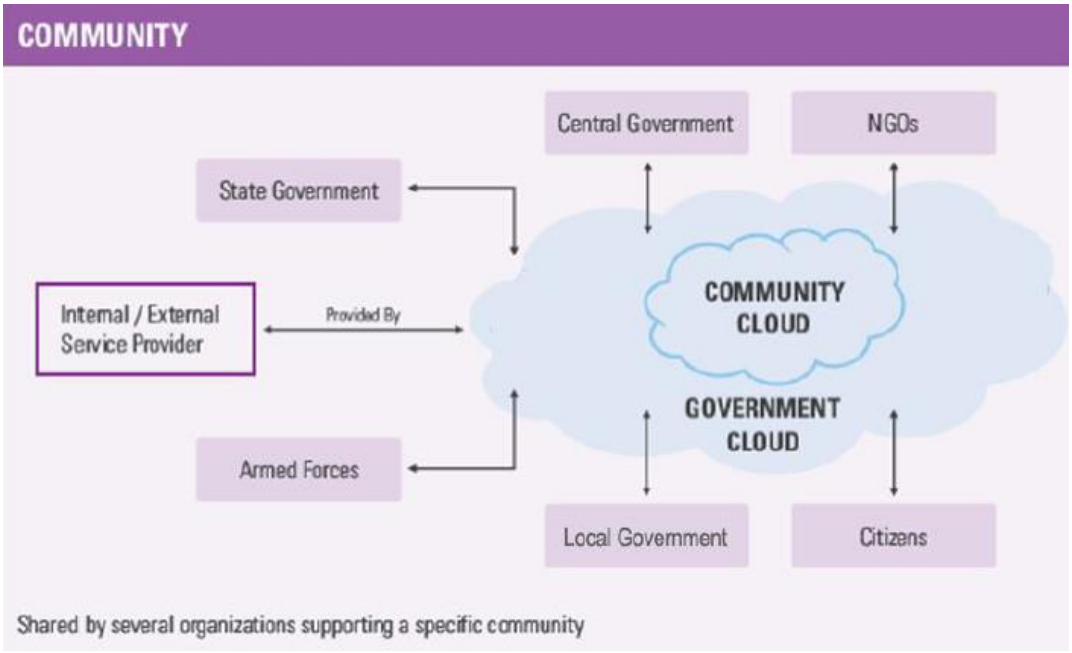

Figure 2-7[23]: Community Cloud

2.1.4 Cloud Architecture

The cloud is composed of a massive network of servers or even individual PCs interconnected in a grid. These computers operate in parallel, merging the resources of each computer to produce a power similar to that of supercomputers. In other words, the cloud is simply a collection of computers and servers that are publicly accessible via the Internet. These machines (computers and servers) can run any combination of operating systems; it's the processing power of the machines that matter. Although this architecture appears to be simple, it does require some intelligent management to connect all those computers together and assign task processing to multitudes of users. Figure 2.8 [24], shows the architecture behind a cloud computing system. As shown in the figure, it begins with user interface for the user to interact with the cloud and selecting a task or

service (either starting an application or opening a document).After selecting the required service, a request is passed to the system management. In the system management, correct resources are found and then the appropriate provisioning services are called. Later on, these services choose the necessary resources in the cloud, launch the appropriate web application and either creates or opens the requested document. After that, the web application is launched and then the system's monitoring and metering functions track the usage of the cloud so that resources are apportioned and attributed to the proper user(s).[24]

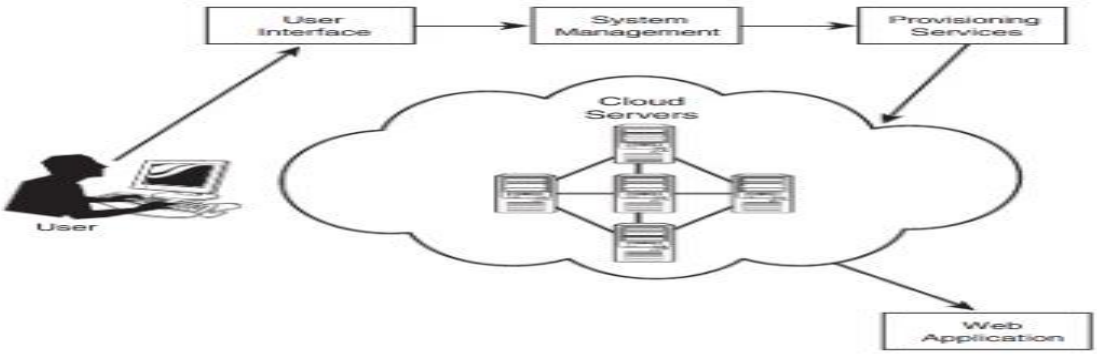

Figure 2-8[24]: Architecture behind the cloud computing system

2.4.1.1 Cloud computing reference architecture

In this section, we are going to explore the NIST cloud computing reference architecture. This reference architecture identify the five major actors, their activities, and functions in cloud computing. These actors are: cloud consumer, cloud provider, cloud carrier, cloud auditor, and cloud broker. Each of these actors is an entity (a person or an organization) that participates in a transaction or process or performs tasks in cloud computing. Figure 2.9[25] shows cloud computing reference architecture.

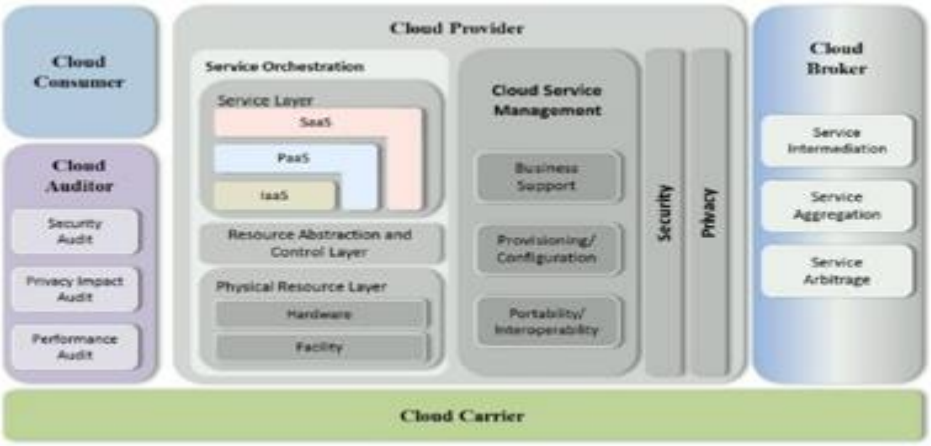

Figure 2-9[23]: The Conceptual Reference Model

1. Cloud Consumer

The cloud consumer is the main actor that makes use of the cloud computing services. It can be a person or an organization that maintains a business relationship with, and uses the service from, a cloud provider. Moreover, a cloud consumer navigates through the service catalog offered by a cloud provider in order to request, set up, and use the appropriate services.

In order for the cloud consumer and provider to set a framework for their relationship, they make use of Service-Level Agreements (SLAs).In these SLAs, the cloud consumers specify their required technical performance requirements from quality of service, security, and remedies for performance failures. On the other hand, cloud provider uses these SLAs to impose a set of restrictions or limitations, and obligations that cloud consumers must accept.

The cloud consumers can make use of the three main services offered by the cloud providers-SaaS, PaaS, and IaaS. The SaaS consumers can be either an organization which provide its members with access to software applications or an end users who directly use software applications, or software application

administrators who configure applications for end users. These customers can access applications through the network. While the PaaS customers employ the tools offered by cloud providers to develop, test, deploy, and manage the PaaS applications hosted in the cloud. These consumers can be application developers who design and implement application software, application testers who run and test applications in a cloud-based environment, application developers who publish applications into the cloud, or application administrators who configure, monitor, and manage applications deployed in a cloud. For IaaS consumers, they use the virtual computers, network-accessible storage, network infrastructure components, and other fundamental computing resource offered by the cloud. These consumers can be system developers, system administrators, or IT managers who are interested in creating, installing, monitoring, and managing services and applications deployed in an IaaS cloud [25].

2. Cloud Provider

A cloud provider is the entity in charge of making a service available to interested parties. It can be either a person or an organization that acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes the arrangements to deliver the cloud services to cloud consumers through network access.

The cloud provider offers three main service models which are SaaS, IaaS, and PaaS. In SaaS cloud, the cloud provider is mainly responsible for deploying, configuring, maintaining, and updating the operation of the software applications on a cloud infrastructure. However, it's mostly responsible for security, managing the applications, and the cloud infrastructure. On the other hand, the SaaS cloud

consumers are offered limited administrative control over their applications by the provider. While in the PaaS cloud, the provider is mainly in charge of managing the computing infrastructure for the platform and running the cloud software that provides the components of the platform. In addition, the provider supports the development, deployment, and management process of the PaaS cloud consumer by providing tools such as integrated development environments (IDEs), software development kits (SDKs), and deployment and management tools. On the other hand, the provider offers the PaaS cloud consumer some control over their applications and possibly over some of the hosting environment settings, but no or limited access to the infrastructure underlying the platform such as network, servers, operating systems (OSs), or storage. For IaaS, the cloud provider owns the physical computing resources underlying the service, including the storage, servers, networks, and hosting infrastructure. In addition, it runs the cloud software necessary to render the necessary computing resources to the IaaS cloud consumer through a set of service interfaces and computing resource abstractions, such as virtual machines and virtual network interfaces. In return, the provider provides the IaaS consumer access to more fundamental forms of computing resources and thus he/she has control over more software components in an application stack, including the OS. The IaaS cloud provider, on the other hand, has control over the physical hardware and cloud software that make the provisioning of these infrastructure services possible [25].

3. Cloud Auditor

A cloud auditor is an entity that performs an independent task of examining the cloud service controls with the intent to express an opinion thereon. Moreover, the

auditor evaluates the services provided by a cloud provider such as security controls, privacy impact, and performance in order to verify their conformance to standards through a review of objective evidence [25].

4. Cloud Broker

A cloud broker is a party that manages the use, performance, and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers. The cloud consumers may ask for cloud services from a cloud broker, instead of directly contacting a cloud provider. In general a cloud broker is divided into three main categories: Service Intermediation, Service Aggregation, and Service Arbitrage.

a) Service Intermediation: In this category, the cloud broker improves a given service by enhancing some specific capability and providing value-added services to cloud consumers. These enhancements can take many forms as identity management, enhanced security, managing access to cloud services, performance reporting, etc.

b) Service Aggregation: In this category, the cloud broker merges multiple services into one or more new services where it provides data integration and ensures the secure data movement between the cloud consumer and multiple cloud providers.

c) Service Arbitrage: This category is similar to service aggregation except that aggregated services are not fixed. Service arbitrage means that a broker has the flexibility to choose whatever services from multiple agencies. For example, the broker can use a credit-scoring service to measure and select an agency with the best score [25].

5. Cloud Carrier

A cloud carrier plays an intermediate role between cloud consumers and cloud providers in providing connectivity and transport of cloud services. Cloud carriers provide consumers access to services through network, telecommunication, and other access devices. For example, cloud consumers can obtain cloud services through network access devices, such as desktop computers, laptops, mobile phones, and other mobile Internet devices (MIDs) [25].

2.1.5 Cloud Storage

One of the main uses of cloud computing is data storage. Cloud storage is considered to be an online virtual distributed storage provided by cloud computing vendors. In other words, data stored in cloud is stored on multiple third-party servers, rather than on the dedicated servers used in traditional networked data storage. In addition, it appears to users as if the data is stored in a particular place with a specific name. But that place does not exist in reality. It is just a virtual name used to reference virtual space carved out of the cloud. In reality, the user's data could be stored on any machine in the cloud and the user can access it via a web service interface, or a web based user-interface. Security and finance are the two main advantages of cloud storage. Financially, cloud users can save space, cost and complexity of installing their own storage devices. Moreover, this cloud storage is much cheaper than dedicated physical resources connected to a personal computer or network. As for security, data stored in the cloud is secure from hardware crashes or accidental erasure, because it is duplicated across multiple physical machines; since multiple copies of the data are kept continually, the cloud continues to function as

normal even if one or more machines go offline. If one machine crashes, the data is duplicated on other machines in the cloud [26] [27].

2.5.1 Overview on Cloud Storage products

There are many commercial cloud storage services offered by different vendors. However, we have to differ between basic cloud storage services and advanced Basic cloud storage services.

Basic cloud storage services: they are services mainly offer storage space and some limited functionalities as file sharing. These services are not designed to be accessed directly by users but are embedded into custom software using application programming interfaces (API). Examples of such basic cloud storage services are Amazon S3 [28] and Google Cloud Storage [29].

Advanced cloud storage services: They are services that employ the basic cloud storage services functionalities, however, they differ from basic cloud storage services in that they provide interfaces such as client or web applications for allowing all types of user to use them. Moreover, this interface allows them to offer more functionalities than basic cloud storage services as file sharing, synchronization. An example of such advanced cloud storage services is Dropbox [30].

In the following few lines, we are going to give examples, and brief descriptions, of two basic cloud storage services and one advanced cloud storage services.

a. Amazon Simple Storage Service (Amazon S3)

Amazon S3 is one of the cloud storages offered by Amazon. It provides data storage and retrieval facilities with any amount at any time via web services interfaces, such as AWS Management Console. Moreover, it stores data as objects within buckets. An object

is a file and any optional metadata that describes this file. Amazon S3 best fits in cases of large file, (e.g. up to 5 terabytes of data). But in case of small size files, it is better to use other Amazon's data storage. Furthermore, for managing these files , Amazon S3 use NoSQL database solutions instead of using -traditional relational database systems or MySQL that are very complex and inapplicable in some cases. In addition, to reduce complexity, Amazon S3 has purposely minimal functionality, so data can only be written, read and deleted. Every object/file is stored in a bucket and retrieved via a unique key. It supports storing 1 byte to 5 terabytes of data, and the number of files to be stored is unlimited [28].

b. Google Cloud Storage

Google Cloud Storage is a service offered by Google's cloud for developers to write and read data. In addition to data storage, Google's cloud offered users direct access to Google's networking infrastructure, authentication and sharing mechanisms. Moreover, it is accessible via its REST API or by using other tools provided by Google as Google storage manager and gsutil.

In order for Google Cloud Storage to be more efficient and reliable in storing, sharing, and managing data , it provides the following features and capabilities: High capacity and scalability ,Strong data consistency , OAuth 2.0 authentication , Cookie-based authenticated browser downloads, Google APIs Console Projects , Google account support for sharing, REST API, and Bucket locations.

Google Cloud Storage uses ACLs for controlling access to the objects and buckets. Every time a user requests to perform an action on an object, the ACL belonging to that object determines whether the requested action should be allowed or denied [29].

c. Dropbox

Dropbox is considered to be an advanced cloud storage services that have the same functionalities of basic cloud storage services for the actual storage of data, but provide interfaces such as client or web applications for allowing user to directly use the service. Dropbox can also be understood as a file hosting service that allows users to store and share their data across the internet. It makes use of file synchronization for sharing files and folders between users' devices. In addition, it provides user clients for many operating systems on desktop machines, such as Microsoft Windows, Mac OS X and Linux, and also on mobile devices, such as Android, Windows Phone 7, iPhone, iPad, WebOS and BlackBerry. Moreover, users can also access their data through a web-based client when no local clients are installed [30].

Dropbox makes use of Amazon's cloud storage, namely Amazon S3, as their data storage. Dropbox founders claim that it has a solid security for users' data, and they use the same security solutions as banks. For synchronization, Dropbox uses SSL file transfer protocol, and the stored data are encrypted at the server side using AES-256 encryption [31].

2.5.2 Cloud Storage Security Requirements

In the process of storing data to the cloud, and retrieving it back from the cloud, there are mainly three elements involved, which are the client, the server (CSP) and the communication link between them. In order to make sure that the data is stored securely, all the three elements must have a solid security. For the client, he/she has to make sure that unauthorized party can access his machine. While for cloud storage provider (CSP), he has to make sure that data have confidentiality, integrity and availability in rest. Last

but not least, the communication between client and server must be performed through a secure channel, i.e. the data must have confidentiality and integrity during its transfer between server and client. One of the ways to achieve secure communication is having a cryptographic protocol, such as SSL [32].

2.1.6 Benefits and drawbacks of cloud

a) Benefits

The benefits of using cloud computing are varied. They include a cloud's inherent flexibility and resiliency, the potential for reducing costs, availability of very large amounts of centralized data storage, means to rapidly deploy computing resources, and scalability [20].

1) Flexibility and Resiliency

Cloud computing offers much more flexibility than past computing methods. They provide their customers with the ability to choose among a number of computing and storage resource configurations at different capabilities and costs what fits their requirements.

On the other hand, resiliency can be achieved through the availability of multiple redundant resources and locations. In addition, with the advancement in autonomic computing, self-management and self-healing mechanisms helps in ensuring the increased reliability and robustness of cloud resources [20].

2) Reduced Costs

Cloud technology is paid incrementally, saving organizations money. Instead of buying large servers for storage and backup, the organization can hire what it need from the cloud. This leads to reduction in capital costs and saving the money

for managing operating expenses .Another factor to be considered is that client organizational support and maintenance costs are reduced dramatically because these expenses are transferred to the cloud provider, including 24/7 support.

All in All, cloud computing offers reductions in system administration, energy costs, software licensing fees, provisioning expenses, and hardware costs [20].

3) Centralization of Data Storage

The cloud provides much data storage recourses than that available in local, corporate computing systems. Moreover, there is flexibility in increasing or decreasing these cloud storage resources according to operating cost adjustments. This form of centralization of storage infrastructure results in cost efficiencies in - utilities, trained personnel, and real-estate. In addition, it will be much easier to implement and monitor data protection schemes in a centralized system than on large numbers of computing platforms. However, there is one main disadvantage of having centralization of data storage of large amounts of sensitive information stored in a centralized; it provides an attractive target for hackers or criminal organizations to gain access to critical information by focusing on a central repository [20].

4) Reduced Time to Deployment

As cloud provides means to make use of powerful computational resources in a short period and with large amounts of storage without the need to require sizeable initial investments in hardware, software, and personnel. This leads to rapid provisioning that can accomplished at relatively small cost and offers the customers access to advanced technologies that are constantly being acquired by the cloud provider [20].

5) Scalability

Cloud computing allows the customers, with limits, to provision their computational resources in order to meet their demands, either increasing or decreasing .This approach provides an alternative to in-house systems that are used only during peak periods and stay with partial capacity most of the time [20].

b) Drawback

Although cloud computing provides a number of benefits and advantages over the previous computing paradigms, there are still a number of challenges. These challenges include: performance, security and privacy, control, bandwidth costs, and reliability [33].

1) Performance

The performance can a major issue in case of intensive transaction-oriented and other data-intensive applications because cloud computing may lack adequate performance. In addition, users who are long distance away from their providers may suffer from high latency and delays [33].

2) Security and Privacy

Customers are always worried about vulnerability to attacks, when their data is sent to the cloud as they have no control over it [33].

3) Control

Some customers are worried because cloud computing providers have full control of the platforms. However, cloud computing providers do not typically design customized platforms for companies. Therefore, all cloud computing providers will have control over their customer's platforms [33].

4) Bandwidth Costs

Although most companies using cloud computing save money spend on hardware and software, they need much more money to acquire higher network bandwidth. Bandwidth cost varies depending on application type. It can be low for smaller Internet-based applications, while could be significantly high for data-intensive applications [33].

5) Reliability

Cloud computing do not offer high reliability all the time. There were some cases where cloud suffers from reliability problems, a few-hours outages [33].

2.1.7 Cloud Service Provider infrastructure

In the following sections, we are going to describe the technical infrastructure of a cloud service provider (CSP) .In Section 2.7.1.1, we will give a detailed description to virtualization in cloud computing. As for Section 2.7.1.2, we will describe how data is stored in cloud computing. Finally, in Section 2.7.1.3, we will describe how this data storage is combined with a virtualized environment.

2.7.1.1 Virtualization

In recent few years virtualization has had a significant impact on cloud computing. Virtualization in cloud enables multiple operating systems and applications to run concurrently and in isolation from each other on a single physical machine. Each operating system with its applications represents a virtual machine. These virtual machines (VMs) share the physical resources of the single physical machine leading to better utilization, optimization and resource efficiency. In addition, virtualization allows resources to be automatically allocated when and where needed and for dynamic provisioning and de-provisioning. Moreover, CSP offers two options in delivering a VM:

a client can create its own VM with operating system and configuration, or the CSP delivers a standard VM with pre-installed operating system [34].

Figure 2.10[34] illustrating the concepts of multiple virtual machines running on and utilizing a single host operating system and the physical computer hardware.

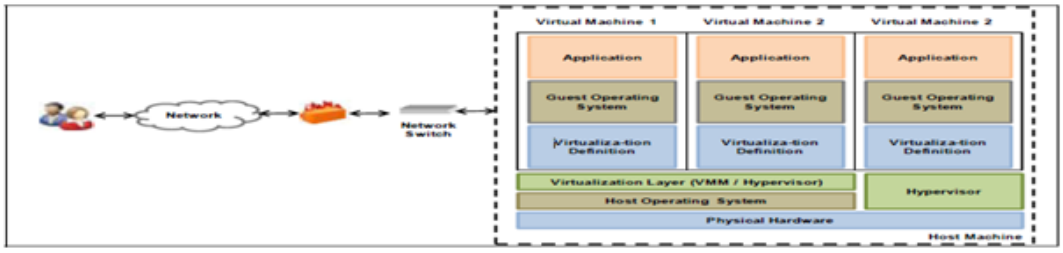

Figure 2-10[34]: Virtual Environment

The primary benefits of virtualization are a reduction in costs, server consolidation and utilization. Figure 2.11[34] shows the major benefits including disaster recovery and service continuity (availability), easier or quick deployment, seamless portability and migration, increased flexibility and service agility, reduced downtime, easier and quicker developments and testing, ease of management and administration, isolation, and improved security and control.

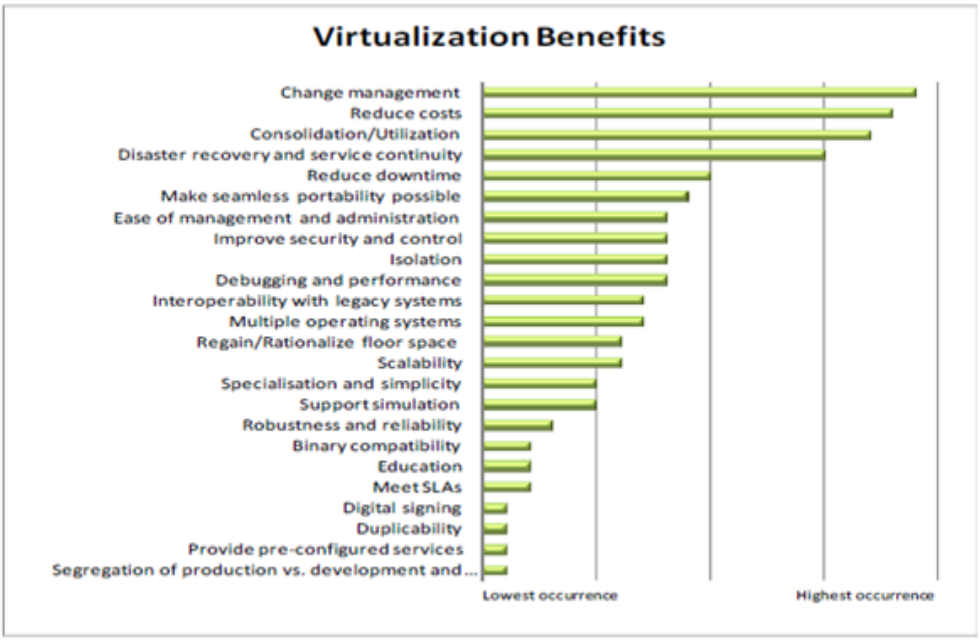

Figure 2-11[34]: Virtualization Benefits

The management of VMs on a physical machine is performed by the hypervisor, which is also known as virtual machine manager. The hypervisor presents a virtual operating platform to the guest operating systems and manages the execution of the guest operating systems. In addition, it gives the guests operating systems the impression that they are running on physical hardware, by assigning processing capacity, data storage and networking facilities. Examples of hypervisors are VMware [34].

2.7.1.2 Data storage

In data centers, data is not stored on server's hard disk, but they are stored on large storage clusters. An example of these storage clusters is Storage Area Network (SAN). A SAN is a dedicated storage network that provides access to consolidated, block level storage. In addition, these SANs are used to make storage devices that are accessible to servers appear as if they are locally attached to the operating system. Moreover, A SAN

has its own network of storage devices that are generally not accessible through the regular network by regular devices [35].

It is also important to note that, the SAN itself does not provide file abstraction, but only block-level operations. In contrast to SAN, there is Network Attached Storage (NAS), which use uses file-based protocols such as NFS or SMB/CIFS where the storage is remote, and computers request a file rather than a disk block [35].

2.7.1.3 Data storage virtualization

The customers of cloud think that their operating system writes directly to their dedicated hard disk, while this is not true. In reality, the hypervisor converts customer's operations to a virtual disk. These virtual disks are often referred to as LUNs (Logical Unit Numbers).A LUN can be defined as a logical reference to a portion of a storage subsystem , which can comprise a disk, a section of a disk, a whole disk array, or a section of a disk array in the subsystem. This logical reference, when it is assigned to a server in the SAN, acts as a physical disk drive that the server can read and write to. Using LUNs simplifies the management of storage resources in the SAN [35].

In addition, it is important to know that the CSP usually move data to different countries. There are two main reasons that make CSP do this. Firstly, dynamically spreading data over multiple locations leads to more redundant and delivers higher availability. When one data center becomes unavailable, other data centers can take over the tasks. Secondly, storing and processing data at different locations leads to more efficiency, when data can be stored or processed at a location with spare capacity or low processing (e.g. electricity) costs for specific moments, e.g. when solar power is available in overcapacity [35].

2.2 Cloud Security Issues

2.2.1 Overview

Although cloud computing saves enterprises millions and encourages more innovation by simplifying access to scalability for greater numbers of developers and organizations, it suffers from multiple security threats that come from the way cloud computing infrastructures are constructed. As a result, security issues have many guises both technical and socio-technical and covering all these security issues in-depth within the cloud is an impossible task. Therefore, this chapter will cover only two classifications of security issues. The first classification will present threats to Cloud Computing according to European Network and Information Security Agency (ENISA) [36], which classifies security risks related to cloud computing into three main categories: Policy and Organizational, Technical, and Legal. While the second classification will present seven top threats to cloud computing according to Cloud Security Alliance [37], which is a non-profit organization that seeks to promote the best practices for providing security assurance within the cloud computing landscape. These threats are Abuse and Nefarious Use of Cloud Computing, Insecure Application Programming Interfaces, Malicious Insiders, Shared Technology Vulnerabilities, Data Loss/Leakage, Account and Service Traffic Hijacking, and Unknown Risk Profile.

2.2.2 Top Security Risks Categories

The security risk should always be understood in relation to overall business opportunity and appetite for risk – sometimes risk is compensated by opportunity. Therefore, the risks of using cloud computing should be compared to the risks of staying with traditional solutions, such as desktop-based models. Because cloud

services are not only about convenient storage, accessible by multiple devices, but it includes benefits such as more convenient communication and instant multi-point collaboration. Moreover, it is important to note that the level of risk may vary significantly with the type of cloud architecture.

In addition, it is possible for some risks to transfer to the cloud provider from

Cloud customer and these risks should be taken in consideration against the cost benefit received from the services. However, not all risks can be transferred to cloud but if a risk is transferred to the cloud, it may lead to the failure of the business, serious damage to reputation or legal implications. At that moment, it would hard or even impossible for any other party to compensate for this damage.

In the following sections, we will go through the three main categories of security risks related to cloud computing. In each category, we will highlight some the associated risks.

2.2.2.1 Policy and Organizational

Figure 2.12[38] shown below illustrates the policy and organization risks.

Figure 2-12[38]: Policy and Organizational Risks

a) Lock In

Lock-in means that the data is locked to a certain CSP because there are no standards followed in data formats or in services interfaces that could guarantee data,

application and service portability. This make the customer migration from one provider to another or migration of data and services back to an in-house IT environment a difficult task. This implies that there is not much portability or interoperability for data and services provided by the current cloud providers. Moreover, lock-in is considered to be a very high risk for a company as it may cause the following vulnerabilities: lack of standard technology and solutions, poor provider selection, lack of supplier redundancy, and lack of completeness and transparency in terms of use. These vulnerabilities may affect the following assets: company reputation, personal sensitive data, personal data, and service delivery – real time services. In addition, they may also cause catastrophic business failures that could drive the cloud provider to go bankrupt [38].

b) Loss of Governance

While using cloud infrastructures, the client cedes control of a number of issues which may affect security to the cloud provider leading to loss of governance and control. Although there are SLAs between cloud provider and client, these SLAs do not offer clear promise to provide such issues on the part of cloud provider. This leads to a gap in security defenses which in turn leads to loss of governance and control. This loss of governance and control severely affect the organization's strategy and ability to meet its mission and goals. In addition, it make it impossible to comply with security requirements leading to lack of confidentiality and integrity, un-availability of data, and a deterioration of performance and quality of service, not to mention compliance challenges. Furthermore, this shall affect the company reputation, customer trust, employee loyalty and experience, personal sensitive data, and service

delivery – real time services [38].

c) Cloud Service Termination or Failure

In any IT market, competitive environment, inadequate business plan, and lack of financial support may push some cloud providers to go out of business or at least force them to restructure the services they offer. In other words, some cloud computing services may terminate in the short or medium term due to any of the reasons stated above. This is considered to be a medium risk that may lead to a loss or deterioration of service delivery performance, and quality of service, as well as a loss of investment.

Moreover, failures in the services outsourced to the cloud provider may have a great impact on the cloud customer's ability to meet its duties and obligations to its own customers and employees. The customer of a cloud provider may hold liable for any injuries suffered by its own customers and employees [38].

2.2.2.2 Technical

Figure 2.13[38] shown below illustrates the technical risks.

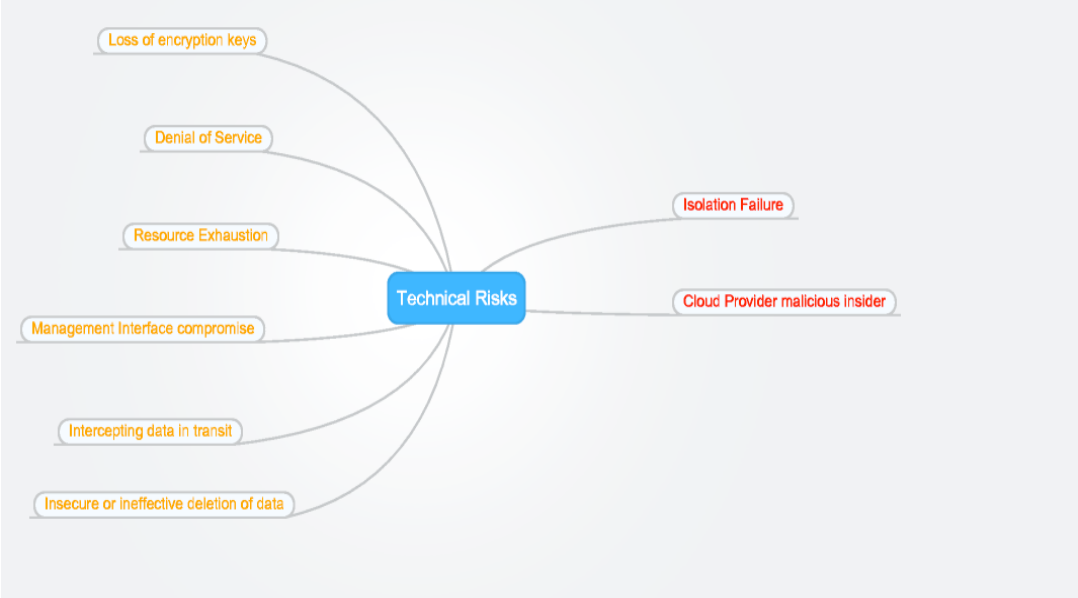

Figure 2-13[38]: Technical Risks

a) Management interface compromise

The customer management interfaces of public cloud providers are accessed via the Internet and this allows clients to access to larger sets of resources (than traditional hosting providers). Therefore, an increased risk is posed especially when combined with remote access and web browser vulnerabilities. Examples of these vulnerabilities include man-in-the middle, script attacks etc. In addition, back-end technology could allow unauthorized connections leading to data theft and account compromise. Moreover, management interface includes customer interfaces which control a number of virtual machines and the CP interfaces controlling the operation of the overall cloud system [38].

b) Denial of Service (Dos)

An attacker could launch a denial of service by using the public channel to use a customer's metered resources. The mitigations against DoS attacks would depend on the capabilities and configurations of the provider's cloud technology [38].

c) Loss of encryption keys

Poor management of keys may cause disclosure of secret keys (SSL, file encryption, customer private keys, etc) or passwords to malicious parties. This shall either lead to corruption or loss of those keys and potentially may result in unauthorized use for authentication and digital signatures [36].

d) Isolation Failure

Two of the main characteristics defining the cloud are multi-tenancy and shared resources. These two characteristics make computing capacity, storage, and network shared among multiple users. This leads to the creation of a class of risks that include the failure of mechanisms separating storage, memory, routing, and even reputation of different tenants of the shared infrastructure [38].

2.2.2.3 Legal

Figure 2.14[38] shown below illustrates the legal risks.

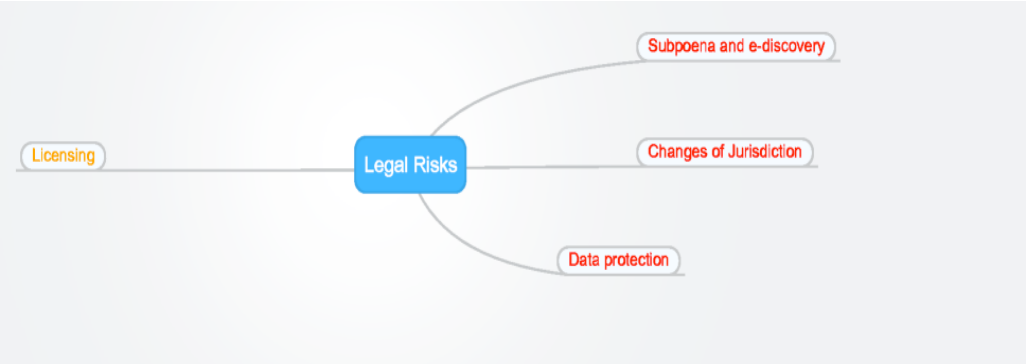

Figure 2-14[38]: Legal Risks

a) Risk from changes of Jurisdiction

In cloud, the user's data is stored in multiple jurisdictions, some of these jurisdictions are unsafe at all and this exposes the data to high risk. High-risk countries are those lacking the rule of law and having an unpredictable legal

framework and enforcement, autocratic police states, and states that do not respect international agreements, etc [38].

b) Data Protection

The cloud computing model poses a number of data protection threats to both cloud providers and customers. These threats include the following:

- The task of checking data processing that is carried out by the cloud provider is the main responsibility of the cloud customer (data controller). Although this task is difficult for the cloud client, he has also to make sure that the data handled by provider is handled in a lawful way. Failure to comply with data protection laws may lead to administrative, civil and also criminal sanctions, which vary from country to country, for the data controller.
- The cloud provider does not notify the cloud client of any data security breaches that may take place.
- The cloud customer may lose control of the data processed by the cloud provider. This issue is increased in the case of multiple transfers of data (e.g., between federated cloud providers) [38].

2.2.3 Top Threats to Cloud Computing

2.2.3.1 Abuse and Nefarious Use of Cloud Computing

Legal CSP can be abused for nefarious purposes, supporting criminal or illegal activities towards the cloud customers. As a result, the provider's services can be used to host malicious code or facilitate communication between remote parties. This abuse for CSP can also lead to provision of purposefully insecure services used for data capture. Moreover, the abused CSP may attack potential users with offers that seem to be true. For

example, the attack may promise the customer with unlimited resources or '30 day free trial' for certain service. During the registration process the customer will be asked to provide extra information than usual under the pretense of providing service personalization. Attackers can then use this extra information for nefarious purposes. Even if the CSP was not malicious, the disclosure of personal information to CSP can also be considered an abuse of service by the CSP himself. For example, the CSP can make use of the extra information collected by marketing them to third parties for data mining purposes.

This type of attacks can be prevented by using strong initial registration and validation process, powerful monitoring and coordination for credit card fraud, and comprehensive introspection of customer network traffic [39].

2.2.3.2 Insecure Interfaces and APIs

Application Programming Interfaces (APIs) and other interfaces are used by cloud customers to access their data in the cloud. Any errors or malfunctions in these interfaces software or the software used to run the cloud can lead to disclosure of user's data and rebut upon the data's integrity. An example of this attack is the flaw in apache server which allows an attacker to gain complete control over the web server. Moreover, exposure of data can take place when a software malfunction affects the access policies governing users data leading to user's data exposure to unauthorized entities.[40][41].

Threats can also exist as a result of poorly designed or implemented security measures. Regardless of any threat APIs are exposed to, APIs need to secure against accidental and malicious attempts to circumvent the APIs and their security measures. In addition, strong authentication and access controls should be implemented besides the encrypted transmission. Good understanding of the dependency chain associated with the

API and security model analysis of the APIs should be achieved so as to prevent any attacks to the APIs [39].

2.2.3.3 Malicious Insiders

The cloud customers trust the cloud providers by leaving their data under their control. Although the CSPs may be honest, their employees may not be. A malicious insider is an employee working for a certain CSP and abuses his position by collecting user's information/data either for nefarious purposes or for marketing this data to third parties. In other words, CSP employees will have access to consumer's data for legitimate purposes but they abuse this power for their own means [42]. Another form of the malicious insider problem is in PaaS based services. If developers were allowed to interact with users through the service provider platform, users may ignorantly allow these developers to access their data. For example, in Facebook platform once a user adds an application, the application will automatically have the ability to access the entire user's information, regardless of the applications function. Even if the application developers are not malicious this does not mean that the application cannot be hacked [43] [44].

2.2.3.4 Data Loss or Leakage

Although insecure APIs can lead to data loss or disclosure of information, there are other means in which customers can lose their information / data. The two most important means that lead to data loss are availability and data leakage.

a) Availability issues appear when the customer is unable to access his data. This lack of availability can be a result of data deletion, access privilege revocation or restricting physical access to the data itself. Attackers can also use flooding based attacks to attribute customer's data availability [45]. For example, Denial of Services attacks,

attempt to flood the service with requests in an attempt to overwhelm the service and halt all of the services intended operations.

Fault tolerance protocols are mainly used to resist the node failure within the cloud. Fault tolerance protocols can be defined as protocols that duplicate the data across different machines, and data centers, so as to ensure that if part of the cloud fails for whatever reason, customer's data in this part will still be available in other places within the cloud. However, poorly designed fault tolerance protocols can lead to availability issues. Furthermore, the presence of multiple copies of the same data can introduce confidentiality problems because the increased number of data instances increases the attacker's chance of accessing the data [39].

b) Another form of data leakage comes from the disclosure of information. This disclosed information, through hidden, is deduced from freely available information. Famous examples of this type of attacks include: unwelcome linkage and social graph merging. Usually when users interact with a service, they may leave a public trail which takes the form of status/update messages or new postings. Unwelcome linkage take place when new information identifying an individual is obtained through analysis of the individual's public trail i.e. links. This unwelcome linkage could be accidental or the result of the individual not covering their tracks. A social graph is a graph that describes the person's social information such as friends, groups and interests [46]**.** Social graph merging is similar to unwelcome linkage; however, the links formed occur through the aggregation of social graphs [39].

2.2.3.5 Account or Service Hijacking

During the communication between the customer and the CSP, malicious entities may

seek to affect the integrity and authenticity of this communication. There are several ways in which the authenticity and integrity of a customer's session can be impugned. Customer uses browser-based interfaces and authentication in order to establish a session with their service provider. A malicious entity can attempt to capture or hijack this session in order to steal the customers credentials, or access/influence the users data from within the browser [47]**.**Since most browsers operate using the same origin policy which states that client scripts are allowed to access resources if they share the same origin, attackers can make of this policy to access users resources.

Furthermore, the effects breaking session integrity between the customer and provider are two-fold, for one the attacker will be able to steal the identity of their victim, and secondly the fake data will confute the reputation of the victim .These man-in-the-middle attacks will have lasting repercussions such as violation of the services terms of use. Moreover, these man-in-the-middle attacks may affect the confidentiality of data if the attacker has the ability to read the data as well as modify it [48] [39].

2.2.3.6 Shared Technology Vulnerabilities

The vulnerabilities related to the construction of the cloud and the services themselves represent a more interesting form of confidentiality issues. In the following lines, we will explore two issues related to the cloud itself. These issues are virtualization issues and service aggregation.

a) Virtualization Issues

The virtualized architecture of the cloud gives IaaS service providers the ability to host several machine images on a single server. This architecture brings into the cloud more vulnerability. The attackers could make use of this architecture to map the internal

structure of the cloud so as to determine if two virtual machines were running on the same physical machine or not. Moreover, the attackers may be able to add a virtual machine to the cloud so that it was co-resident with another machine. Finally, once the attacker is able to co-resident his machine with another in the cloud , he would be able to launch several attacks that would allow him to learn information regarding CPU cache use, network traffic rates and keystroke timings[39][49].

b) Service Aggregation

Service aggregation can be defined as combination of the functionality of existing services so as to allow rapid service construction. Although service aggregation offer new functionalities than that of normal cloud, it has several interesting problems [49]. In service aggregation data is shared across multiple service providers. Each provider has his own privacy policy that is subject to change. Furthermore, service aggregation can occur in an ad-hoc and rapid manner implying less stringent controls to be applied to the protection of data, thus increasing the likelihood of a problem [39].

2.2.3.7 Unknown Risk Profile

Risk Management can be defined as a business process that users can use to identify and mitigate threats. It also allows customers to know their current position towards the security of their data. Auditing information such current security practices, and software version, code updates are used as a basis for determining this position. In order for the customers to adopt any service, they have to accept the Terms and Conditions (including privacy policy) of the service, together with any Service Level Agreements made. These Terms and Conditions together with the SLAs define the existing laws and regulations that customers as well as providers need to comply with. However, there is not any clear

information about security practices and legislation that the CSP follows to secure its customers. Therefore, the consumers are left with an unknown risk profile and they are unable to determine the risk to their data as they do not have sufficient information to do so [39].

2.3 Cloud Threat Models

2.3.1 Overview

The threat model is used to organize the system threats and vulnerabilities into general classes so as to be addressed by the storage protection techniques. In other words, thread modeling can be defined as a proactive systematic engineering approach that identifies all possible vulnerabilities and threats irrespective to their probability of occurrence. Therefore, it is important for engineers and system designers to aware of all the threats and vulnerabilities present in the storage system before they begin designing or implementing any storage protection solution. Because the type of threat determines the security counter measure that can be used , threats analysis cannot depend only on brainstorming or respond to threats that have recently occurred as this will leave large portion of the attack space unprotected. In this chapter, we will go through two different processes to creating a threat model for storage systems:

(1) A threat model process based on the CIAA principles of confidentiality, integrity, availability, and authentication and (2) a threat model process based on the Data Life cycle model.

The CIAA thread model is based on the four basic security requirements which are confidentiality, integrity, availability, and authentication. Each threat presented in this model is classified based on its relationship with each requirement not with the context of its implementation. While the Data Life cycle model considers the original of the threat and when the threat is likely to occur during the service operation [50].

2.3.2 Threat Overview

In this section, we shall spend some time understanding the origin and the nature of the threats.

2.3.2.1 Origin

The knowledge of threat origin can basically come from the attacks the system is exposed to. Attackers need only to find one security flaw to compromise the whole storage system. Attackers are generally classified into broad categories which include organized crime, espionage, terrorists, individual criminals, hacker cells, and insiders with privileged access, individual attackers, and nation-states – real attackers rarely fit neatly into one of these categories. However, in this paper we shall divide the origin of threats to data into three categories [50]:

1) Outsiders: These are entities that exist outside the system and attempt to damage and destroy the security infrastructure of the service. They also work on stealing user's account and password, disguising as a legitimate service so as to trap the users. In addition, they can stand in the data transmission process for starting middle man attack, stealing user data in transmission network, modifying the data, and even creating some invalid data.

2) Insiders: These are entities that exist inside the system. Examples of these types include current or past employees of the CSP and users of cloud services. These employees have great knowledge of the actual infrastructure including that of the security; therefore, they represent the most serious type of threats. Moreover, since the CSP have full control over resources in cloud, including user’s data, they can easily move or backup the user’s data modify user's profile information and gain the unencrypted information in memory when data is in use stage without data owner knowledge.

3) Natural: Both insiders and outsiders cause errors to infrastructure and these errors are intended errors; however, not all errors are intended. There are errors that take

place naturally due to software or hardware failures. Example of this nature error is the software update that Google offered to Google Docs. This software malfunction changed the sharing settings of several users' documents to include those with whom the affected users had share documents before [40].

Therefore, it is important to understand the attacker type so as to understand the resources and capabilities they have at their disposal.

2.3.2.2 Goals

Attackers are always goal driven towards a particular asset. These assets (system resources) can be either tangible (e.g. data) or abstract (data consistency). It is impossible to find a threat without finding its corresponding asset because assets are the threat goals. Hasan, Myagmar et al. [50] provides an incomplete list of what these assets maybe. However, we added more items to the list that are cloud specific asset.

An incomplete list of possibly targeted assets includes:

- Data blocks
- Buffer cache
- File handles
- Device drivers
- Communication channel
- Storage media
- Data management software
- Data availability
- Data secrecy
- Data integrity

- Data consistency
- Virtual image availability
- Virtual image secrecy
- Service availability

2.3.2.3Means

Attackers usually need to gain access to the assets in order to accomplish their goals. The access points could be configuration files, hardware ports, open sockets, RPC interfaces, and file system read/write. Hasan, Myagmar et al. [50] provides an incomplete list of access points that may be exploited by attackers:

• Access data from outside through network connection

• Access data from inside via trusted access or system compromise

• Physical access to SAN fabric

• Management interface from remote location to SAN fabric

• Compromised server accessing data and SAN fabric

However, it is important to note that internal attacker does not need to exploit any technical insecurity because he has direct access to the data or he can gain access through privileges escalation.

2.3.3 The Lifecycle of Data

Data life cycle can defined as the entire process from generation to destruction of data. The data life cycle is usually composed of seven stages [shown in figure 2.15[51] as follows:

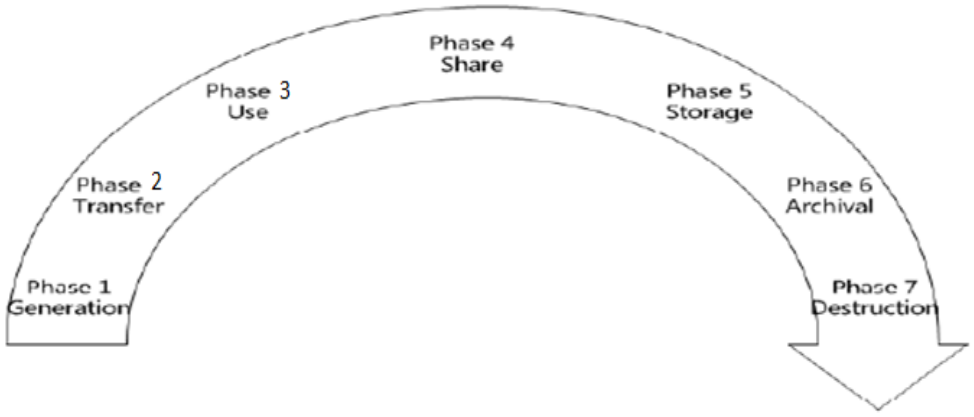

Figure 2-15[51]: Data Life Cycle

1) Data Generation

This is the initial stage of the data life cycle in which the data is created or modified / altered /updated by the user. After the creation process, the data is ready for uploading to the cloud for consumption [51].

2) Transfer

The second stage of data life cycle is the transfer and store stage. In this stage, the data is transmitted from the user machine to the cloud. This data that is transmitted across enterprise boundaries requires both confidentiality and integrity measures through the entire transfer process so as to prevent data from being tapped and tampered with by unauthorized users[51].

3) Use

In this stage, the data is usually viewed, processed or used in other sorts of activities except data modification. Data in this stage can be divided into two categories: static and dynamic data. Static data is usually stored on the cloud without any manipulation over it, while dynamic data is the data that operations are performed over. Due to the multi - tenant feature of cloud model, all users' data that are processed by the cloud is stored

together. Therefore, data should be encrypted so as to protect it from security problems. However, static data is feasible for encryption as it is not processed, while dynamic data are not feasible for encryption as data encryption will lead to problems of indexing and query [51].

4) Share

Data sharing is the process of making data accessible to others. In other words, the data owner can authorize other parties to have access his data. In cloud model, when a user give other party the right to access his data, this party can in turn share this data with others without the data owner's permission. Therefore, the data owner has to make sure that the party he is sharing his data with follows the protection measures and usage restriction [51].

5) Storage

The storage process is the process of committing data into the storage repository. In cloud model, the SPI model divides the data into two groups :(a) Data in IaaS environment; (b) Data in PaaS or SaaS environment related to cloud based applications. The data stored in cloud storage is similar to other traditional storage where it needs to be secured against security issues. Therefore, three aspects of information security, confidentiality, integrity and availability, should be taken into consideration when dealing with data in cloud. Data confidentiality can be solved easily by using encryption algorithms. While in data integrity two challenges should be considered: a) checking the integrity of data without having to download it then upload it; b) ensuring the integrity in cloud with traditional methodologies which may not be effective. Finally, data availability in the cloud is exposed to more threats than traditional external attacks. These

threats include: (1) The availability of cloud computing services; (2) the cloud provider continuity to operate in the future? (3) the ability of the cloud storage services to provide backup? [51]

6) Archival

Archiving means that the data leaves its active state and enters long term storage. In cloud environment, the CSP should regularly replicate data for archival purposes. This means that a copy of the data will be transferred to an external storage. Therefore, the storage media should always be under the control of the CSP because otherwise the data will be exposed to leakage risk. In addition, the CSP have to provide off-site archival of the data in order to assure the availability of data, because if the CSP did not do so, the data availability will be threatened. Moreover, the storage duration should be consistent with archival requirements so as to protect the data from availability or privacy threats [51].

7) Destruction

When the user no longer needs the data, he asks the provider to permanently delete this data. But the question here is whether the provider actually deleted the data completely from the storage or not, because the physical characteristics of the storage medium, the d may remain the deleted data in the storage which makes its restoration an easy task for attackers. Therefore, the user has to make sure that his deleted data no longer exist in the cloud after deletion [51].

2.3.4 Data Lifecycle Threat Models

The presence of data in the cloud exposes it to more risks than in traditional computer systems. Therefore, a strong threat model is needed to identity risks and vulnerabilities so

that appropriate solutions are found to address these vulnerabilities. Earlier in this chapter two threat models, which are the CIAA and data life cycle, were presented. The CIAA Model is used to model threats to data, however, it lacks context. While the Data Life cycle classifies the threats , first by placing them within the data life cycle and then using the CIAA model.

Figure 2.16 [50] shows storage attacks based on data life cycle.

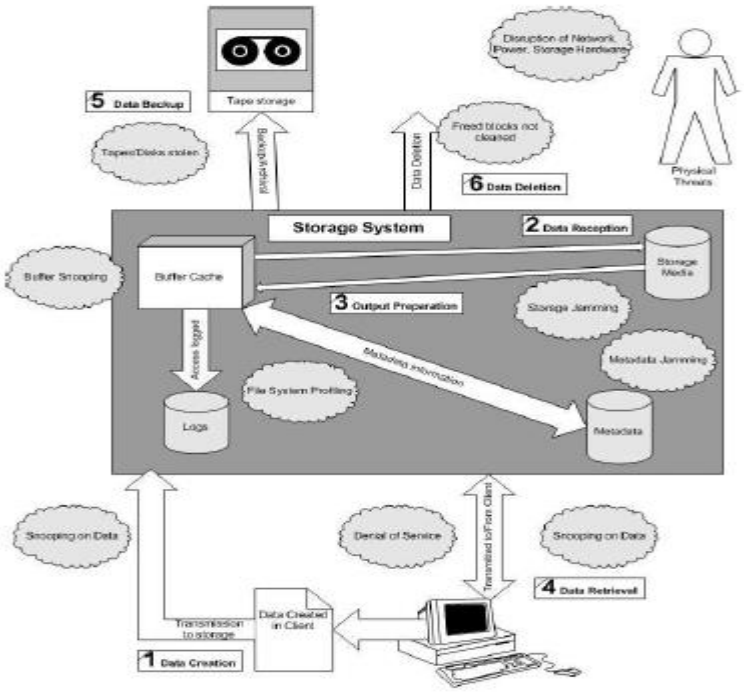

Figure 2-16[50]: storage attacks based on data life cycle

Next, we shall discuss the different stages of data life cycle model with associated threats:

1. Data Creation/ generation: In this stage, the data is created by the user. After the generation of the data, data could have security risks. For example, the attacker can tamper user's data and modify its access right leading to loss in his rights [14].

2. Data transmission: After the creation of data, the data is transferred from user's

machine to the cloud. Through the transmission process of the data, the data may expose to the following risks: a) attackers may sniff data on the communication channel, b) attackers may alter the data by performing a man-in-the-middle attack, c) the communication channel may by disturbed with DoS attacks and d) real or fake data may be created using a stolen identity [14].

3, 4) Data usage/ Data sharing: Data is either used by its owner or shared with other users. This data may be used by corrupted users to cause data leakage. Therefore, data confidentiality, integrity, and consistency should be maintained [14].

5) Data storage: In this stage, the data is stored in the cloud datacenters. This stored data may encounter many security threats. These threats may include unauthorized access and data tampering from malicious CSP and network intruders; the aging risk of backup medium; the risk of Information leakage during the backup; the mistake from legitimate users [14].

6) Data archival: the data in this stage is replicated to disk or tape for backup purposes; therefore, it will not be used temporally. However, this archived data will suffer from security problems. These security problems include the following: a) Backup media may be stolen, b)the backup software is less-protective; for example, a buffer-overflow vulnerability in backup software may allow the attacker to take control of system leading to the execution of malicious code and the launch of DoS attacks; c)backup availability can be denied by attacking backup timing synchronization, power supply, storage media, or network; d) the attackers devices may masquerade as trusted storage system component so as to receive a copy of replicated data; e) if the data is archived to online storage, it may be exposed to illegal access[14].

7) Data destruction: when the data is no longer used, it is deleted from the storage media. The problem appears when the data is not completely destroyed as this may lead to risks of illegal restore. The following attacks can be executed during this stage: a) An attacker may snoop on deleted storage blocks; b) meta data can be changed so as to subvert accurate evaluation for deletion and discarding; c) attackers may disguise under a real identity in order to delete data before its due date or to extend data beyond its due date [14].

2.4 Data Security Requirements

2.4.1 Overview

In this chapter, we shall explore the set of security requirements needed to have secure data in cloud. we address these requirements from two perspectives: user's prospective and cloud storage provider prospective because the data is shared between the user and the provider.

2.4.2 Confidentiality

Confidentiality is one of the most important aspects of data security. It can be defined as the assurance that sensitive information is not revealed to unauthorized users, processes or devices. Therefore, measures should be taken so as to protect users' confidential data from cloud service providers and external attackers, because if the confidentiality is violated either maliciously or accidentally, serious problems will take place. In order to prevent these confidential violations, sensitive /personal data should be encrypted. Encryption of sensitive or personal data should be used in all data forms e.g. when the data is in transit, when the data is at rest, and when the data is in manipulation. Moreover, the communication channel between the cloud provider and the customer as well as data centers should be encrypted .Remote administration of the cloud platform should only take place via a secure communication channel. Furthermore, if the user is not going to only store data in cloud but plans to process it, he must take in consideration that encryption cannot be used during processing of data(except for very specific computations). However, current research in encryption tried to solve this problem.

Although data confidentiality is primarily solved via encryption, there are other aspects that need to be considered such as user and resource privacy and deducible data [52] [53].

I. User and Resource Privacy

In the cloud computing environment, data confidentiality extends to how the data is being processed / used. The means that CSP used to store or process the data are bounded by the law and these laws must be followed. Example of this include: auditing records indicating access attempts and changes (and their results) to the data; properties of the data including size, access policies and origin [52] [53].

II. Deducible Data

Deducible data refers to the hidden data that can be deducted from existing ones. It is important to unable the attacker to use existing information or information related to confidential data (e.g. meta- data) to deduce any other information. In addition, such attacks should be made as difficult as possible so as to protect data [52] [53].

Confidentiality is an aspect that both the customers and CSP need to be aware of. Although the security of data is the responsibility of the cloud provider, in some cases (e.g. an IaaS storage service) the cloud customers have to encrypt the data by themselves before sending it to the cloud. Specifically, the confidentiality of the plain-text data itself should be the responsibility of the customer before it goes into the cloud. The CSP should guarantee the user and resource privacy, and deducible data because he is in a better position to provide such guarantee [52] [53].

2.4.3 Integrity and Consistency

In the cloud computing model, the mobility of data increases the threats that can affect the data integrity. Integrity can be defined as the protection of data from unauthorized access that can take place either maliciously or accidentally during processing, storage or transmission of data. As the data being transmitted to and from the

customer and the cloud service provider, and also internally within the cloud, data integrity should be guaranteed within the cloud. Ensuring data integrity can performed by several techniques which include:1)cryptographic authentication mechanisms(e.g. message authentication codes or signatures) that is used to detect alterations to personal data 2) creating hashes of stored data and comparing them with newer hashes of the same files. In addition, there should be a secure communication channel between the customer and the provider in order to ensure integrity of data during transit. It is also important to check the data accuracy during manipulation so as to prevent fraud. Moreover, interference with the integrity of IT systems in the cloud can be prevented or detected by means of intrusion detection / prevention systems (IPS / IDS).There should also be rules on using “lossy” compression techniques on files that are not text based[51][52].

Data consistency can be defined as a measure of accuracy and integrity of data. In other words, if there are several copies of the same data, these copies should be identical. Consistency problems appear in the cloud because of two main failures which are omission and commission failures. Omission failures take place when an entity fails to act upon input. Examples of omission failures include crash failures, failing to receive a request, or failing to send a response. While commission failures are these types of failures that take place when an entity responds to input with an output that is not what was expected. Examples of commission failures include processing a request incorrectly, corrupting local state, and/or sending an incorrect or inconsistent response to a request. Data stored in the cloud is replicated for many reasons such as availability, scalability or archival purposes. This replicated data may be exposed to inconsistent state, therefore, the consistency of the replicated data must be ensured [52] [53].

2.4.4 Availability

Another issue that should be addressed when considering data security requirements is data availability. Data Availability can be defined as availability of data for access whenever it is requested. In other words, availability means timely and reliable access to personal data is always ensured.

The availability requirements of the data vary depending on how critical the data is. If the data is highly critical, then data should be made redundant and it should also be backed-up regularly so that the data is available even under hard circumstances. But if the data is not that many important, regular backups are not important and one or two copies of the data are sufficient. It is also important to note that, it is the responsibility of the CSP to guarantee the availability of data. If the data was unavailable for whatever reason, the users will not be able to access their data and become unsatisfied and this will lead to loss for both the user and the provider. The most severe threat that availability exposed to in cloud is accidental loss of network connectivity between the client and the provider or of server performance caused by malicious actions such as (Distributed) Denial of Service attacks. While other threats to availability could include accidental hardware failures both on the network and in the cloud processing and data storage systems, power failures and other infrastructure problems. To minimize or eliminate these types of attacks, data controllers have to check whether or not CSP has adopted measures to cope with the risk of disruptions, such as redundant storage ,backup internet network links and effective data backup mechanisms[52][53].

2.4.5 Access and authentication

The cloud is a public place where services are exposed over HTTP, public medium. Access to these services need to be managed and access should be kept to only authorized

users. Moreover, as data is stored and managed remotely, users should trust the service offered by CSPs and security offered by providers over their data. On the other hand, CSP must ensure that anyone trying to access the data is not only who he says he is (authentication) but also he has the right to do so (authorization). Doing this is not an easy task because it require from the CSP ,who is interacting with multiple users from multiple companies (domains) , to offer different levels of management and access policies ;all these operations are done remotely[52][53].

Remote access should include the following operations:

a) Authentication

In this process, the CSP should make sure that anyone trying to access any service is authenticated to do so. Unauthenticated users should not be able to access the data under any condition. In other words, the identity of the users must be assured and this implies some form of identity management.

b) Authorization

Once the user is able to authenticate himself to the CSP, he is able to gain access to CSP services and to his data in the cloud. Therefore, CSP should regulate and control the access to its services and data so as to prevent unauthenticated users from accessing services and data they are not authorized to access. For example, two users subscribe to the same CSP but work for two different companies should not be able to access each other's remote data held by the CSP unless the access has been explicitly allowed.

c) Location

Users usually do not access their data and CSP services from one fixed location. They

can access it from home or work. Therefore, user authentication should always be performed and should not be linked to the device from which the he accesses the service.

d) Revocation

An important requirement is that of revocation. The revocation of access to individual data and to the service itself must be permissible.

2.4.6 Data retention

Data retention is used to define policies related to persistent data and records management. These polices should meet legal and business data archival requirements. In addition, each data retention policy should weighs legal and privacy concerns against economic concerns so as to determine the retention time, data formats, archival rules and the permissible means of storage, access, and encryption. In cloud environment, the user generally should aware of the following when concerning data retention: 1) how long are the data retained by CSP and is this retention period enough to satisfy his legal obligations? 2) when the CSP destroys the data, what process is used, is this process robust enough to actually destroy the data, and is this process secures enough? while in case of cloud storage, the user should ensure that 1) data retention rules are clearly define2) stored data and the storage duration are well defined3) the data retention policy has to form the basis of the storage plan because without a coherent plan and retention policy, the user will just be busting money and this will only result in underutilized technology and capability [52].

2.4.7 Audit-ability

Audit-ability can be defined as the task of auditing a system or an environment In cloud model, it should be identifiable who created, viewed, and modified the data

because this will make it possible to track back everything that has been done over the file throughout its entire lifecycle. This collected information about the file is required for auditing and control. A security audit can be defined as a systematic evaluation of CSP security by measuring how well the CSP conforms to the set of established criteria. These audits usually assess the security of the system's physical configuration and environment, information handling processes, software, and user practices. On the other hand, data audit-ability is required for compliance with many regulations even if there were no regulation enforcing data audit-ability. it is also important for Cloud Service Consumer to ensure that the integrity and confidentiality of their data is respected by the Cloud Service Provider. Poor audit-ability means that the system has poorly-maintained records and systems that enable efficient auditing of processes within the cloud, therefore, good audit-ability should be maintained so as to get better security within the cloud. Audit-ability is also an enabler of accountability where it allows any action to audit against a pre-determined policy to determine if the action was compliant or not [52].

2.4.8 Portability

Presently, most CSPs do not follow any standards for data formats and service interfaces facilitating interoperability and portability between each other. Therefore, if a client decided to leave his current CSP and migrate to another CSP, this may be impossible or at least there may be difficulties in the transfer process due to the lack of interoperability. The same will also be true for services developed by the client on a platform offered by the original cloud provider (PaaS). As a result, the cloud client should check that the provider he is using guarantees the portability of data and services prior to ordering him [53].

2.4.9 Accountability

Accountability can be viewed as the ability to demonstrate what a user did at a certain point in time in the past and how he did. Accountability in the field of data protection can be defined as the ability of parties to demonstrate that they took appropriate steps to ensure that data protection principles have been implemented. In cloud computing, accountability is used to investigate personal data breaches, where cloud components from cloud clients, providers and sub-processor bear a degree of operational responsibility in order to provide reliable monitoring and comprehensive logging mechanisms. Furthermore, the CSP should provide documentary evidence of effective measures that deliver data protection required outcomes. Examples of these measures include: independent certification procedures, respond to access requests, allocation of resources, and procedures to ensure the identification of all data processing operation [53].

2.4.10 Erasure of data

Data should be deleted when it is no longer needed or after fixed time interval. After the data deletion, cloud providers have to attest that the data that has been destroyed will never be reconstructed. However, many cloud providers will not be able to do so(will not be able to attest the deletion) because of the way cloud data is rapidly replicated and relocated on many disk drives, servers, and data centers.

The principle of erasure of data applies to personal data regardless of whether they are stored on hard drives or on other storage media (e.g., backup tapes). Since personal data may be stored redundantly on different servers at different locations, it must be ensured that each instant of the data is deleted irretrievably. Moreover, cloud client should be aware of the fact the log data facilitating audit-ability may be considered as the

personal data of the person who initiated the processing operation. In order to ensure secure erasure of personal data, we need that either the storage media to be destroyed or demagnetized or the stored personal data is deleted effectively through overwriting. The overwriting of data requires special software tools to be used. These software tools should overwrite data multiple times in accordance with a recognized specification. The cloud client should also make sure that the cloud provider ensures secure erasure in their SLAs. The same holds true for contracts between cloud providers and subcontractors[53].

2.4.11 Transparency

In cloud computing, transparency means that the cloud client should be aware of all subcontractors contributing to the provision of his cloud services as well as of the locations of all datacenters that his personal data may be processed at. In addition, if the provision process of the service requires the installation of software on the cloud client's machine, the cloud provider should inform the client about this circumstance and explain its effect on the his data protection and data security . Inversely, the cloud client should raise any issue that is not addressed sufficiently by the cloud provider.

Transparency is considered to be an important key for a fair and legitimate processing of personal data. Moreover, transparency should be ensured in the relationship(s) between cloud provider, cloud client and subcontractors (if any). The cloud client is able to lawfully assess the processing of his personal data in the cloud only if the provider informs him about all relevant issues [53].

2.4.12 Isolation

In cloud computing infrastructures, resources such as memory, networks, and storage are shared among many users. The sharing of these resources within one cloud creates

risks of data disclosure and processing for illegitimate purposes. Therefore, isolation between user's data is required so as to protect them from each other. Isolation means that guarantees should be taken so that user's data is not used beyond its initial purpose and its confidentiality and integrity is maintained. In order to archive isolation, two procedures shall be followed:1)adequate governance of the rights and roles for accessing personal data should be review on regular basis so that implementation of roles with excessive privileges is avoided (e.g., no one even the no administrator is allowed to access the entire cloud). More generally, users including the administrators must be able to access information related only to their legitimate purposes (least privilege principle). 2)isolation should not be applied on users only but also on technical measures such as proper management of shared resources and the hardening of hypervisors if virtual machines are used to share physical resources between different cloud customers [53].

2.5 Principles of Cloud Storage Services

This section introduces the typical features of cloud storage services. A particular service must offer at least one of these features, and may offer multiple features at the same time.

2.5.1 Features

In this section, we shall explore the main feature of the clouted storage services. These features are: copy, backup, synchronization, and file sharing. Any storage service must include at least one of these features, and may include multiple features at the same time. Figure 2.17[54] show the main features of the cloud storage service.

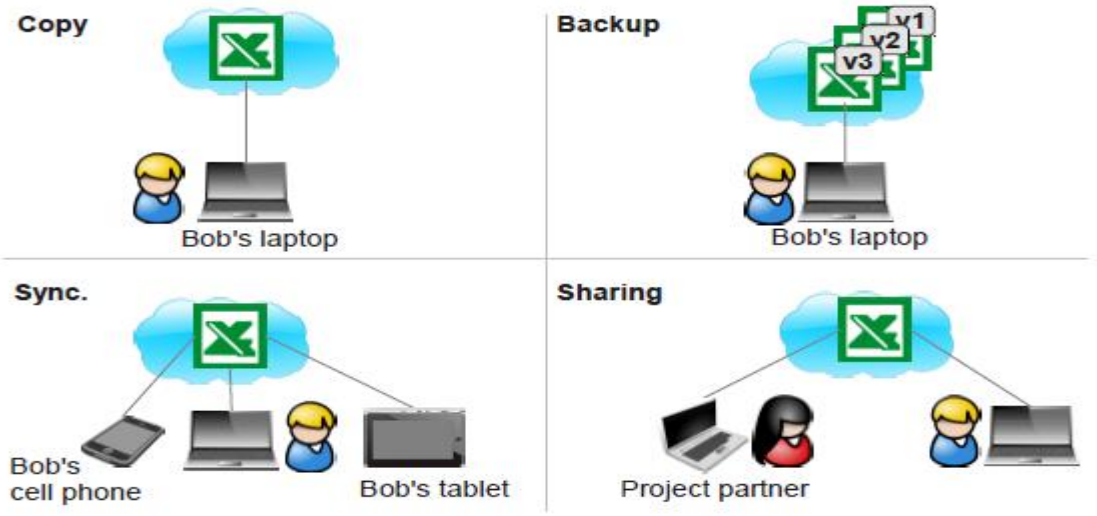

Figure 2-17[54]: Features of Cloud Storage Services

2.5.1.1 Copy

The copy feature creates an image of the user's local data into cloud. The user uses this copy to make sure that his data is always available even if there is a local hardware failure (e.g. a hard disk crash).Moreover, this copy will help the user to access his data from any place (e.g. through web browser) even if his local hardware is not available.

This gives the user two ways to store his data in the cloud. In the first way, the user manually uploads his files and folders into the cloud through the web browser. while in the second way, the user make use of a client software, that is locally installed in user's machine ,to automatically upload his files and folders from a given folder belonging to the client to the cloud storage. It is also important to note that the copy feature is different from backup feature. Where in backup feature the data is stored in predefined periods of time, while in the copy feature the data is stored continuously[55][56].

2.5.1.2 Backup

The backup feature allows users to restore any previously stored version of a file or a folder over a long period of time (usually years).To create backups, the cloud storage services usually make use of an automatic process. In this process, data is copied, transmitted, and stored periodically in the cloud so as to be recovered in case of original data loss. To perform this process, the cloud storage service providers have to offer client software that is installed locally in the user's machine. This software enables the users to select the data to be backed up, to configure the retention period, and schedule for backups. In addition, this software can either run continuously in background or is configured to perform the backup on regular basis so as to backup the newly created or changed files. Moreover, one more task for the client software is to check which data needs to be backed up. Finally, it could enable users to monitor the backup process since the previous backup [55] [56].

2.5.1.3 Synchronization

Synchronization means having consistency among data stored in different sources. For example, a user can own a set of devices, e.g. a pc, laptop, tablets and a Smartphone,

and wants to have the same data available on all devices and whenever data is changed in one device the others are modified with these changes. Therefore, the cloud storage service providers have to provide users with client software that is able to detect any changes in any file in any device and reflect these changes to other devices. This client software Can do this by offering the user a number of choices: merge the files, overwrite one version, or keep both versions by applying a renaming scheme [55] [56].

2.5.1.4 Sharing

Data sharing is the process of sharing data files or folders with others. Cloud storage service providers offer users different forms of sharing. Users can share data with other subscribers of the same service, with a closed group of people from other services, or with everybody. For example, users can collaborate with colleagues, project partners, or friends. It is also important to note that any shared file / folder has a set of fixed or configured access rights as read, write or delete[55][56].

2.5.2 Interfaces

This section shall illustrate the different interfaces that the user can use to access the data stored at the cloud storage provider.

2.5.2.1 Proprietary Software Clients

The most comfortable interface offered by the cloud storage service providers is the proprietary software clients. Each provider has his own proprietary client and users make use of this client software by just installing it on their machines. This proprietary client offers the users a variety of services that they can use .These services include: selection of data to be transmitted to the cloud, management of service and configuration of features like synchronization or sharing [54] [57].

2.5.2.2 Browser Interface

A web browser interface is a method used to access user's data. Accessing the date from any place and from any device that do not have a software client installed is the main advantage of this method. A browser interface is usually preferred by organizations that do not want to spend time and money in managing software for their employees. Moreover, it is also preferred by end users who want to share their data whenever they desire [54].

2.5.2.3 Application Programming Interface

Most cloud storage providers provide their users with access to application programming interface (API).Developers can use these APIs to integrate access to cloud storage service into their applications. e.g., to provide games for a mobile device game across multiple devices and platforms. In order for cloud storage providers to grant customers with access to APIs, they need to expose a web service or web application that can be accessed using a standardized communication protocol[54][58].

2.5.3 Optimization

In this section, we shall explore some optimization techniques that are provided by some cloud storage services so as to save bandwidth. These optimization techniques are: de- duplication, delta encoding, and compression.

2.5.3.1 De-duplication

The term De-duplication (also Data De-duplication) describes a popular technique that allows cloud storage providers to significantly decrease the amount of needed storage space. The principle of de-duplication is as follows: only a single copy of each piece of data is stored. If a user wants to store data that the cloud storage provider already has

stored in the past, the storage provider simply creates a link to that data instead of storing another copy. There are some variations of how de-duplication may be realized:

In other words, De- duplication is a technique that is used frequently by cloud storage providers. This technique works by only keeping a single copy of each piece of data stored and if a user wants to store an already stored data, the provider only creates a link to this data instead of storing another copy. By using this technique, cloud storage providers can significantly decrease the amount of storage space needed to store user's data. De- duplication can take many forms as we shall illustrate in the following lines [59]:

(1) File level de-duplication vs. block level de-duplication

File level de-duplication means that de- duplication is performed on file basis where only a single copy of each file will be stored. While Block level de-duplication means that de- duplication is performed on block basis, where each file will be divided into blocks and only a single copy of each block will be stored. Identical files or blocks are detected by comparing the hash value with a list of known files or blocks [60].

(2) Server-side de-duplication vs. client-side de-duplication

Server-side de-duplication is a file level de- duplication, in which de-duplication is performed on server side. For each file transmitted from the user, the provider checks if he has to store the file or it is already present and only a link needs to be created. By using this method, the user cannot detect if de - duplicated is performed or not. While in client -side de- duplication, the client software only send a hash value of the file not the file itself to the provider. Only if the transmitted hash value is not present at the provider, the file is sent to be stored [60].

(3) Single user de-duplication vs. cross user de-duplication

Single user de-duplication means that de-duplication is performed for each user separately. For example, if user A wants to store a file that he stored in the past or in different folder, the cloud provider only creates a link to that file. While in case of cross user de-duplication, de - duplication is done across all users. For example, if user a want to store a file that user B already stored, the cloud provider only creates a link to that file instead of storing another copy of the same file [60].

2.5.3.2 Delta Encoding

Delta encoding is a technique used for minimizing the data transfer, thus saving the bandwidth. It works by only uploading the differences to uploaded file from the last change instead of transmitting the whole file with the new modifications. Suppose that a user modifies a certain file and wants to store it. In this case, instead of uploading the new modified file, it would be sufficient if we store only the modifications (Only store parts that were modified).It is important to note that, delta encoding make no sense with encrypted data because an encrypted file with modification differs completely from the encrypted file without modification[54].

2.5.3.3 Compression

Compression is a technique used to save bandwidth. It works by compressing data on client side. The main drawback of this technique is that it consumes computing power of the user, and this may cause troubles to users because the transmission of data to the cloud is a continuous process [59].

2.6 Security Requirements in Cloud Storage Services

In this section, we shall explore the minimal set of security requirements needed to have secure cloud storage services. These security requirements include interaction with the web application via web browser in login and registration processes, the transmission of data through transport layer, actual data storage and basic as well as special features of cloud storage client applications as file sharing, synchronization and de-duplication. Last but not least, it is important to note that this proposal focuses on some security requirements of cloud storage services not all; however, we might explore the other security requirements in future work.

2.6.1 Registration and Login

Before the cloud customers are able to make use of cloud storage services such as synchronization, file sharing and backup, they have to register for an account in any storage service. In the beginning of the registration process, the customer provides the service provider with his email, username and password to tie him to an account. Later on, during the registration process, the service provider and the cloud customer agree upon credentials that shall manage their relationship. These credentials shall later be used in the login process and in using the services. Furthermore, this registration process if not secured properly, it shall suffer from the following security problems: 1)If an attacker is able to eavesdrop on the communication channel between the provider and the customer, he might obtain a version of the credentials, and later he can compromise the account and gain access to all customer's uploaded data. 2) If an attacker is able to manipulate the messages exchanged between the provider and the customer, he might act as a proxy and deceive both of them. In order to protect the customers from these and other attacks, the

following procedures should take place: 1) all communications between the service provider and the customer must be secured in terms of confidentiality, integrity and authenticity. In order to achieve these goals, we have to use the Transport Layer Security (TLS) protocol [61]. Because the service providers need to authenticate themselves against the client machine by presenting a certificate, the customer can examine this certificate to make sure that they are communicating with their service providers.2) In case of a security breach, the best way to minimize potential data theft is to limit data collected to the minimum needed to operate the service.3) The service provider can optionally bring a third party payment service that can handle the entire process.

After the customer completes the registration process, he is able to login to the services offered by the service provider. These login systems are publicly accessible, therefore, they are exposed to attacks as brute - force attack or directory attack on the credentials .In order to guard themselves against these kinds of attacks, service providers should enforce complex passwords(ideally, these passwords contain 12 characters which include letters, numbers and special characters) that are difficult to be hacked. However, these complex passwords cannot guarantee that attackers will not able to guess them in the given time. To end this problem, service providers should implement additional measures to make these attacks infeasible. These measures could be time penalties or a temporary account lock down after a certain number of incorrect login attempts within a time frame[54][10].

In addition, the service providers have to provide its customers with strong authentication method in order to ensure that only authorized customers have access to its services. These authentication methods do not have to rely only on the knowledge of

credentials but rather demands possession of a token as smartcard or mobile phone. Example of this authentication schemes is two-factor authentication which combines something that is known to the customer, like username and password with something he owns, like a mobile phone or smartcard. Overall, these schemes can significantly improve the security level of the login process [54] [62].

Moreover, let's assume that the service provider builds a strong authentication mechanism with a sufficiently high security level; it also needs to implement additional measures to protect standard processes during account management. The e-mail verification during registration process is an example of these additional measures. The e-mail used by the customer during the registration should be verified by sending an activation link to the customer so as to complete the process. This verification step prevents any possible incrimination where an attacker registers using an email address which does not belong to him. Because if the system was implemented in such a way that credentials are directly associated with any newly created account without any verifications, the attack could abuse the password-reset process to gain a customer access to the service. This type of attack is called a denial of service attack. To prevent this type of attack, the provider has to send the customer an email containing a link leading to a password-reset form or temporary credentials that have to be changed directly after the first login. Furthermore, the provider should not provide feedbacks in the form of careless error messages through its web applications to customers because attackers can make use of these messages to gather information. For example, if the registration or login process informs the customer that the entered e-mail address or username already exists, the attack can use this piece of information to collect valid e-mail addresses and usernames

and sell them to spammers. Worse, the attack can use the valid user names and guess the associated password to hijack the customer's accounts. To prevent this type attacks, the provider should not reveal more information to customer than necessary [63].

2.6.2 Transport Security

The cloud storage providers provide its customer with client software that resides on their machines. This client software assists the customers in setting up synchronization or backup schemes on their local machines. Moreover, they handle the actual transmission of all data with remote storage servers which make them more exposure to attacks in case of insecure communication channel between them and the client. In these attacks, the attackers may be able to steal credentials, learn the content of the data or even manipulate it. As a result, the server must authenticate itself to the client software and make sure that all its communication with the client is encrypted and integrity is satisfied. In addition, the server must ensure that appropriate, up to data cryptographic functions are used. Therefore, standard protocol TLS is the appropriate solution for transport security [54] [61].

2.6.3 Encryption

One of the main reasons that make both individuals and companies use cloud storage provider is to always have a backup of their valuable data. This backup will always make the data available at any time so that the customers can easily access it in whatever time and place they want. Before the popularity of cloud storage, both individuals and companies had their own backup strategies so as to protect their data against any loss or damage. These strategies always rely on additional physical devices which are usually present at the same location as the original data. This mean that the data owner has full

control over his original and stored data, therefore, the protection of the data is not an important issue. However, nowadays with the widespread of cloud storage providers, customer's data in the cloud is not safe, since it is stored on public servers that are visible to data storage over on the internet. This makes the data subject to external as well as internal attacks. The external attacks come from outside the cloud while internal attacks come from the cloud storage provider itself. Therefore, the data itself should be protected in such a way that even if an attack successfully take place, the content of the stored data remain unchanged (secure). To this end, all data stored in the cloud storage needs to be stored in an encrypted form and there are a lot of secure encryption schemes that are freely available for use. Most cloud storage providers nowadays encrypt all data stored on their servers with a company key which is only known to them. This scheme of protection may protect customer's data from external attack, but does not protect it from internal attacks. Therefore, all customers' data need to be encrypted locally with a key unknown to the provider before it is transmitted to the cloud. The customer can do this by either encrypting the data by himself or use standalone software. However, this standalone software has a number of drawbacks: the software has to be installed, administrated and operated on the customer's machine in addition to the client software of the cloud storage provider. It is important to know that the key used to encrypt the data needs to be distributed across all devices that have access to the data; because if the key is lost, the data can never be decrypted again. Another method that can be used by customer is to sign their data. This method enhances security because it enables to user to verify his data [64] [65].

2.6.4 File Sharing

File sharing is one of the features that distinguished advanced cloud storage from basic cloud storage. Cloud storage providers usually offer file sharing in three different flavors: 1) Sharing files with other subscribers of the same service. 2) Sharing files with a closed group who are not subscribers of the same service 3) Sharing files with everybody.

Sharing files with a certain group of people, as in case one and two, creates a group of closed users and the data owner (sharing user) plays the role of an administrator of this group. As an administrator, the data owner has control over the data and has the right to assign privileges to others. This group of users including the owner requires from the provider the following security requirements:

1) The shared files should be accessible only to privileged used.

2) It should be possible to regain sharing for any file.

3) The provider should grant the user the right to access the list of their shared files either through their client applications or through the web interface of the provider.

4) The user (sharing user) should have the right to grant, edit or remove individual access rights from other users.

5) If the sharing user use client side encryption to their files, this sharing should not weaken the security level. In other words, the cloud storage service provider should not be able to read the content of the shared files.

6) If the sharing user use client side encryption to their files and there is a user disinfected from the sharing group, the encryption key used should be new and the old key should be discarded.

Usually sharing files with non-subscribers of the same service, as in case two, is accomplished by providing users with URLs to the shared files. These URLs usually

contain important information about users and the files and knowing these URLs means having the right to access the shared files. Therefore, additional security measures should be taken so as to prevent any attack. These security measures shall include the following: 1) The URL should be inscrutable. This means that the URL should not contain any information about the user or the file itself; because if the URL contains any of this information, the attacker can easily gain information about the user. Moreover, if the URL is free from any credentials, it should contain a randomly generated unique identifier. Because if the identifier size is too small or it is just an increment from published documents, the attacker can easily guess it by iterating over all possible links and thereby gain access to all published files.2) The cloud storage service provider should exclude the shared files that are hosted on the web server from being indexed by search engines 3) The cloud storage service provider should allow the user to choose the option and credentials they want to secure their shared files. This will help in preventing unauthorized from having access to shared files.

Sharing of files with everyone requires from the cloud storage provider to hide any information related to user, e.g. username, in order to maintain the user security [66] [67] [68] [69].

2.6.5 De-duplication

Data de-duplication is a compression technique that is used to eliminate duplicated copies of repeated data. This means that only one unique instance of the data is retained in the storage media while the redundant data is replaced with a pointer to the unique data copy. This data de-duplication technique is employed in most cloud storage providers so as to save large amounts of storage space, thereby reducing costs. There are different

types of de- duplication techniques that can be used by cloud providers. All these techniques were described in section **2.5.3.1**. However, the two main de-duplication strategies are file-level and block-level duplication. Any of these strategies can be used with one of the two basic approaches of de- duplication: client-based de-duplication or cross-user de-duplication. In client - based de-duplication, the de duplication is performed at the client side and this saves the bandwidth. The result of applying this approach is that the client can observe whether a certain file or block is de-duplicated. While in cross-user de-duplication is performed between the data of different users. This results in saving both storage and bandwidth. Although these two approaches save bandwidth and storage, they are the ones that suffer mostly from privacy and security issues. The following are examples of attacks that they can be exposed to:

1) An attacker who has an account at the cloud storage provider can easily know about files stored in cloud storage by using de-duplication. He can do this by transmitting a file to the storage provider and observe what happens to this file. If the file is uploaded by the client software, he knows that the file already exists at the storage provider. That is, he knows that the file exist, but he does not know the owner of the file.

2) If the attacker is able to get hash value of the file, he can easily know the content of the file. However, with the use of a well known hashing algorithm with a sufficiently large hash size, the probability that an attacker can guess valid hashes for (random or specific) files is negligible. Therefore, these types of attacks are hard to take place.

3) The attacker can also make use of the information described above in case 1 to deduce information about specific user of the cloud storage provider .Assume that an attacker knows that a specific user stores his files at the cloud storage provider and these files take

yes/ no format. The attacker can easily create different versions of these files and in each he inserts the user's name and the answers to the yes / no questions. After that, he consecutively uploads these files to the cloud storage provider and observes what happens. If one of these files is not uploaded, the attacker knows that user already stored this file, and thus the attacker knows the content of this file. Therefore, it is mandatory for the provider to protect its users from these attacks. One of the solutions that can be used regarding the de- duplication problem is based on the introduction of a random threshold. This threshold is assigned by the provider to every stored file. De-duplication will only take place if the number of uploads of a certain file exceeds this file's specific threshold. This means that the attacker who wants to know if a specific file has been uploaded by another user, will has to repeatedly upload the file until the de-duplication is performed. But at this point he cannot know exactly if the de-duplication is performed because the file is de-duplicated or because he has reached the threshold.

Ideally, in order for the provider to offer a secure service, he has to use client-side de-duplication within a single account or, when using cross-account de-duplication; Because in these two cases the provider would always upload any files added by the user even if they are already on the server and this will disable any feedback that the attackers can make use of. In addition, currently there are no known privacy issues related to server- side de-duplication [70] [71] [72] [73] [74].

2.6.6 Synchronization

Another important feature that distinguishes advanced cloud storage services from basic ones is synchronization. Synchronization is the process of making two or more

data storage devices or programs (in the same or different computers) having exactly the same information at a given time. Nowadays with the rapid development of ubiquitous computing, a typical user has multiple devices to access his data from depending on his current location. These devices might include the user's personal computer at home, the user's computer at work and his Smartphone. The user wants to be able to access the most recent version of his data from any of these devices depending on his current location. Therefore, multiple different devices shall be associated with a single user account. Each of these devices added to the user's cloud storage account have a different location. As a result, the way a new location is added to cloud storage account should be taken in consideration when considering the security of that account. Moreover, the security of backup, file sharing and storage of multiple independent devices associated with one account should be considered.

During the installation of any new device to the user account, the user should provide his credentials, the credentials he created during the registration process, to add the new device to his account. After the first login from the newly added device, the credentials are stored locally in that device. Later on, the user can directly use the cloud storage applications without having to enter his credentials in each login. This direct use of cloud storage applications creates a trade-off between usability and security. Because this usability may allow an attacker to steal the user credentials (e.g. the attacker may inspect the user's username and password from an unsecured channel) which he can later uses to attach his device to that user account. Worst, if this attack is not noticed, the attacker, in addition to, accessing the data will be able to notice any changes or modifications performed in the user account. A famous cloud storage services that is exposed to this

type of attacks is Dropbox. In Dropbox, the data configurations are stored in a file called "config.db". This file contains information about the user's email, dropbox_path and the host_id. If an attacker is able to copy this file from the user's machine, he can easily gain access to all users' data. Worst, the dropbox did not notify the user when a new device is added by the attacker [9]. To guard the user account against these attacks, any newly added device should be activated by the user. Even if the user's credentials are secure and are never compromised, there is still a chance for third parties to gain access to user's data. For example, when a user's Smartphone that have access to file stored at cloud provider is lost, anyone who finds the phone will have access to all user's files through the cloud storage application on that phone. Therefore, the user should have the right to remove certain devices from his account. This can be done easily by providing the user with a list of devices currently attached to his account [9].

2.6.7 Server Location

The cloud storage service provider has to indicate exactly where its servers are located. In other words, which country will host the user's date? Ideally, the cloud storage provider has to offer different storage locations from which the user can choose [54].

3 Approaches for Securing Data in cloud Storage

This thesis focuses on security problems in cloud computing. Specifically, it addresses outsourced data security in cloud storage services, secure and efficient cloud-based content delivery service, and secure cryptographic key usage in cloud-based data computing service. In this chapter, we review the previous state of the art research related to data security in cloud storage services. First of all, we shall explain the current work done in the service level agreements between cloud customers and cloud service provider for ensuring data confidentiality, integrity, and availability (CIA) in cloud storage. Second, we review modern researches on security problems in the area of cloud computing with two categorize. The first category is concerned with securing data in cloud storage service in terms of data confidentiality and integrity. While the second category focuses on efficient data access control in the cloud storage

3.1 Service Level Agreements (SLAs)

To mitigate the security risks associated with data in the cloud, there should be a clear security mechanism for securing data in the cloud. This mechanism should be formalized in a form of a contract so as to force both parties to follow. A Service Level Agreement (SLA) is a common way to specify the conditions under which a service is to be delivered. Although several SLAs are offered by different cloud service providers to regulate the relationship between the customer and the provider, all these SLA s are usually limited to availability levels and credits/penalties. The absence of effective security measures in SLAs has been a major hurdle for the adoption of cloud services, especially for enterprises and cautious consumers. Recently, researchers have tried to solve this problem by offering a number of secure SLAs

In [75], the author argues that the current research on SLA and QoS metrics has given more attention for areas as e-commerce and web services. However, this work produced good SLAs metrics that satisfy the required purpose, the SLAs metrics in these technologies are not suitable for cloud computing as the nature and type of resources being provided and delivered is different. Therefore, new SLA metrics are still required

to provide flexible the main requirement of the cloud which is security. This paper presents the requirements and the main points that should be followed at each stage of the SLA design to ensure that it addresses all aspects required by the cloud. It specifies the parameters (from data security, availability, integrity.....etc.) needed to ensure that a service provider delivers the agreed terms of services to the cloud consumer so as to maintain the trust and reliability between each of the parties involved in the negotiation process. SLA metrics for Storage as a service is an example of SLA metrics offered in this paper. In this metric the author tries to address all the basic requirements for data storage service metric.

In [75], the author states that the cloud vendor has to provide some assurance in their service level agreements (SLA) about the security issues. Since guaranteeing the security of data in the cloud is a difficult task because the cloud provides different services. Each of these services has its own security issues. Therefore, the SLA has to describe different levels of security and their complexity based on the services in order to provide the customer with a good understanding of how these security policies are implemented. In other words, there should be a standard SLA irrespective to a specific provider. Although there is a huge effort done in the area of providing standard SLAs that satisfy cloud customers and providers needs, there is no guarantee if the services do not met the agreement.

In [77], the author presents a framework to ensure data security in cloud storage system through SLAs. In addition, he discusses a number of technologies that can be used to provide safe storage for data in the cloud. These technologies are mainly divided into three categories: storage security, transfer security and authorize. In secure storage, the

main requirement is ensuring data security during data crash, stolen or disaster, such as fire and storm. While in secure transfer, Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) should be used to ensure that data transferred in secured communication channel over networks such as the Internet. Finally, authority is important means to assure unauthorized access. For the storage provider, it needs to do user authority and access control.

In [78], the author provides a way to manage the security of SLA through its life cycle. He makes use of a framework for security mechanisms as input to contract requirements. Security SLAs is an important area of study, however, current SLAs did not resolved all issues related to security SLAs. Although this paper tries to cover most of the security requirements, there are still more requirements that need attention from SLAs designers. They have to be aware that security is something that cannot be handled in isolation from other requirements.

In [79], the author presents Service Level Agreements for Security (Sec-SLAs). In Sec - SLAs, the author tires to provide an overview of what service levels the provider will offer to the customers and the difficulties faced during the security metrics definition process. The Sec-SLA deals with the "what" good security metrics should be, not the "how" they are implemented.

There is an important remark that should be noted about SLAs. Although they offer a good framework for data security measurements in the cloud, there are no guarantees that cloud providers will follow these security measures. Because there is not any good way to monitor cloud providers in the cloud.

3.2 Data Confidentiality

Currently, cloud computing systems pose serious limitation to protecting users' data confidentiality. When data is moved to the cloud, there are some changes to user's data. First, the data will be stored away from the customer's local machine the control over the data will be provided by the cloud provider. Second, the data is moving from a single-tenant to a multi-tenant environment. These impose high risk over users' data in the cloud [80]. In the following subsections, we shall explore the data confidentiality from external as well as internal attacks through the data life cycle. In other words, we will explore data confidentiality in transit, at rest, in use and data remanence.

3.2.1 Data-in-transit

When using a public cloud, user's data as well as user's credentials are transmitted over the network links to either be stored or processed over the cloud. Therefore, it is important to ensure the security of the communication channel between the cloud customer and the cloud service provider. Moreover, in cloud model, the data storage and processing is logically centralization in the cloud side which results in a major increase in the amount of network traffic flowing over the Internet backbone. The Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) [60] protocol is a network security protocol that provides secure key agreement and encrypted network traffic between client and server. This protocol uses symmetric encryption for data transmission security. However, relying on these network security protocols for providing confidentiality and integrity of network data transport in cloud communication systems would be infeasible and inefficient in the cloud environment.

The main disadvantages of these encryption protocols (e.g. SSL and TLS) are: 1) They provide rigid security services with complex APIs which make their

implementation difficult and make them prone to development errors. This can also have negative effect on the network security of the entire cloud service; 2) They never consider the particularities of the data components comprising the enterprise service and their security requirements. They encrypt all data transmitted over the network with the same encryption algorithm irrespective of their sensitivity. This can lead to unnecessary or even energy-inefficient for some cloud services supporting battery-operated client devices as in the case of mobile clients. Therefore, we have to design network security protocols that are easy to implement and specialized in content categorization to cope with the nature of cloud services [81].

X. Wang et al. [82] proposed a way for ensuring data security in transmission. All the data to be transmitted will be encrypted with homomorphic encryption, thus improving the security of data, even if the data is stolen, there is no corresponding key cannot be restored. In other words, the user is the only person who knows the key, while the clouds do not know the key. While Y. Xiang et al. [83] propose a secure protocol for trusted path suitable for the Cloud Computing Environment (CCE). The proposed protocol analyzed the current protocols and proposed the one that most fits the requirement of the data security over the network. The proposed protocol make use of session keys and tickets to ensure security of communication channel while it delegate the authentication to the Application Server (AS) and Ticket Granting Server (TGS).The proposed scheme have advantages over SSL in that it gets session key and ticket by surpassing cryptographic protections unlike SSL which uses only symmetric key that once detected by an adversary can cause replay attacks in the form of man-in-the-middle attacks.

When using cloud computing, the users have to perform remote user authentication which requires a secure channel for confidentiality of user authentication information. The majority of remote user authentication systems use SSL/TLS protocol for securing the communication channel. However, these protocols are vulnerable to phishing, web spoofing and man-in-the-middle (MITM) attacks. The two main reasons for making this protocol vulnerable are: 1) the end user usually does SSL/TLS server authentication poorly which leads to a situation in which the user communicate with MITM, thereby revealing his/her credentials; 2) Developers decouple SSL/TLS session establishment from user authentication which helps the MITM to reuse the credentials they revealed to spoof on the users.

A. J. Choudhury et al. [84] propose a strong user authentication framework that provides identity management, mutual authentication, session key establishment between the users and the cloud provider. The proposed scheme ensures that legitimate user proves his/ her authenticity before entering into the cloud by using two-step verification to verify the user authenticity. Moreover, the proposed scheme uses two separate communication channels to make it difficult for the adversaries to attack in two different channels at the same time.

R. Hauser et al.[85] propose a SSL/TLS session-aware user authentication (TLS-SA) protocol for secure user authetication solving the MITM problem. The proposed scheme does not depend only on user's secret credentials when performing user authentication but also depends on state information related to the SSL/TLS session in which credentials are transferred to the server. K. Sarikaya et al.[86] proposed three protocols for user authentication of SSL/TLS extension based on username/password information. The first

protocol uses ElGamal encryption and client certificate in order not to enable authentication of an attacker with reveal of secrets in the server's database. As a result, it supports perfect forward secrecy (PFS) in which the attacker cannot learn any information about past sessions. The second and third protocols use Chebyshev polynomials. While in the second protocol, the security of protocol is based on the security of server's database. Therefore, the attacker is able authenticate itself to the server, if the attacker is able to compromise the database. As a result, this protocol is not able to satisfy PFS. Finally, the third protocol it makes use of session-specific random values in order to create the pre-master secret key. Therefore, an attacker cannot calculate the pre-master secret key, even if he compromises the server's database.

3.2.2 Data-in-use

When using public cloud computing, the data transferred to the cloud is usually encrypted using standard methods to secure the operations and the storage of the data. Since the processing of data is performed on a remote server, this implies that the cloud providers need to access the raw data for processing purpose. Therefore, encrypting data before uploading it to the cloud introduces much difficulty to performing effective processing over the data. As a result, a method is needed to execute operations on encrypted data without decrypting it. In addition, this used method shall provide the same results after calculations as if it has been done directly on the raw data. In other words, we want to delegate processing of data to remote server without giving away access to it. Searchable encryption provides a solution for searching for keywords over encrypted data. While fully homomorphic encryption provides a solution for performing operations and calculation over encrypted data.

Song et al. [87] proposed the first practical approach for symmetric searchable encryption. In this approach, each word in the document is encrypted with a special two-layered encryption algorithm. Later on, users can search this encrypted data with certain keywords. Goh [88] proposes an improvement to the work done in [87] by formalizing a definition of the security requirements of searchable symmetric encryption. The proposed scheme by Goh introduces a bloom filter to construct secure indexes for the keyword search. This filter allows the server to examine whether the document include a certain keyword without decrypting the entire document. Curtmola et al. [89] try to overcome the shortcomings presented in [87] [88], by considering the adaptive adversaries which could generate queries depending on the outcomes of previous queries. Curtmola et al. propose an adaptive security definition for searchable encryption schemes by using "index" approach. The proposed approach build an array and a look-up table to contract the entire document collection. In each entry of the array, an encryption of document identifier set is stored associated with a certain keyword. On the other hand, look-up table is used to allow the users to locate and decrypt the appropriate element from array.

R. Koletka et al. [90] propose improvement over the previous searchable encryption techniques by providing a searchable secure storage for searching over encryption data. Through this scheme the user will be able to store their data securely on an un-trusted cloud storage service with the ability to search this encrypted data. Client/Server architecture model is used in this scheme for secure search for user's data. The client side of the architecture will perform all the cryptographic operations, while the server side will perform the search operations over the encrypted data. In addition, this scheme supports secure sharing of files between users with the ability of users to search through

encrypted content and to return results matching such queries.

M. S. Islam et al. [91] propose an efficient scheme for similarity searchable symmetric encryption. The proposed scheme uses locality sensitive hashing (LSH) based on secure index for fast similarity search in high dimensional spaces for encrypted data. In addition, the proposed scheme maintains confidentiality by defining strict security requirements and following them.

M. Li et al. [92] propose a fine-grained authorization framework for authorized private keyword search (APKS) that allows data owners to share their files with data users for performing keyword search over them. In this scheme, the data owner encrypts his data with a keyword index to be available for authorized data users for performing search over it. The authorized users obtain their privileges over the data through a local trusted authority that assign to each user certain attributes that satisfy his search privileges. In addition, the authors propose two solutions APKS over encrypted data based on Hierarchical Predicate Encryption (HPE). In the first solution, the APKS provides enhancements in the efficiency of search by using attribute hierarchy. While in the second solution, APKS provides enhancements in query privacy via the help of proxy servers.

Although a lot of work have been done in searchable encryption, this work still is not suitable for data sharing in cloud computing. Because the current searchable encryption schemes for data sharing work by distributing private key on authorized users and this implies that operations are performed only by a group of users with the private key leading to the risk of key exposure and key abuse. In addition, this methodology of search makes user's search and decryption revocation very hard. Moreover, the currently used

methodologies do not provide an efficient solution for searching in multi-user system with differential searching privileges [93].

J. Li et al. [93] proposed novel framework for searching in multi-user system with differential searching privileges on hybrid cloud, which is composed of a trusted private cloud and public cloud storage. The proposed framework provides keyword-based search on encrypted data for authorized users without sharing the same private key. In addition, the authors offer a two-layered access control to achieve fine-grained sharing of encrypted data. In the first layer, the trusted private cloud provides access control mechanism so as to realize users' authorization and revocation. While in the second layer, the data owner is responsible for enforcing access control and restricting users' access to the encrypted data.

Homomorphic Encryption is a method that is used to perform operations over encrypted data without decrypting it. Recently, cloud computing deploys full homomorphic encryption to perform dynamic as well as static operations over encrypted data so as to preserve the confidentiality of the encrypted data in the cloud. C. Gentry [94] proposes the first fully homomorphic encryption scheme for performing operations over encrypted data stored in the cloud. The proposed scheme evaluates an arbitrary number of additions and multiplications over data that enables it to perform any type of function on encrypted data. M. Brenner et al. [95] propose a secret program on an un-trusted resource for performing dynamic operations over encrypted data using fully homomorphic encryption. The secret program uses algebraic homomorphism that constructs boolean circuits as cryptographic foundation for encrypted storage access with encrypted addresses and encrypted branching.

M. Tebaa et al. [96] propose a scheme for executing calculations over encrypted data using fully homomorphic encryption. Initially, the user encrypts his data before uploading it to the cloud. Later on, he/she asks the cloud service provider to perform a number of calculations over encrypted data using fully homomorphic encryption and returns the results back to the client that can then decrypt the data. The resulted results are the same as the ones performed on unencrypted data on local machine.

Although fully homomorphic encryption provides secure computation over outsourcing data, all these results are theoretical results that have not been applied in practice. In addition, fully homomorphic encryption has not offered a clear encryption-based access control till now [97].

3.2.3 Data-at-rest

Data-at-rest means that the data that is stored in a readable form on a cloud computing service, whether in a storage product like S3 or in a virtual machine instance as in EC2. Since the data is stored on the cloud, users loss their control over their data and this is the main issue of data at rest. In order to protect data-at-rest, we need to prevent unauthorized access from users sharing the same storage in the cloud infrastructure and from system administrators who run the cloud computing service from reading the data. Moreover, we need to protect data against data alteration and theft. There are a number of protection mechanisms that can be used to protect data at rest as encryption, marking data with different access levels to enable access control, and integrity verification. In addition, backup techniques, such as a redundant array of independent disks and data recovery, insure against data loss [98] [99].

Traditionally, most the cloud storage service providers will be offering a way to protect their customers' data from disclosure by unauthorized insiders or outsiders. A common way used by service providers to protect user data is encryption of data before the storing it in the cloud. This implies that the providers will be responsible for the storage, encryption/decryption, and key management of the data. Although this appears to be a convenient way of providing easy data access for their customers from everywhere and also of allowing customers to share their files with others, it forces the customers to put a high level of trust in the cloud storage service provider. As a result, customers will lose their control on the file encryption process and on who may have access to their data.

A straight forward solution to this problem is user encrypting his data before uploading it to the cloud and saving the encryption keys on his side and decrypts it when he needs to retrieve it. However, this solution appears to be a simple solution, it suffers from a number of problems. First, the user has to do all computation on his side and the communication traffic between the user and storage servers is high. Moreover, not all the users have the computation power to encrypt the data on their side. Second, it is the responsibility of the user to manage his cryptographic keys. If for whatever reason the keys stored on user's machine are lost or compromised, the users will not be able to retrieve the data forever.

W.-G. Tzeng et al. [100] propose a threshold proxy re-encryption scheme and erasure codes for securing users data. In this scheme, data is encrypted locally by user then the user distributes his cryptographic key to key servers that shall perform cryptographic functions on user side. In the other hand, the threshold proxy re-encryption is used to encode, forward, and partial decryption operations in a distributed way. For example,If

the user wants to forward his data to other users, he calculates a re-encryption key using the other user public key and send it to cloud storage which in turn uses proxy re-encryption to transform the user's encrypted data into data encrypted in the other user secert key.S. Kamara et al. [101] propose a cryptographic scheme for providing confidentiality and integrity of data stored in the public cloud storage. The proposed scheme is composed of four components that reside on user machine. These components are data processor (DP) which is main responsibility is encrypting data before uploading it to the cloud; and data verifier (DV), which is responsible for ensuring the integrity of data; and a token generator (TG), that generates tokens that enable the cloud storage provider to retrieve segments of customer data; and a credential generator that is responsible for implementing the access control policy for the file sharing. However, the proposed scheme tries to solve the problem; it leaves the key management process for the user as each user has his private master key for encryption and decryption.

Amazon S3 [28] provides server-side encryption for data at rest. The data stored on Amazon cloud servers is encrypted by an encryption key and an S3 master key and both keys are stored at Amazon's servers. Amazon S3 encrypts users' data while it is performing write operation to the disks in its data centers and decrypts it when users request access to it. Therefore, Amazon is able to arbitrarily decrypt data. As a result, it cannot provide confidentiality or integrity. A similar solution is provided by Dropbox because Dropbox rely Amazon S3 for its backend storage. Wuala [102] is cloud storage services that is similar to Dropbox, but it provides client side encryption to users' files. Based on the user's password, a public-private key pair is generated that serves as the basis for confidential file sharing. Every file is encrypted with a symmetric key that is

derived from the file's content and the user's password. However, if the user forgets his password, it is not possible to recover decryption credentials. Other solution for providing data confidentiality at rest can include encryption tools as TrueCrypt or BoxCryptor[103][104] that provide encrypted virtual hard disks that can be mounted into the user's local file system and synchronized with. Since encrypted data by itself is not feasible for sharing among users, these tools will not help much unless Advanced Encryption Standard (AES) key are shared among participating users over a secure out-of-band channel. This is clearly a drawback in terms of usability.

J. Hwang et al.[105] propose a Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service. In this model, cloud users will make use of two cloud computing service providers. The first provider will used to perform encryption and decryption of data while the other is used for storing encrypted data. This models implies that an agreement between these service providers to establish a model for cooperation and division of responsibilities in providing a common service to clients. This model seems to be infeasible since it is very difficult that to guarantee that these providers will not collude with each other. V. Joshi et al [106] propose a general framework for securing the data by using three tier securities in cloud environment. In the first tier, the problem of data leakage is solved by classifying the data based on importance of data to either confidentiality, integrity, availability (CIA) before storing it in the cloud. Data is encrypted before uploading it to the cloud. Each of the classifications (CIA) offers different security level to solve the problem. While in the second layer, the clients are categorized according to the three rings. Each ring represent certain users with certain privileges (e.g. ring 1=> core users, ring 2=> employees, and

ring 3=> external users).Users within the same ring belong to same group and have same privilege. This aggregation of users helps in preventing unauthorized access and increase security level. Finally in the third tire, a face fuzzy vault technique is used to provide unique identification where it binds ring secrete key with user's face feature to provide controlled data access to authorized user. The proposed system works initially when the user/company uploaded data marked to either CIA and certain ring. Later on, when a certain user requests for data access, user authentication and authorization the tier architecture are used to allow only authorized users to have access to the data. L. Hao et al. [107] propose a cloud security storage system for securing data in private cloud storage systems. The proposed system tries to solve the problems of information isolation, accessing control, virus detection, metadata safeguard of crucial data and fast-speed retrieval. The proposed system designs a prototype of private cloud security storage system. The prototype consists of five entities: connect interface, a distributed file system, an access control module, a security-auditing module and a classification-write module. Each of these entities tries to solve the problems listed above. F. Rocha et al.[108] in their paper showed that malicious insider can steal user's confidential data in the cloud, therefore, users have not to trust cloud provider for securing their data. The paper is divided into two parts. In the first part, the authors show four attacks performed by malicious insider to steal user's data. These attacks are compromising passwords, cloud users' private keys, files, and other confidential data that might be extracted from a hard disk. If the malicious insider succeeds in any of these attacks, he can easily get access to all users' data without users being aware of this unauthorized access. While in the second part, that author discusses how a set of recent research mechanisms fail to

protect data from the previous attacks, however, this does not mean that these mechanisms are not useful.

3.2.4 Data Remanence

Through the data life cycle, it is important not only to secure data storage, processing and transmission but also to secure the deletion of data. Typically, in most storage media when a file is deleted, only the file name is removed from its directory or folder, while the file's content remains stored on the physical media until the data blocks are overwritten [109].In order to ensure the confidential of deleted data, three methodologies can be accomplished: 1) physical destruction of the storage medium; 2) overwriting all of the sensitive data; 3) secure overwriting the key of encrypted sensitive data.

1. Physical Destruction

Data can be deleted physically by many ways including smelting, shredding, sanding, pulverization, or acid bath. Another method that erases data permanently from the storage media is magnetic degaussing that work by exposing a hard drive platter to an inverted magnetic field, which leaves data unrecoverable [110]. Although, these methods for data deletion assuredly delete the data permanently, they make the storage medium unusable. Moreover, this method is not suitable for deleting only one file. Therefore, this method does not support flexible security policies. As a result, physical destruction is not a suitable method for assured file deletion in the cloud environment that contains huge data centers [111].

2. Data Overwriting

There are three main methods for deleting data securely from electronic storage media using data overwriting. These methods are: 1) overwriting the file content; 2)

deleting the file normally, and then overwrite all free space in the partition; 3) erasing the entire disk or partition. All the three methods are easy to use. However, they are not suitable for cloud environment, because are initiated by the cloud provider and there is no guarantee that the provider is honest enough to do so [112].

3. Erasing the key of encrypted data

The third method to assuredly delete data from storage media is erasing the encryption key of the data. Usually most of the data stored on the cloud is encrypted, therefore, deleting the encryption key will make the data inaccessible and we are sure that no one can access the data. Although, this method suits the cloud environment, it requires that all data stored in the cloud is encrypted data. Moreover, it requires that the data encryption is not performed by the cloud provider because if the provider encrypts the data, he will have full control over encryption keys and there will be no guarantee that he assuredly deleted the encryption keys [113].

In the cloud environment, most of approaches used to assure the deletion of files use cryptographic protection, which removes the cryptographic keys that are used to decrypt data blocks to make the encrypted blocks unrecoverable. Y. Tang et al. [114] propose a policy-based assured deletion, in which data can be assuredly deleted according to revoked policies. While R. Perlman[115] proposes a scheme for assured deletion with three flavors; 1) deletion based on expiration time known at file creation; 2) deletion of individual files on-demand ;3) deletion based on custom keys for classes of data. In this scheme data is encrypted on nonvolatile storage, and keys are destroyed at the appropriate times. Whereas A. Rahumed et al. [116] proposed a layered scheme of cryptographic protection where the data is encrypted with the first layer of keys called the

data keys, and later on these data keys are encrypted with another layer of keys called the control keys. Each control key corresponds to a fine-grained policy that specifies how each file is accessed. In order to access a file you must have both the data key and the control key. Therefore, if the policy associated with file is revoked, then its associated control key is deleted and the file will be inaccessible. However, this scheme assuredly deleted the file; it would be difficult to manage the keys with the increase of number of files and versions.

3.3 Access Control

Access control in cloud is gaining attention as being critical security mechanisms for data protection in cloud applications by only allowing authorized users have access to valid data. Since large amount of information stored in the cloud is sensitive, care should be taken for access control of this sensitive information. Unfortunately, traditional data access control approaches used to solve this problem assumes that data is stored in a trusted data server for all users and the cloud service provider(CSP) is in charge of enforcing the access policy. However, this assumption cannot hold in cloud computing because this approach gives CSP has access to the plain data. In addition, the data gets compromised once the CSP gets compromised. Moreover, the access control policy is not bound to the data because the access control policies are maintained by CSP (CSP have full control over policies).Thus users do not have mechanisms to bind the access control policy to the data, they can rely on the server to enforce access policy[19].

In general, access control is divided into three types: User Based Access Control (UBAC)[117], Role Based Access Control (RBAC)[118], and Attribute Based Access Control (ABAC). In UBAC, users are authorized to access the data based on an access

control list (ACL) which contains a list of users who are authorized to access data. This type of access control not feasible in cloud because there are many users each with different privilege and this makes the management of keys impossible. In RBAC, users are authorized to access the data based on a role he/she is assigned to by the system. Only the user with the matching role can access the data. In ABAC, users are given attributes and the data is attached with access policy. Users having set attributes satisfying the access policy can easily access the data.

The first trials to protect sensitive data shared in the cloud is to encrypt the data (using either symmetric-key, also known as private-key encryption(PKE), or asymmetric-key)before uploading it to the cloud storage**,** while the decryption keys are disclosed only to authorized users. However, this trivial solution succeeded in the beginning later on it brings in a number of problems. The first problem associated with solution is that it requires an efficient key management mechanism to distribute decryption keys to authorized users, which has been proven to be very difficult. Moreover, with the wide spread of cloud computing, more users joined the cloud which make adopting the previous solution inefficient as it lacks scalability and flexibility. Furthermore, when a data owner wants to revoke a data user, all data related to this user has to be re-encrypted and new keys must be distributed to the remaining data users. Last but not least, data owners need to be online all the time so as to encrypt or re-encrypt data and distribute keys to authorize users. Moreover, these methods incur high storage overhead on the server, because the server should store multiple encrypted copies of the same data for users with different keys [8,119].

Attribute Based Access Control (ABAC): An access control method in which the user requests to perform operations on objects. These operations are granted or denied based on assigned attributes of the subject and a set of policies that are specified in terms of those attributes and conditions.

Attributes are characteristics of the subject, object, or environment conditions. Attributes contain information given by a name-value pair.

A **subject** is a human user or non-person entity (NPE), such as a device that issues access requests to perform operations on objects. Subjects are assigned one or more attributes.

An **object** is a system resource for which access is managed by the ABAC system, such as devices, files, records, tables, processes, programs, networks, or domains containing or receiving information. It can be the resource or requested entity, as well as anything upon which an operation may be performed by a subject including data, applications, services, devices, and networks.

Policy is the representation of rules or relationships that makes it possible to determine if a requested access should be allowed, given the values of the attributes of the subject.

Each subject that uses the system must be assigned specific attributes. The user is established as a subject within the system by an administrator and characteristics about that user are captured as subject attributes. This subject may have a name, a role, and an organization affiliation. Other subject attributes may include US Person status, nationality, and security clearance. These subject attributes are assigned and managed by an authority within the organization that maintains the subject identity information for the file management system. As new users arrive, old users leave, and characteristics of subjects change, these subject attributes may need to be updated. Consider the example of the

headmaster who wants to encrypt a document to all the professors of 45 years old in the computer science department, the document would be encrypted with access structure ("professor" ∧" CS department" ∧ "age 45"), and only the users who hold the private key containing these three attributes can decrypt the document while others cannot get any information from the ciphertext.

Subject attributes are provisioned by attribute authorities—typically authoritative for the type of attribute that is provided and managed through an attribute administration point. Often, there are multiple authorities, each with authority over different attributes. For example, Security might be the authority for Clearance attributes, while Human Resources might be the authority for Name attributes.

Every object within the system must have at least one policy that defines the access rules for the allowable subjects, operations, and environment conditions to the object. This policy is normally derived from documented or procedural rules that describe the business processes and allowable actions within the organization. For example, in a hospital setting, a rule may state that only authorized medical personnel shall be able to access a patient's medical record. In some system, if the object is a document with a RecordTypeAttribute of PatientMedicalRecord, then the MedicalRecordRule will be selected and processed so that the subject with a PersonnelTypeAttribute value of NonMedicalSupportStaff trying to perform the Read operation will be denied access and the operation will be disallowed. Note that this is only one approach to implementing the connection between attributes and rules[215].

Attribute-based encryption (ABE) is a cryptographic primitive that addresses the above issues and finds applications to a wide range of settings, from regular users over the world

wide web to large multi structural corporations. Being different from an identity-based encryption (IBE)[121] , attribute-based encryption (ABE) provides a sound solution to encrypt a message for all users who hold the required attributes, without any knowledge of their exact identities. The first ABE scheme was proposed by Sahai and Waters [120] based on linear secret sharing, where both the ciphertext and the secret key are labeled with a set of attributes. A user can decrypt the ciphertext if and only if there is a match between his secret key and the ciphertext. This idea was originally used to design an error-tolerant (or fuzzy) IBE. In other words, ABE relates the cryptographic components with attribute sets, corresponding to available credentials for users, and access policies, corresponding to the possibly complex restrictions that the credentials have to satisfy. Since its introduction, two complementary schemes have been proposed, which are: key-policy ABE (KPABE) [122][123] and ciphertext-policy ABE (CP-ABE) [124].In a KP-ABE scheme, the ciphertext is defined by a set of attributes; while the secret keys of the user are associated with an access policy (access structure).A user can decrypt the ciphertext, if and only if he has the required secret keys corresponding to attributes listed in the ciphertext. As a result, the encryptor does not have entire control over the encryption policy because the encryption policy is described in the keys. Therefore, the encryptor has to trust the key generators for issuing correct keys for authorized users. Furthermore, KP-ABE is not naturally suitable to certain applications. An example of such applications is a type of sophisticated broadcast encryption, where users are described by various attributes and the one whose attributes match a policy associated with a cipher text can decrypt the cipher text. On other hand, Ciphertext policy attribute-based encryption (CP-ABE) is becoming a promising cryptographic solution to this issue in KP-ABE. It enables data owners to define their own access policies over user attributes and

enforce the policies on the data to be distributed. In CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution. The authority can be the registration office in a university, the human resource department in a company, etc. The data owner defines the access policies and encrypts data under the policies. Each user will be issued a secret key according to its attributes. A user can decrypt the ciphertext only when its attributes satisfy the access policies. Moreover, in CP-ABE schemes, the access policy checking is implicitly conducted inside the cryptography. That is, there is no one to explicitly evaluate the policies and make decisions on whether allows the user to access the data[124], [125]. Most of the ABE approaches take a centralized approach and allow only one single authority [114,126-133] for issuing users' keys. Although single authority ABE achieves fine grained access control ,it works well only in the setting where data is managed within one organization or trust domain. In addition, it still suffers from failure or corruption, which may leak out the data because the authority can decrypt all the encrypted data. Furthermore, the authority has full control over users' keys so it is able to decrypt all users' encrypted data. Moreover, the authority may become the performance bottleneck in the large scale cloud storage systems.

To address this issue, multi-authority or decentralized attribute-based access control schemes[17,123,134-140] were proposed, where multiple parties could play the role of an authority. Although, multi-authority ABE tries to solve the problem of single authority CP-ABE, it needs to tie together different components of a user's secret key from multiple authorities(AA). [123,134,135] suggest using a central authority to provide a final secret key to integrate the secret keys from different attribute authorities. However, the central authority would be able to decrypt all the ciphertext in , since it holds the master key of the system.

Thus, the central authority would be a vulnerable point for security attacks and a performance bottleneck for large scale systems. To overcome this problem [136,137] , propose a multi-authority attribute-based access control schemes without a central authority. [136,137] presented secure multi-authority CP-ABE scheme that remove the central authority by using a distributed PRF (pseudo-random function). But it has the same limitation of defining a pre-determined number of authorities in the system initialization. In addition, they can tolerate collusion attacks for up to *N-2* authorities' compromise . Besides, they degrade the performance of the system due to interaction among the authorities during the system setup. [138] is similar to [136,137], as it has a pre-determined set of authorities, however, it requires the interaction among the authorities during the system setup. Moreover, this scheme can tolerate collusion attacks for up to *m* colluding users, where *m* is a system parameter chosen at setup time.[139] proposed a new comprehensive scheme that is secure against any collusion attacks and it can process the access policy expressed in any Boolean formula over attributes. However, their method is constructed in composite order bilinear groups that incur heavy computation cost. In addition, they did not consider attribute revocation, which is one of the major challenges in multi-authority access control for cloud storage.[140] presents a fully secure multi-authority CP-ABE scheme in the standard model. In this system, there are multiple CAs and AAs. Each CA or AA operates independently from the others. Before requesting the attribute-related keys from the AAs, the user must ensure that he has obtained the identity-related keys from all the CAs.[17-18], eliminates the collusion problem associated with previous work while maintaining high performance and attribute revocation.

3.4 Revocation

Revocation is the act of preventing any future usage of the key towards decrypting ciphertexts within an encryption scheme. Revocation takes two forms: user revocation and attribute revocation. User revocation takes place when one or more attribute are revoked, the user loses all the decryption privilege of all the ciphertexts (e.g., a user is leaving a company). Attribute revocation takes place when one or more attribute are revoked, the user still can use its other attribute to decrypt ciphertexts (e.g., a user is degraded from PM to Developer).User revocation can be solved by either broadcast revocation or dynamic revocation or by utilizing proxy re-encryption (PRE)[126,133,141] to delegate most tasks to servers. In a broadcast encryption scheme, the sender (broadcaster) sends a ciphertext to a group of recipients such that only non-revoked users inside the group can decrypt the broadcasted content. Such a scheme allows the broadcaster to specify the list of revoked users who are not allowed to decrypt the digital content that is broadcasted. The main drawback of these schemes is that the private key size blows up by a multiplicative factor of log(*n*), where *n* represents the maximum number of attributes in the system. In addition, Broadcast ABE requires knowledge about the list of all possible users during encryption. Knowing the list of all possible users in advance do not provide secure systems[142-144].Dynamic revocation was supplied by Xu and Martin [145], which allows the revocation of keys without requiring any modifications to ciphertexts or other keys. This scheme uses a proxy which is supplied with an additional part of the ABE keys, and uses this part of the key at decryption through an additional pairing. Like other schemes, secret sharing is used to convert the user's share with the values from the proxy in order to get the proper value to compute the pairing. Overall, this method is not compatible with the goals of this thesis due to the fact that it is based around storage-centric environments, which already handle some

issues with key management by having files on a trusted server. The keys revoked through this are revoking the access from the trusted servers, not local keys from users. Similar work is presented in [146], instead of dividing the ABE key, they split the symmetric encryption key and encrypt part by the CP-ABE encryption algorithm while the other part is maintained at the server.

On the other hand, attribue revocation can be realized by revoking attribute itself using timed rekeying mechanism, which is implemented by setting expiration time on each attribute. Indeed, these approaches have two main problems. First problem is the security degradation in terms of the backward and forward secrecy . An attribute is supposed to be shared by a group of users in the ABE systems by nature. Then, it is a considerable scenario that membership may change frequently in the group that shares an attribute. Then, a new user might be able to access the previous data encrypted before he comes to hold the attributes until the data are re-encrypted with the newly updated attribute keys by periodic rekeying (backward secrecy). On the other hand, a revoked user would still be able to access the encrypted data even if he does not hold the attribute any more until the next expiration time (forward secrecy). Such an uncontrolled period is called the window of vulnerability[124,147,148]. Ibraimiet al. [149] provides an option for attribute revocation within a CPABE scheme. However, this mediated CPABE scheme does not provide much functionality with regards to revocation, as it is not immediate revocation, requiring users to wait until a time period ends. The revocation is performed by requiring a mediator to hold each half of the user's key for each attribute. These two shares are combined after decryption, with both the user and the mediator decrypting using their shares, giving a single decrypted message. An advantage of this scheme is that it does allow the mediators to be

distributed, not requiring a single proxy server. However, each proxy still has to contain the same shares of every user's key, and a single compromised server can still give a malicious user the half of every key. Due to the fact that this only provides revocation of single attributes, and the fact that it is not immediate. Moreover, these attribute revocation methods are designed only for ABE systems with single authority.

The attribute-based signature (ABS) is a recent cryptography primitive, in which a signature does not attest to the identity of a signer, but to a policy regarding the attributes possessed by the underlying signer. The advantages of un-forgeability and signer privacy make ABS a good prospect in access control and anonymous authentication systems. Digital signatures in general are needed for a variety of security services, including data integrity, authentication, non-repudiation and certification (in conventional PKC). The main security goal is to prevent forgery, i.e. preventing someone not in the possession of the secret key producing a valid signature. In this sense, ABS is similar to signature variants like Group signatures [150], Ring signatures [151] and Mesh signatures [152]. The dominant idea of all these signature primitives is that they allow the signer fine-grained control over the amount of personal information exposed. However, it is important to note that a valid ABS signature guarantees that only a person possessing the required attributes that satisfy the predicate can produce a signature. The basic concept of ABS, however, has a serious problem that only a single authority exists in a system. Therefore, ABS should take into account the scenario of multiple authorities, which is more likely to be used by real world applications. The concept of *multi-authority* (MA-)ABS, was introduced [153-157], in which there are multiple authorities and each authority is responsible for issuing a secret key associated with a

category or sub-universe of attributes, i.e., a user obtains several secret keys, each of which is issued by each authority.

3.5 Proxy Re-encryption

In a proxy re-encryption scheme, introduced by Mambo and Okamoto [158], a proxy is a semi trusted entity which can transform an encryption computed under Bobs' (delegator) public key to an encryption computed under Alice's (delegatee) public key. The proxy is a semi-trusted entity i.e. it is trusted to perform only the ciphertext re-encryption, without knowing the secret keys of Bob and Alice, and without having access to the plain data. Blaze, Bleumer and Strauss [159] introduced the notion of "atomic proxy functions" - functions that transform ciphertext corresponding to one key into ciphertext corresponding to another key without revealing any information about the secret decryption keys or plain data. However the scheme presented in [159] is bidirectional where one re-encryption key can be used to transform ciphertext from the delegator to the delegatee and vice versa, and is useful only for the scenarios where the trust relationship between involved parties is mutual. To overcome this situation Jakobsson[160] and Zhou et al. [161] proposed a quorum-controlled protocol where a proxy is divided into many components. Dodis and Ivan [162] propose a number of unidirectional proxy re-encryption for El-Gamal, RSA and IBE scheme, where the delegator's secret key is divided into two shares: one share for the proxy and one share for the delegatee. The drawback of the proposed schemes is that they are collusion-unsafe, i.e. if the proxy and the delegatee colludes then they can recover the delegator's secret key. Matsuo[163] and Green and Atteniese [164] propose the identity-based proxy re-encryption scheme, where the encrypted data under the public key generated by delegators' identity is re-encrypted to an encrypted data under the public key generated by delegatees' identity.

3.6 File Sharing

With the evolution of cloud computing, many individuals and organizations store and share their data in the cloud. This implies that the data owners have limited control over their outsourced data and that the cloud service provider has excessive privileges in terms of control over user's data. This leads to very low level of trust on keeping and sharing data on the cloud. Therefore, security of data in cloud storage and through the sharing process is a must. Data security in cloud storage implies that only authorized users have access to the data and even if an attacker can directly read the content of the disk containing confidential data, he cannot get understandable plain data. Moreover, the data security in the sharing process requires fine grained sharing. In other words, partial access permissions of the confidential data can be shared to the others with satisfying the least privilege constraints

G. Zhao et al.[165] propose a scheme for implementing scalable and fine-grained access control systems based on attribute-based encryption (ABE). This scheme tries to prevent the usage of illegal key sharing among colluding users by defining and enforcing access policies based on data attributes through user accountability. Through this method, user specific information is inserted with user's attribute private keys so that user's attribute private keys can be viewed as default attribute. Therefore, if a user is able to share his decryption keys with others, he is not able to change the user specific information in the attribute private keys.

W. Lafayette et al. [166] propose a scheme for delegating privacy-preserving fine-grained access enforcement to the cloud based on a recent expressive Group Key Management (GKM) scheme; unlike the current methods which delegate most of

computation for managing keys for access control and encryption for data owners, the proposed scheme delegate most of the activity for the access control enforcement to cloud storage service; W. Lafayette et al. proposed a two layer encryption (TLE) approach. This approach is based on two layers of encryption for the uploaded data to the cloud. In the first layer, the data owner performs a coarse grained encryption over his data before uploading it to the cloud in order to ensure the data's confidentiality. While in the second layer, the cloud provider performs fine grained encryption over the encrypted data uploaded by the data owner based on the attribute-based access control (ABAC) policies provided by the data -owner. Therefore, any change in a policy or user dynamics will require change only in the outer layer of the encryption. Because the outer layer encryption is performed by the cloud provider, there is no need for data transmission between the data owner and the cloud. As a result, most of the computation for managing access control keys is performed by cloud provider.

S. Ruj et al.[167] propose a decentralized access control scheme that not only enforces fine-grained access control but also preserve the authenticity of the user without knowing his/her identity. The proposed scheme make use of the two protocols attribute based encryption (ABE) and attribute based signature (ABS) in order to enable only valid users to decrypt the stored information and to verify the user's credentials without knowing the identity of the user who stores information. However, the main limitation of this scheme is that the cloud knows the access policy for each record stored in the cloud.

Another category of cryptographic access control focuses on time-based access control. An example of this type is web-based electronic newspaper that allows users to subscribe to one package for a certain period of time (e.g., a week, a month, or a year).

Time control is of particular significance and has been concerned in access control. Y. Zhu[168]proposes a scheme for temporal access control for cloud services using three cryptographic techniques: integer comparison, proxy-based current-time re-encryption and attribute-based encryption (ABE). In the proposed scheme, user's data is associated with an access policy on a set of temporal attributes, e.g., period-of-validity, opening hours, or hours of service. Users satisfying assigned privileges and specified time period are allowed to access the data.

In [169], the author tries to have trusted data storage and sharing over entrusted cloud storage providers. To ensure data confidentiality in the storage, the author encrypts all the data before storing it on the cloud. While for the shared data, the encrypted data will be re-encrypted without being decrypted. The re- encryption of data will allow the authorized users only to have access to the data. The whole process from encryption and re-encryption does not reveal the plain data to the cloud provider and allows only authorized users to access the data according to the designated permissions from the data owner.

The process of securing the stored and shared data can be summarized in five steps: 1) The data owner encrypts his data and stores it on a service provided by a Cloud Storage Provider; 2) A data user sends a request to the data owner asking for access permission to the data; 3) The data owner sends credential to the Cloud Storage Provider for the re-encryption of the data;4) The data owner sends credential for data user to decrypt the re-encrypted data with his private key; 5) the data user acquires the re-encrypted data from the Cloud Storage Provider and decrypts it.In [170], the author tries to utilize public key cryptography by allowing users to dynamically derive the symmetric

keys at the time of decryption in order to efficiently handle policy changes as well as adding /revoking users or identity attributes. To do so, the author formalize a new key management scheme, called broadcast group key management (BGKM) in order to support attribute-based access control while preserving privacy of users' identity attributes . By making use of BGKM scheme the author construct a secure construction called ACV-BGKM. ACV-BGKM works by assigning some secrets to users based on their identity attributes and later allow them to derive actual symmetric keys based on their secrets and some public information. In other words, the user is able to decrypt the content he is authorized depending on the attributes they have received from the data owner. Moreover, the proposed scheme handled adding users/revoking users or updating access control policies efficiently by only updating some of the associated public information.

In [171], the author tries to solve the problem of revoking users without key-redistribution and data encryption. The proposed scheme makes use of attribute-based/predicate encryption and proxy re-encryption to revoke users without re-encryption from the data owner. It works as follows: the data owner encrypts the actual data using symmetric key encryption (use random key K for symmetric encryption). Then he picks another random key K1 and computes K2 = K⊗ K1.later on encrypts K1 using attribute-based encryption, and encrypts K2 with proxy re-encryption under owner's own public key. From the previous we note that, users that have both K1 and K2 can obtain the data. For example, when a data user requests data from the cloud, if he is allowed to access to data, the cloud should have the re-encryption key that enables it to transform the part of the cipher text corresponding proxy re-encryption into one data user public key. Thus the

data user can obtain k2 and as long as user have access rights then he can get K1, therefore, he user is able to access data easily as he has both K1 and K2. In order to revoke a user, all the data owner has to do is to simply command the cloud to destroy the re-encryption key and create a new re - encryption key. Therefore, the data user will not be able to access data with his old keys

In [172], the author proposes a solution that prevents the leakage of unauthorized data when a revoked user rejoins the system different access privileges to the same data record. The solution uses homomorphic encryption and proxy re-encryption schemes to solve the problem. Moreover, the proposed solution tries to prevent the information leakage in case of collusion between a user and the Cloud Service Provider by using data distribution technique. Unlike the proposed scheme in [171] which suffers from problems in the re-authorization of revoked users, who rejoin the system but with different access privileges, and collusion between a user and the Cloud Service Provider. The proposed scheme tries to offer one solution that solves the following issues: 1) achieve fine-grained data sharing and access control over data in the cloud; 2) prevent the leakage of unauthorized data when a revoked user rejoins the system; 3) prevent collusion between a user and the Cloud Service Provider. The proposed solution [Secure Data Sharing (SDS) framework] is composed of 5 stages.

1. Key Generation and Distribution- In this stage the data owner generates two of key pairs based on homomorphic encryption. Then he/she distributes them to the cloud and to the data users. Moreover, the data owner generates the proxy re-encryption key for each authorized users.

2. Data Outsourcing- In this stage the data owner encrypts his data and generates the

authorization tokens associated with this data. Then data owner sends the encrypted data along with the associated the authorization tokens to the cloud.

3. Data Access- In this stage the data user requests the data from the cloud service provider who in turn checks whether the data user is an authorized user or not and takes the corresponding action.

4. User Revocation- In this stage the data owner commands the cloud to remove the authorization token corresponding to data user.

5. User Rejoin- In this stage, the data owner grants access to a data user who was revoked some time ago. The data owner generates an authorization token corresponding to the new set of attributes and sends the token the cloud

Moreover, the proposed solution solves the problem of information leakage in the case of collusion between a user and the cloud by distributing the encrypted data and authorization tokens corresponding to the data between two clouds. The author argues that this solution is valid as the user can collude with at most one of the clouds.

In [173], the author proposes a public-key deniable scheme that protects the data privacy against powerful adversaries who can force users into opening their encrypted content. In particular, the author uses plan-ahead deniable encryption scheme in the context of file sharing among collaborating users so that if any participant is forced to decrypt one or more shared files, these files just open the non-deniable (fake) files without revealing any sensitive information. The developed prototype of a deniable file system DenFS considers the cloud storage as un-trusted and does not rely on ACLs for access control. Therefore, DenFS implements access control through public key encryption. In order to make use of DenFS file system, the user has to determine the

mount point, the backend directory, and a password. The backend directory is the directory containing sensitive looking files that are used to fill the non-deniable part of a deniable encryption. This directory is called non-deniable pool. DenFS can be mounted to either deniable or non-deniable mode. When non- deniable mode is used, random data is used to fill the deniable part of the cipher text. On the other hand, when deniable mode is used, the non-deniable part is filled with the encryption of a file selected from the non-deniable pool. By following this methodology, any adversary who has access to several snapshots of the encrypted content of the shared folder is unable to have a clearly distinguish between deniable and non-deniable operations by comparing snapshots.

3.7 Data Integrity

Ensuring the integrity of data through its transmission requires ensuring the security of the communication channel from attacks that eavesdrops data such as man-in-the-middle attacks. These attacks are cryptographic attacks that take place when an attacker can place himself in the communication's path between the users, which enables his to Data Integrity in Cloud Storage modify users' data [174]. Moreover, current systems using cloud communicated with each other anonymously, therefore, full trust between cannot be assured between these system[49].As a result, a secure communication and execution service should be supported to prevent the interception and tampering of sensitive information. In current cloud systems, the integrity of data-in-transit can be guaranteed by the SSL protocol through encryption techniques. However, the data integrity depends on not only the security of both the uploading and downloading sessions, but also the security of the data in the storage media [175].

To ensure the integrity of the data stored in the cloud, we have to make sure that the data is securely stored on cloud servers with no violations (e.g., data is lost, altered, or compromised). Data stored in the cloud faces two main threats to its integrity: 1) Data loss/ manipulation; 2) Dishonest computation in remote servers. Data can be lost or modified in the cloud either maliciously or accidentally. This loss can take place by either administration errors (e.g., backup and restore, data migration) or by adversaries that initiate attacks benefiting from data owners' loss control over their own data. On the other hand, since the user's data is outsourced to the cloud for storage and processing, there are no transparent measures that can be used to ensure the integrity of computation executed over user's data by cloud provider. The cloud provider may behave unfaithfully and return incorrect computing results [176].Therefore, we have a set of requirements that should to be satisfied to face the two threats above:1)The verifier has to ensure data integrity without maintaining any copies of the data; 2) the protocol should be efficient in terms of communication;3) It could be possible to run the verification an unlimited number of times;4) The integrity checking have to support dynamic data operations as insert, delete, and modify; 5)Allowing Public verifiability: Public verifiability allows a trusted entity other than data owner or service provider to perform the integrity checking operation;6) Privacy should be maintained if the data is verified by a third party verifier in order to prevent in data leakage. (not by a client), the protocol must ensure that no private information contained in the data is leaked; 7) the data integrity should be lightweight by allowing users to perform storage correctness checks with minimum overhead [174].

In order to overcome these threats two solutions are available: 1) Provable Data Possession (PDP);2) Third Party Auditor (TPA). Provable Data Possession is used to check the integrity of static data by client sending a challenge to the cloud server. The server answer for the challenge helps the client to determine to prove whether the integrity of the data is violated or not [177]. The Provable Data Possession has many variations as Original Provable Data Possession, Scalable PDP, and Dynamic PDP. In addition, there are two similar schemes that are Proof of Retrievability (POR) and High-Availability and Integrity Layer (HAIL).Scalable PDP is considered to be an improved version of PDP. It is better than PDP in the following:1) It uses symmetric key encryption instead of public-key leading to reduction in computation overhead; 2) it supports dynamic operations over remote data. However, it have the following limitations: 1) It does not support public verification due to symmetric encryption 2) it has to pre-compute the all challenges and answers; 3) the number of updates is limited and fixed as a priori [178].While Dynamic PDP provides improvement over PDP by supporting full dynamic operations such as append, insert, modify, and delete; However, the efficiency of their scheme remains in question [179]. In addition, Proof of Retrievability (POR) which was first presented by Juels presents a protocol that enables clients to check the integrity of static data only stored on cloud storage without having to retrieve it. The client stores the data files along with set of sentinel values embedded in each file. The client checks the integrity data by sending a challenge to the cloud server asking for subset of sentinels in F'. If the server is notable to solve the challenge, then there is high probability that data will be corrupted or lost. It is considered to be a lightweight auditable protocol[16].Furthermore, HAIL is considered to a distributed setting where the client

distribute the file across multiple servers with redundancy and only store a small constant state in local machine[180].On the other hand, TPA is trusted by the cloud users and providers for performing the task of integrity checking for users data[176].R. S. Kumar et al.[181] propose a protocol for Proof of retrievability (POR) in cloud. The proposed protocol tries to verify that the data stored in the cloud storage is not modified by the archive and thereby the integrity of the data is assured. These verifications prevent the cloud storage archives from altering or deleting any part of the data without the permission of the data owner. In addition, the proposed protocol reduces the computational and storage overhead on the client by only storing two functions on client side for checking the integrity. These functions are the bit generator function g, and the function h which is used for encrypting the data. Finally, the proposed protocol reduces the size of the proof of data integrity so as to reduce the network bandwidth consumption.

L. Wenjun et al. [182] propose a protocol for ensuring data integrity by using HLAs and RSA signature with the support public verifiability. The usage of public verifiability allows the client to delegate the integrity checking process over the data to the TPA. On the other hand, RSA signature is used with large public exponent for enhancing data storage security.

W. Zhi-wei et al. [183] constructed a new scheme called aggregatable signature based broadcast (ASBB) that is used to build an efficient homomorphic public verifiable scheme with zero knowledge privacy that supports static data only. The proposed scheme prevents any adversary from deducing any information through the audit interaction that take place between cloud server and the TPA. Therefore, eliminating the load of auditing task from the cloud user. In addition, the TPA reduces the client fear from outsourced

data leakage. C. Wang et al.[184] propose approaches and system requirements needed to be performed in order to make publicly auditable secure cloud storage service become a reality. These include a set of systematically and cryptographically desirable properties that can be used to make public auditing service a reality in the cloud storage services. In other words, the purpose of the paper is to make use of a publicly auditable for delegating the auditing process to a trusted entity (TPA) instead on depending on client for verifying the data integrity. K. Yang et al. [185] propose a privacy-preserving auditing framework for supporting auditing of dynamic data. He proposed framework protects users' data privacy in the auditing process from external auditor by using cryptography method with the bi-linearity property of bilinear paring. In addition, it moves the computing loads of auditing from the auditor to the server leading to less communication cost and less computation cost on the auditor side. Q. Wang et al.[186] construct a model for allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The constructed model allows the TPA to perform improved proofs on dynamic data by manipulating the classic Merkle Hash Tree construction for block tag authentication. In addition, it enables the TPA to efficiently handle multiple auditing tasks by using bilinear aggregate signature techniques. L. Li1[187] propose a TPA mechanism integrated into file sharing system that is build atop of service delivery platform for achieving reliable system.

From the above, we can conclude that most of PDP and POR schemes do not consider the privacy protection of users' data against external auditors. Because the cloud service provider may reveal users' data to auditors or adversaries during the auditing. In addition, most of the work done in data auditing considers only static data files with little attention

given for the dynamic data updates. Furthermore, the current provable data possession (PDP) or proof of retrievability (POR) schemes support for dynamic operations over data lead to security loopholes. Finally, the auditing models that support both public audit-ability and data dynamics are not fully addressed in the research context.

3.8 Data Availability

Although there is a great effort done by cloud provider to ensure the availability of the data, there is still service interruption. In the year 2008, four of the most popular cloud providers announced their service outage. These service providers include AWS, Google App Engine, and Gmail. Each of these service providers was not available from 1.5 to 8 hours [188]. The reasons for these outages include software protocol error, programming error authentication service overload, and outage in other contact systems [189].

One of the most famous solutions to address the availability issue is data redundancy technique. This technique can be categorized into replication-based solutions and erasure codes-based solutions. Data replication is the process of having multiple identical copies of the same data on different locations. It is works by copying data from healthy server to corrupted server when the data is corrupted on any server. The two main disadvantages of replication-based solutions that make them infeasible to apply in cloud are: 1) high storage cost; 2) high-throughput requirement, where in cloud there exist a large number of users who access the service at the same time and each user wants to access different pieces of data on a server leading to less cache hits but frequent disk I/O requests [190]. On the other hand, erasure code provides redundancy by breaking a file into smaller

chunks and storing the chunks in different locations. Data can be recovered from any combination of a smaller number of those chunks. Therefore, when erasure codes-based solutions are compared to replication-based ones, they achieve higher reliability level with much less data redundancy [191]. Moreover, erasure codes-based solutions better fits the cloud environment because every block of data on a server is useful for decoding the original data, which leads to a high cache hit rate of the system. However, the main disadvantage of existing optimal erasure is the high communication cost needed for repairing a corrupted storage server.

Despite the efficiency of most data redundancy techniques to recovery data, disasters can still take out an entire data center and make service unavailable. This means that relying on a single cloud service provider exposes data to a single point of failure, despite the fact that the failure probability is very small. Therefore, the cloud user should not rely on a single cloud service provider but he/she should employ multiple cloud service providers [192].

Y. Singh et al. [193] propose a secured cost-effective multi-cloud storage (SCMCS) scheme. The proposed scheme works by dividing user's data block into data pieces and distributing them among different service provider in a way that at least a threshold number of service providers can be used to successful retrieval of the whole data block. This helps in preventing adversary from retrieving data if he was able to intrude in one network, because the complementary pieces of data are stored in the other networks. Moreover, the proposed scheme use a redundant distribution scheme that enables it to retrieve the data even if adversary causes a service outage even in one of the data networks, because there is another server which maintain the same block. P. Liu et al.

[194] propose a scheme for selecting multiple cloud providers based on a mathematical model. The model is used to derive two algorithms that help in selecting multiple clouds but the selection must be within a given budget. In other words, the proposed scheme wants to replicate the data over multiple cloud providers but the replication is constrained to the cost and performance requirements. E. Pardede et al. [195] propose a Multi-clouds Database Model (MCDB) based on Shamir's secret sharing algorithm for supporting multi-clouds service providers instead of single cloud service provider. The Shamir's secret sharing algorithm is used in this model to divide the data among the different cloud service providers. Bessani et al. [196] propose a virtual storage cloud system called DepSky. Depsky is constituted from different clouds to build a cloud-of-clouds so if one of the cloud providers is damaged, users are still able to retrieve data correctly. The proposed system achieves availability by using multi-cloud providers, combining Byzantine quorum system protocols, cryptographic secret sharing and erasure codes.

4 Proposed Approach for Securing Data in cloud Storage

4.1 Introduction

This chapter explains the details of our secure cloud storage framework. The suggested framework transfers the trust from the cloud to a trusted third party service. This service provides security for users' data with minimal overhead on cloud users. Particularly, this service ensures data confidentiality against cloud and unauthorized users. Data confidentiality against cloud can be achieved by storing data in an encrypted format in cloud storage so that malicious insiders are notable to view/decrypt it. Given the vulnerabilities found in both client and cloud side encryption services [54, 64, 65, 103,201], we propose addressing these vulnerabilities by deploying a trusted third party (TTP) service. This TTP service has encryption/decryption service that can be employed either locally or remotely according to level of severity of the data. This service shall remove the burden of key management and maintained from data owners. Moreover, this service takes advantage over the current software encryption/decryption service that offers full disk encryption and client side storage services that can encryption keys. For achieving data confidentiality against unauthorized users, the TTP service collaborates with a number of attribute authorities (i.e. more details in section 4.2.1) to achieve fine grained access control. By doing so, we prohibit cloud service providers and unauthorized users from getting access to owners' plaintext or credentials, unlike most of the currently available cloud storage services that either not do not provide file sharing services or give the cloud service provider full power over access control. In addition, we support user and attribute revocation without depending on the data owner for re-encrypting the affected files or regenerating system parameters and users' keys.

Moreover, we provide read or write or both accesses to a file stored in the cloud, unlike most of the systems that supports 1- write-many-read. Last but not least, we shift most of the heavy computations such as verification and re-encryption from the owner/user to the cloud.

4.2 Models and Assumptions

4.2.1 System Model

We consider a secure cloud storage system with multiple authorities, as shown in Fig.1. The system model consists of six entities: certificate authority (CA), attribute authorities (AAs), the cloud storage provider (CSP), trusted third party (TTP) service, the data owners (owners), and the data consumers (users).

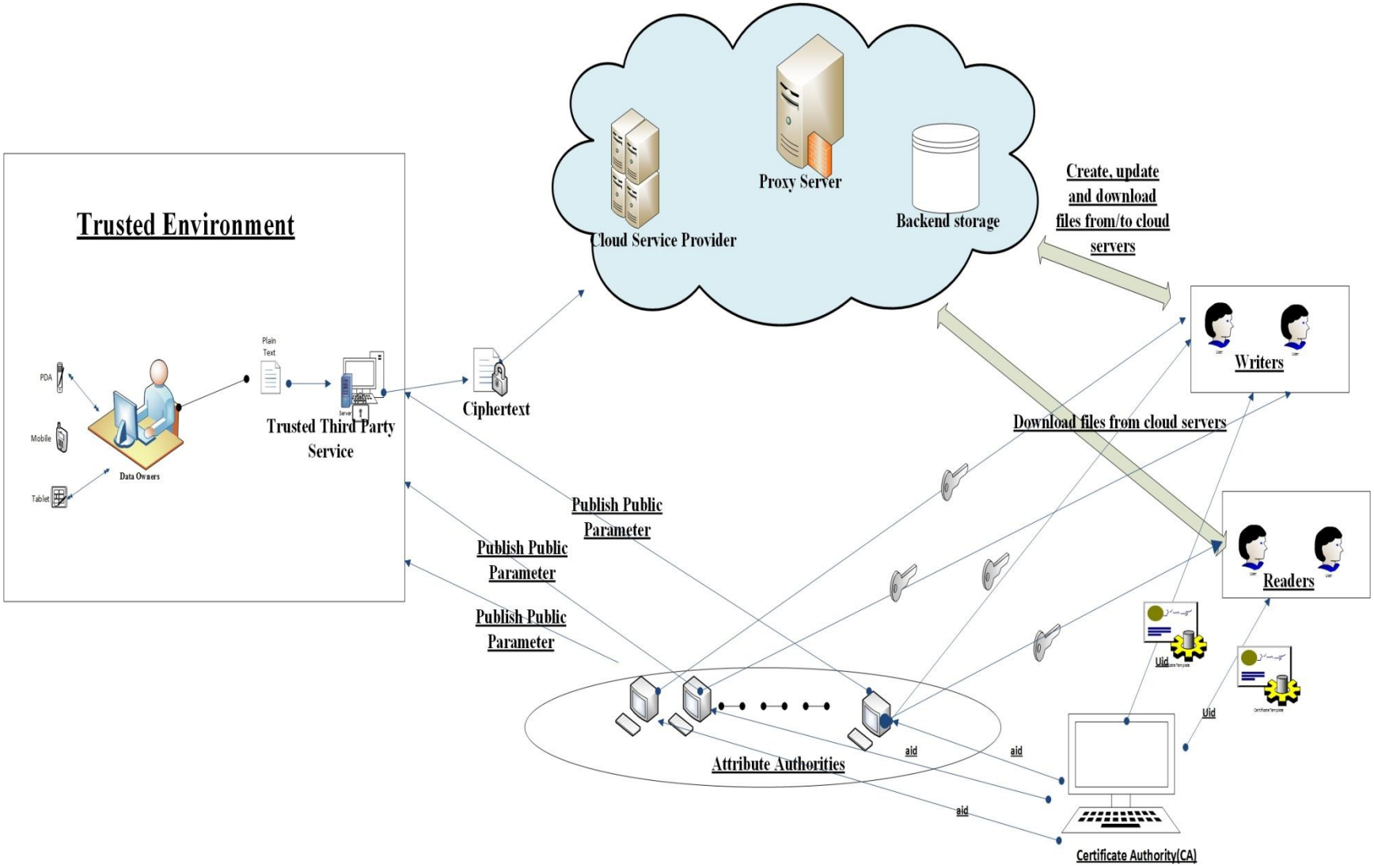

Figure 4-1: Secure Cloud Storage Service Design

Certificate Authority (CA) is a global trusted certificate authority in the system. It sets up the system and accepts the registration of all the users and attribute authorities (AAs) in the system. The CA is responsible for the distribution of global secret key (i.e. (SK_{aid},GSK_{uid}) secret key for attribute authority and user respectively) and global public key (i.e. (PK_{aid},$\mathrm{GPK}_{\mathrm{uid}}$) public key for attribute authority and user respectively) for each legal user and attribute authority in the system. However, the CA is not involved in any creation of secret keys or management of attributes [18]. The CA can be an independent government agency.

Attribute Authorities (AAs) are a set of trusted entities which take on the responsibility of issuing, revoking and updating users' attributes according to their role or identity. Each AA is an independent attribute authority by itself and it does not communicate with other authorities for issuing or revoking users' keys. Each AA can manage an arbitrary number of attributes, but every attribute is associated with a single AA.

The cloud storage provider (CSP), which includes a proxy server, is a semi-trusted entity. It is responsible for providing data storage service (i.e. Backend Storage Servers) and verification of users' data ciphertexts before it is stored in the cloud. Proxy servers are servers that are always available for providing various types of data services (i.e. proxy re-encryption technique). Proxy Re-Encrypt is a cryptographic technique that transforms the cipher text from one secret key to another without revealing the secret key to the proxy server. For example, ciphertext encrypted with Alice secret key can be transformed to another ciphertext that can be decrypt by Bob without revealing any

information to the proxy server. Homomorphic encryption is the most appropriate algorithm used with re-encryption technique. In our framework, we delegate most laborious tasks of user/attribute revocation to proxy servers without leaking any confidential information to them [159,208].

Trusted third party (TTP) service: is an independent entity that is trusted by all other system components, and has capabilities to perform extensive tasks (i.e. encryption, decryption and signature). It maintains a key management service that creates, manages, and destroys user's data files encryption and decryption keys (DEK). It stores these keys in a trusted hardware to ensure better level of security. It is also responsible for encrypting, decrypting and signing users' data. In addition, it does not store any data at its end as it is only confined to provide security service.

Data owner encrypts the data with the help of trusted third party (TTP) service (which could be local or remote). Then, the owner defines the access policies over attributes from multiple attribute authorities. The access policy ($\bar{A}$) can be expressed by a monotonic access tree (T). The access tree (T) has attributes at its leaf nodes and logic gates e.g., AND ($\wedge$), OR ($\vee$) as intermediate nodes. The AND gates can be constructed as n-of-n threshold gates and OR gates as 1-of-n threshold gates [124]. As an example in figure4.2,access tree = (class2010 $\wedge$ (Department of Neurology $\vee$Department of Computer Science $\vee$Department of Biology)) $\wedge$ (Teaching Assistant $\vee$University Professor).

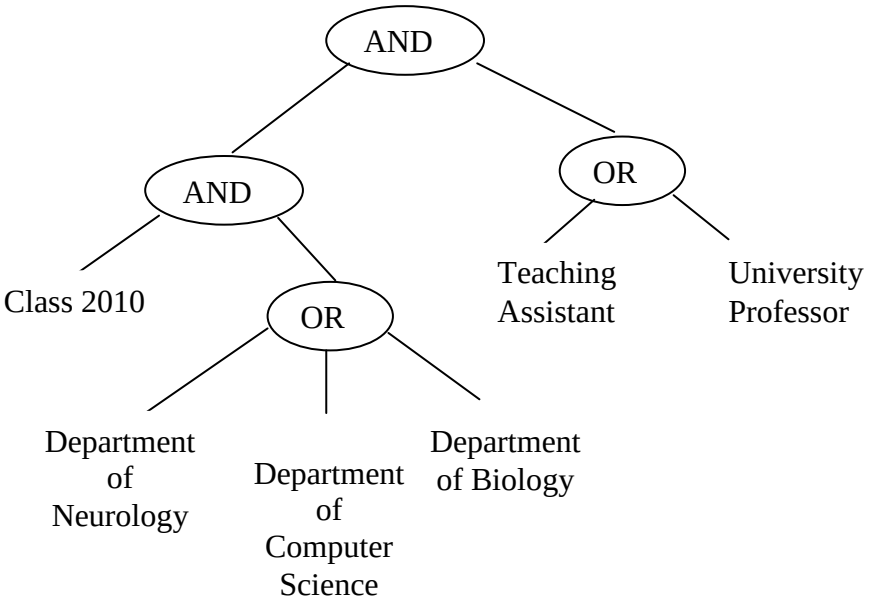

Figure 4-2: Access Structure

The figure illustrates an access structure that allows faulty of class 2010 that are either teaching assistants or university professors belonging to either neurology or computer science or Biology department to access the data.

Next, he sends the policies to TTP service in order to encrypt the content keys under these policies, before transmitting the ciphertext to the cloud service provider (CSP). The owner does- not rely on the CSP to do data access control. Instead, the ciphertext can be accessed by all the legal users in the system, which means that any legal user who has been authenticated by the system somehow, can freely download any ciphertexts from the CSP. The access control happens inside the cryptography. That is, only when the user's attributes satisfy the access policy defined in the ciphertext; the user is able to decrypt the ciphertext. Thus, users with different attributes can decrypt different number of content keys and thus obtain different granularities of information from the same data. As illustrated in figure 4.3, the data owner defines the access policy ($\bar{A}$) before uploading

data to the cloud. Then, he encrypts the data with the access policy and distributes decryption keys to users (Bob, Alice) according to their roles in the system. In our example, Bob is able to decrypt the ciphertext because the attributes associated with his keys satisfy access policy while Alice is not able to decrypt ciphertext because her attributes do not satisfy access policy.

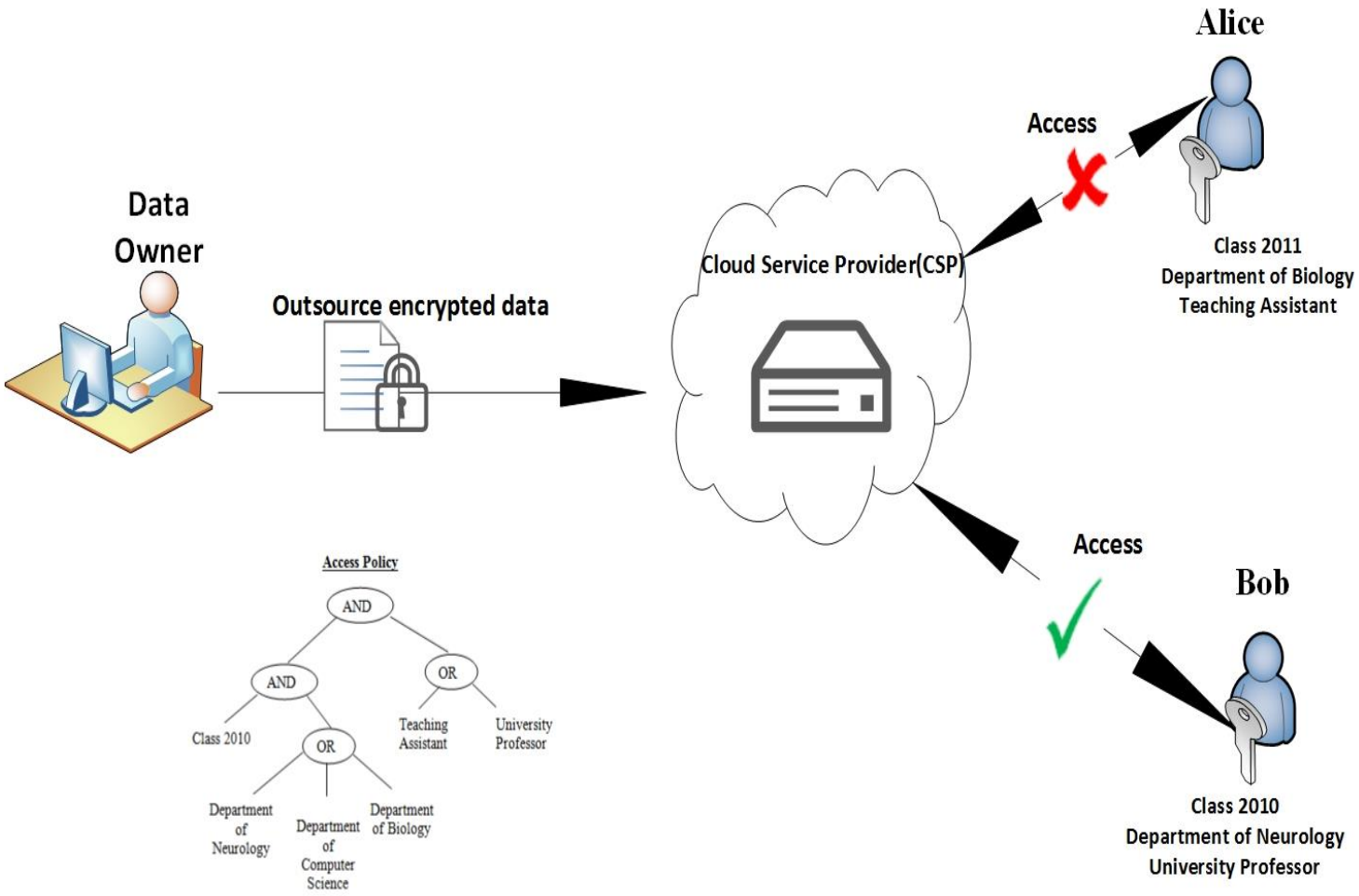

Figure 4-3: Data upload and download

Each **user** has a global identity in the system. A user can be either a reader or a writer and a reader who may be entitled a set of attributes. As an example in figure 4.4, users who only possess attributes (“class 2010”∨“class 2011”)∧ (“Biology”∨“Neurology”)can decrypt (read) the file, called readers; users who possess attributes “class 2011”∧ (“(“Biology”∨“Neurology”) can not only read the file but can also update the file. we differentiate writer from reader not at the individual user level, but at the attribute level. Once the owner creates the file, he defines both the decryption policy and verify policy. After that, the file is uploaded to the cloud storage server, the update policy (ABS’s access structure) will be sent to the cloud storage server for authenticating writers and readers. Since the decryption and verification are not executed simultaneously, we are able to differentiate readers from writers. The attributes that user possess may come from multiple attribute authorities according to his role in the domain. The user will receive a secret key associated with its attributes entitled by the corresponding attribute authorities. Any user can download the encrypted data from the cloud server. But only the user who owns proper attributes can successfully decrypt the encrypted data. Since users obtain key from different authorities in multi-authority CP- ABE, we need to prevent collusion attacks between users. Collusion resilience implies that if multiple authorized or unauthorized users collude by combining their attributes to decrypt a ciphertext that none of them can decrypt alone, they are not are not able to do so. Collusion attacks are built into the CP-ABE security game of [124], where the adversary may make multiple secret key queries both before and after selecting challenge plaintexts.

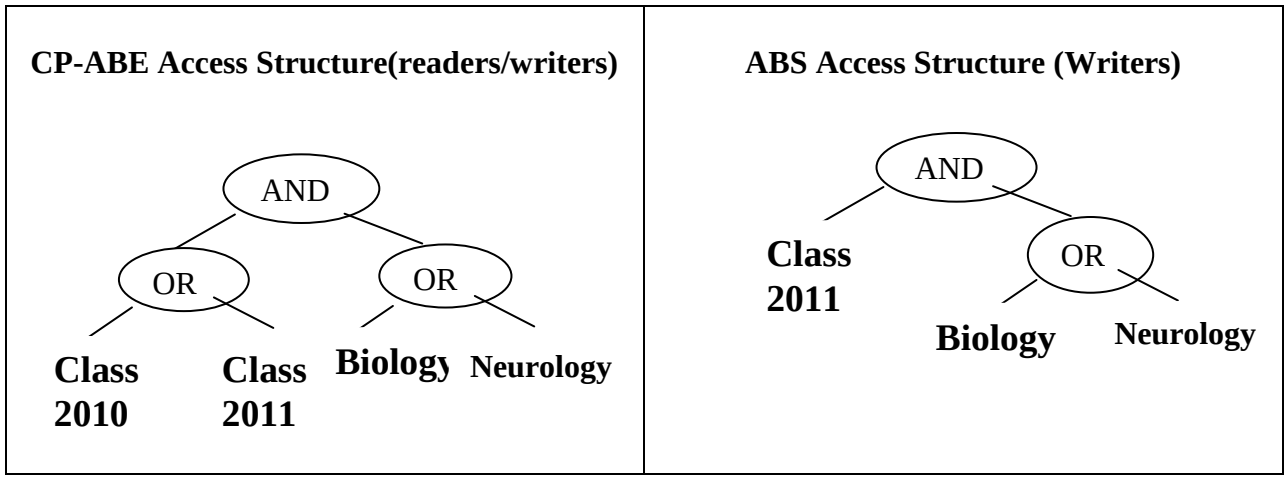

Figure 4-4: Access Structure

4.2.2 Threat Model

Threats faced by the owner when outsourcing confidential data to a cloud server can be primarily divided into two categories, internal and external threats. For internal threats, it is considered to be the most dangerous type attack because either the cloud service provider or its employees can be self-interested, un-trusted, and possibly malicious. Moreover, it may attempt to hide a data loss incident due to management errors, Byzantine failures, and so on. For external threats, leaking data confidentiality may come from an external beyond the control domain of CSP, for example, the economically motivated attackers. Once data is encrypted, CSP cannot learn any information from it that can be utilized to compromise privacy of the outsourced data. However, these encryption algorithms cannot protect the user from the external threats, because users can derive valid decryption key and access outsourced data according to these access structures. To eliminate these external threats owner must ban unauthorized data access by defining access control policy and enforcing it in the entrusted domain. Since, cloud server is not a trusted entity, it should not be able to differentiate between an

authorized and fraudulent user, yet being able to assist unauthorized user in deriving decryption key.

4.2.3 Assumptions

In this thesis the following assumptions are considered with the intent of simplicity. These assumptions conform to the security model, and do not undermine any of the privacy threat:

a. We consider the cloud service providers (CSPs) are honest but curious. In other words, CSP is expected to be curious to learn data content and has full access to everything stored in the cloud, but will honestly follow any protocol provided by the Data Owner (DO). The honest but curious assumption seems realistic since correctly executing its tasks is of personal interest as well. When tasks are not correctly executed no customers will use the service. Since correct operation is of personal interest for the DSSP the amount of computational power, the storage space and the bandwidth is assumed to be considerable.

b. The certificate authority (CA) is fully trusted in the system. The CA is used to certify the attribute authorities and the users that wants to join the system and provide global secret/public keys to both attribute authorities and the users respectively. Therefore, the fully trusted assumption seems realistic since the CA will not collude with any user or authority.

c. The AAs honestly distribute the keys and send the key updating message, but some of them may be corrupted by the adversary which attempts to find out information of the

data file as much as possible. We also assume that the AAs will never collude with any user. This assumption seems realistic since attribute authorities, unlike the single attribute authority setting, allow the adversary to adaptively create corrupt attribute authorities and learn some of the honest authorities' secret keys as long as there is at least a single honest attribute authority managing one of the attributes required for satisfying the policy used in the forgery.

d. We assume that legitimate users behave honestly, by which we mean that they never share their decryption key with the revoked users. This is a reasonable assumption since this internal person is chosen by the client company itself. Correct execution and confidentiality is considered to be at his personal interest as well since it will be one of his or her evaluation criteria. If the user and data manager is corrupt, the security of the entire system is compromised. Also, it is assumed that the user will not collude with the data storage service provider

e. Users can have either read or write or both accesses to a file stored in the cloud.

f. All communications between users/clouds are secured by SSL/TLS protocol in order to secure the data in transit.

g. Whenever needed, the access to cloud storage service is always available by wired network, Wi-Fi, or 3G mobile network. We will not consider the exceptional cases such as connection absence, discontinuity or transmission failure.

4.3 Security Requirements

To build a secure cloud storage services that can secure data, we recognize the following unique but not necessarily complete security requirements:

4.3.1 Data confidentiality against cloud

Although storing data in cloud storage saves the cost of its management and maintenance, it is exposed to a huge number of security threats. Data may be compromised through its transfer, use and at rest. Thus, when using cloud storage, these security threats must be mitigated. In this section we shall explore two of the main data security requirements which are data confidentiality and integrity, while we assume that data availability is satisfied.

1. Confidentiality

Data confidentiality is one of the most important aspects of data security. It can be defined as the assurance that sensitive information is not revealed to unauthorized users, processes or devices. In cloud environment, the data owner trusts the cloud storage service for managing his data but he/she does not want the cloud to access the data. Therefore, we need to assure that the cloud storage provider and other unauthorized users are incapable of learning the content of the stored data. In addition, the secret key used for the encryption should be securely protected so no snooping or stealing of the key is possible. Therefore, measures should be taken so as to protect users' confidential data from cloud service providers and external attackers. In this thesis, we only focus on data confidentiality for data in transit and at rest only and leave data in use as a future work.

2. Integrity

Another important requirement is the integrity of the data. In the cloud computing model, the mobility of data increases the threats that can affect the data integrity. As the data is being transmitted to and from the customer and the cloud service provider, and also internally within the cloud. Therefore, data integrity should be guaranteed within the cloud. Integrity can be defined as the protection of data from unauthorized modification that can take place either maliciously or accidentally during processing, storage or transmission of data. Specifically, the data owners or writers are the only ones that have the privilege to modify the data. Any read operation from data users should be consistent with an update from an authorized user. Any unauthorized modification should be detected by the user and/or the cloud storage provider.

4.3.2 Data confidentiality against accesses beyond authorized rights

In reality, an increasing number of people host their sensitive data in cloud storage services. Usually, they need to share this data with different users with different access privileges over various types of data. In this case, the data owner should define flexible access control policies on his data. Therefore, we want to manage the access control and key distribution of user's data in a way that ensures that unauthorized user and the cloud server cannot access the data and allow users to use the cloud resource for data operations. Specifically, new joining users are able to decrypt the data stored in the cloud before their participation, and revoked users are unable to decrypt the data they have access to in the cloud before the revocation. In this section we shall explore the two main security requirements that achieve efficient access control in cloud: fine grain access control and efficient revocation.

1. **Fine grain access control**

In fine grained access control, data access policy for each user is defined at different data item level based on the user's role in the system. This data access policy should be enforced at each access attempt without the data owner's involvement. In particular, the access policy should be able to define a unique set of data items that the user is authorized to access, and must be enforced via a cryptographic method. In other words, we want the data owner to define unique access control that designates the type/set of files that the user is allowed to access. In addition, we want to prevent CSP from learning either the plaintexts of data files or user access privilege information. All these security goals should be achieved efficiently in the sense that the computation load should be affordable to all types of cloud users.

2. **Efficient user Revocation**

In practical application scenarios, users may join or leave the system frequently. Therefore, we need an effective and efficient user management mechanism that deals with user access privilege revocation. Revocation can take place in one of the two following cases: a) revoking a subset of attributes from a user (attribute revocation), b) revoking the minimal attribute set form a user (user revocation). Specifically, we want to support scalable revocation to take advantage of the abundant resources in the cloud by delegating the ciphertext update to cloud instantly and efficiently. In order to ensure efficient revocation, we shall satisfy two properties: collusion-resistance and forward/backward secrecy. Collusion resilience implies that if multiple authorized or unauthorized users collude by combining their attributes to decrypt a ciphertext that

none of them can decrypt alone, they are notable to do so. On the other hand, forward secrecy means that the revoked user cannot decrypt the ciphertext which is created after he is revoked. While backward secrecy means that the new user are able to decrypt the data stored in the cloud before their participation.

Along with these basic security requirements, we consider some general security features as scalability, availability and mobile access. Scalability can be achieved by allowing multiple devices/users to be connected to a cloud application simultaneously, while the security of all sessions is maintained. In addition, in an enterprise system, users may be created or removed at great frequency, and communication costs related to key management scale accordingly. Availability can be achieved in terms of key distribution services and availability of cloud servers. Key management services must always be available to users to ensure uninterrupted communication and continuity of cloud services. Mobile access should consider the challenges associated with mobile devices in terms of network rate limitations, intermittent connectivity and processing capabilities.

4.4 Design Goals

In this section, we present the technical details of our cloud-based data sharing framework, in which outsourced data, access control policy and identity attributes of a user are considered as confidential information. The proposed framework shall ensure that outsourced data can only be accessed (decrypted) by authorized users, and during the whole process cloud server is unable to learn any useful information that can lead to a potential privacy breach. To achieve the privacy of these components, our scheme

processes the data in three fundamental steps: data outsourcing, file access and revocation.

Notation	Description
CA	Certificate Authority
AA	Attribute Authority
Λ	Security Parameter
GPP	Global Public Parameter
GMSK	Global Master Key of the system
U_{aid}	Attribute Universe
Uid	User unique identifier
Aid	Attribute Authority unique identifier
U_u	u-th Owner/User
A_k	k-th attribute authority
I_u	Set of attributes that user U_u possesses
J_u	Set of attributes that user U_u possesses as claim attributes
Ÿ	Claim predicate
Ā	Access Policy
H	Hash function
CT	Ciphertext
T	Timestamp
T	Access structure(Access control)
I_A	the encryption set , set of AAs from the $S_A$involved in the encryption
S_A	set of all attribute authorities
S_u	set of all users
F	Data File
DEK	Encryption symmetric key
$S_{uid,aid}$	Set of attributes that describes the user in an attribute authority AA_{aid}
H(C)	Hashed Ciphertext
δ	Signature
ẍ$_{aid'}$	Revoked attribute from user(uid) from attribute authority AA_{aid}
(SK_{aid},PK_{aid})	Attribute's authority Private/Public Key Pair
(GPK_{uid},GPK'_{uid})	User's Global Public Keys
(GSK_{uid},GSK'_{uid})	User's Global Secret Keys
SIG_{sk}	User Certification
$\{PK_{x_{\text{aid}}}\}_{x_{\text{aid}}\in U_{aid}}$	Public attribute keys of all the attributes managed by AA_{aid}
$\{VK_{x_{\text{aid}}}\}_{x_{\text{aid}}\in U_{aid}}$	Attribute version keys of all the attributes managed by AA_{aid}
$UK_{S,\ddot{x}_{aid'}}$	Secret key update
$UK_{c,\ddot{x}_{aid'}}$	Ciphertext update key

Table 2: Notations used in our proposed scheme description

4.4.1 Data confidentiality against cloud

- **Data Confidentiality**

To our knowledge, there is still no purely software based solution for providing complete confidentiality for data stored on a public cloud from a potentially un-trustworthy cloud provider. Therefore, in this thesis, we propose a whole service that allows users to protect their data. The service protects user's data before uploading it to the cloud by an encryption service and decrypts it after it is downloaded from the cloud by decryption service. Our service can be either employed locally or build on top of a cloud storage service as a layer of protection. Since users' data does not have the same level of importance, users can divide data into three categories based on its sensitivity (i) not sensitive, (ii) moderately sensitive and (iii) highly sensitive. According to these three levels of trust, user can either use the service locally or remotely on top of the cloud storage. The service can employed locally if the user does not trust the cloud or any trusted third party for his data and his keys. While he uses it remotely when the user trusts the trusted third party for managing the service remotely but does not trust the cloud service provider. The trusted third party service does not store users' data, it only stores the encryption and decryption keys that shall be used in uploading and downloading user's data. These keys are critical components that should be handled securely, therefore, we store them separately using a Hardware Security Module (HSM) or other secure elements, as smart card, TPM or secure USB token or ARM TrustZone technology, which provide mobile device hardware-based key management [197**].** In addition, offering a flat encryption schemes without looking at the importance of data by applying the same

encryption algorithm for all types of data make the client(third party software service) machine suffer from huge computation cost[198][199][200]. Not all data offers the same value and not all require the same degree of protection even if it is encrypted locally on user machine. Therefore, the data need to be encrypted with the different encryption algorithms.

Consider the following example as an illustration of how the service works. When a user Alice wants to upload her data to the cloud, she will choose either to use the service locally or remotely. In both cases the user is requested to choose the level of sensitivity of the data. Next, TTP will chose symmetric encryption algorithm according to the level of sensitivity and create encryption keys accordingly. After that, the service uses this symmetric key to encrypt the uploaded data. After the encryption, the service encrypts the symmetric key with access control key according to user's privilege (we will explain in details this point in the next section).The user only stores the key and uploads the data to the cloud for storage. On the other hand, when Alice wants to download a file from the cloud, the data decryption service will get the key related to the file, and then use the key to decrypt the download, then Alice gets the plaintext. The details of this operation will be presented in the next section.

- **Analysis**

The proposed scheme satisfies the security requirement stated above, because all user sensitive data sent to the CSP are encrypted. Therefore, the cloud has no access to plaintext. In addition, the decryption keys are stored in hardware device

which make it inaccessible to the attacker**.** Moreover, this solution allows all cloud users to access data from anywhere and from any device allowing mobile users with limited capabilities to securely store their data in cloud.

As listed in assumptions above, to ensure data confidentiality in transmission, we make use of the SSL/TLS protocol for encrypting transmissions. This protocol to great extend overcome the network attacks as web spoofing and man-in-the-middle (MITM) attacks [83].

4.4.2 Data confidentiality against accesses beyond authorized rights

- **Fine grained access control**

In section 4.4.1, we provide a scheme to ensure data confidentiality by making the decryption keys inaccessible to attackers. However, there remain issues with sharing data with unauthorized users, efficiently revoking users' privileges without re-encrypting massive amounts of data and re-distributing the new keys to the authorized users, collusion between users. In the following subsections, we shall explore these issues.

One of the most challenging issues in data sharing systems is the enforcement of access policies. Fine-grained data access control is essential in any cloud storage service where data is shared among multiple users with different levels of trust. However, in cloud storage service, the roles of the data owner are separate from the data service provider, and the data owner does not interact with the user directly for providing data access service. Moreover, the cloud server cannot be fully trusted by data owners and traditional server-based access control methods are no longer

applicable to cloud storage systems. Therefore, fine-grained data access control with effective management of rights is considered a challenging issue in cloud storage systems.

To achieve these goals, we propose a novel data sharing protocol by combining and exploiting two of the latest attribute based cryptographic techniques, ciphertext policy attribute-based encryption (CP-ABE) [124] and attribute-based signature (ABS) [214]. Based on multi-authority ciphertext policy attribute-based encryption (MA CP-ABE) in [18] [204] and attribute-based signature (ABS) in [205] [153] [155], we provide fine-grained attribute-based access control scheme for multi authority cloud storage applications with flexible many-write-many-read framework while placing minimal trust on the cloud storage server to ensure data confidentiality. In our proposed scheme, initially, the data owner defines two flexible access policies over descriptive attributes (read access structure and write access structure). Then he calls the service to encrypt the data before uploading it to the cloud severs. The service in turn encrypts the data with symmetric key and encrypts the key by MA-CP-ABE according read access structure **Ā** and then signs the encrypted file by user's write access policies (access structure) **Ÿ**. In order to prevent reply attack, we add a period of validity t (timestamp) along with hashed ciphertext to prevent malicious reader from impersonating the valid writer by uploading an old version encrypted file with its old signature which was signed by a former writer to the cloud storage server is valid. Next, the TTP sends encrypted outsourced data$\{F\}_{DEK}$, the attribute based encrypted decryption key$\{DEK\}_{MA-CP-ABE}$, signature and claim predicate **Ÿ** to the cloud storage server. The cloud verifies the authenticity of the user without knowing the user's

identity before storing information by verifying the signature. If it is a valid signature, the cloud stores the ciphertext (CT) along with the encrypted key, otherwise reject. The cloud shall use the claim-predicate **Ÿ** to verify the modified file that will be uploaded after a write operation. Later on, whenever the owner wants to share the data with any user, he defines the set of attributes the user is allowed to access (either for read or write) and send it to the different authorities to create decryption keys that the user shall use to access the data. A user is authorized to access the data only if he possesses proper attributes which satisfy the access policy deployed in the data.

At a system level, we are interested in the following high level operations: File Creation, User Grant, and File Access.

1. **New File creation**

   The file creation process passes with two phases: Encrypt Phase and Sign Phase

   Encrypt Phase:

   Step 1: Data owner selects the file along with sensitivity level to be uploaded, defines a set of attributes I_u for read access policy and a set of attributes J_u for write claim predicate

   Step 2: It sends the file with its sensitivity level along with read access policy and write claim predicate to trusted third party (TTP) service.

   Step 3: TTP service asks the different authorities for the related public/secret keys for access policy and claim predicate based on their attributes.

   Step 4: Each AA run **SKeyGen** algorithm and return related secret keys and public keys for both access policy (**Ā)** and claim predicate (**Ÿ**)

Step 5: (TTP) service generates a symmetric key according to the sensitivity level.

Step 6: The TTP service encrypts data file (F) with symmetric key (DEK) and encrypts DEK with the different authorities' public keys $\{\boldsymbol{Pk_{aid}}\}_{aid\in I_A}$producing ciphertext CT.

$$\mathbf{CT} \longrightarrow \mathbf{CP\text{-}ABE.Encrypt(GPP,}\{\boldsymbol{Pk_{aid}}\}_{aid\in I_A}\mathbf{,DEK\ ,\ \bar{A}\ )}$$

Sign Phase

After the encryption, the TTP signs the CT both for reader/writer differentiation and for providing integrity verification to all parties that want to access the file.

Step 7: TTP service first hashes the CT which is generated in the Encrypt Phase to produce (H(c)). A timestamp is attached with hash code to prevent replay attacks (H(c) ||t).

Step 8: The hash is then signed by the secret key of claim predicate (**Ÿ)** to produce the signature **δ**

$$\boldsymbol{\delta} \longrightarrow \mathbf{CP\text{-}ABS.Sign(GPP,\ h(CT)||\ t,\ \ddot{Y}\ ,}\{\boldsymbol{Pk_{aid}}\}_{aid\in I_A}\mathbf{\},(}\boldsymbol{GPK_{uid}}$$

$$\boldsymbol{,GSK'_{uid}),SK_{uid,aid})}$$

Step 9: After the Encrypt Phase and Sign Phase, trusted third party service will send the ciphertext CT**,** the attribute based encrypted decryption key$\{DEK\}_{\{\boldsymbol{Pk_{aid}}\}_{aid\in I_A}}$, the signature **δ,** period of validity *t* and claim predicate **Ÿ { CT ,** $\{DEK\}_{\{\boldsymbol{Pk_{aid}}\}_{aid\in I_A}}$**, δ ,** *t* **,** **Ÿ }** to owner.

Step 10: The owner will upload **{** CT**,**$\{DEK\}_{\{Pk_{aid}\}_{aid\in I_A}}$**, δ ,** t **,** $\ddot{Y}$ **}** to the cloud storage provider (CSP).

Step 11: The cloud storage provider (CSP) first checks the validity of t with current time, and obtain all verification keys that corresponds to attributes depicted in the claim predicate $\ddot{Y}$ from the AAs, then verify the **δ** by the boolean value result

$$R0 \rightarrow Verify\ (GPP, h(CT) \| t, \delta, \ddot{Y}, \{Pk_{aid}\}_{aid\in I_A})$$

Step 12: If the signature is a valid signature, the CSP will accept the upload request and save the time t, $\ddot{Y}$ and verification keys with the encrypted file CT.

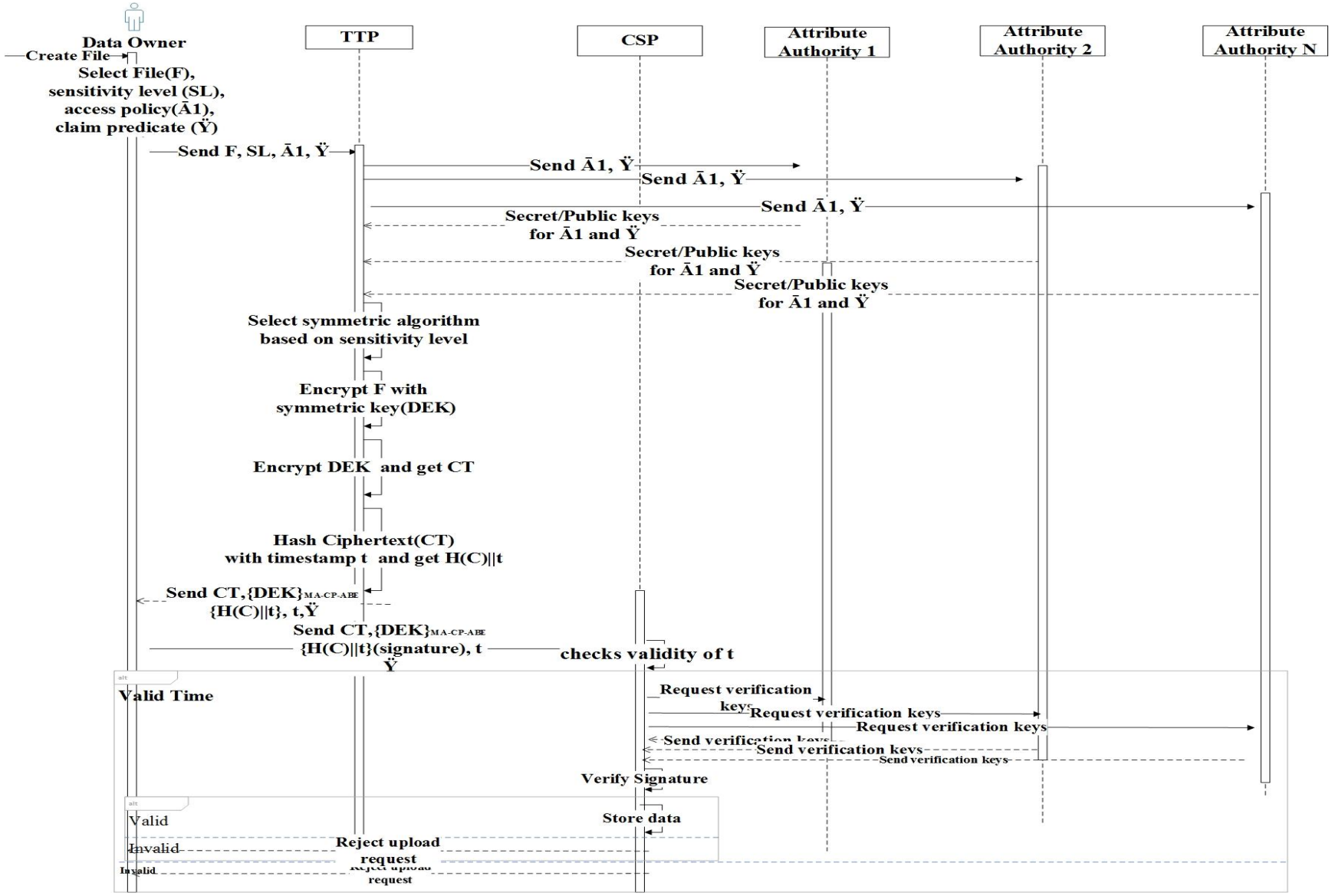

Figure 4-5: File Create

2. New User Grant

When a new user wants to join the system, he goes through the following steps:

Step 1: the data owner defines the role of user and determines if he is a reader or writer and sends this information to attribute authorities (AAs).

Step 2: The user send his certificate to AAs to get his designated keys

Step 3: Each AA validates the signature to check if the user is a legal user or not.

Step 4: If the user is a legal user, then each AA will assign him an attribute set S that is related to his identity/role in its administration domain. Otherwise, it aborts.

Step 5: Each AA runs the **SKeyGen** algorithm to generate all secret key components for the user. If the user is a reader, he will only receive secret key components to decrypt the ciphertext. If he is a writer, he will receive secret key components to decrypt the data. In addition to, secret key components to sign the data.

Step 6: After the user receives the key, he is able to either read or write to data files stored at a cloud service provider.

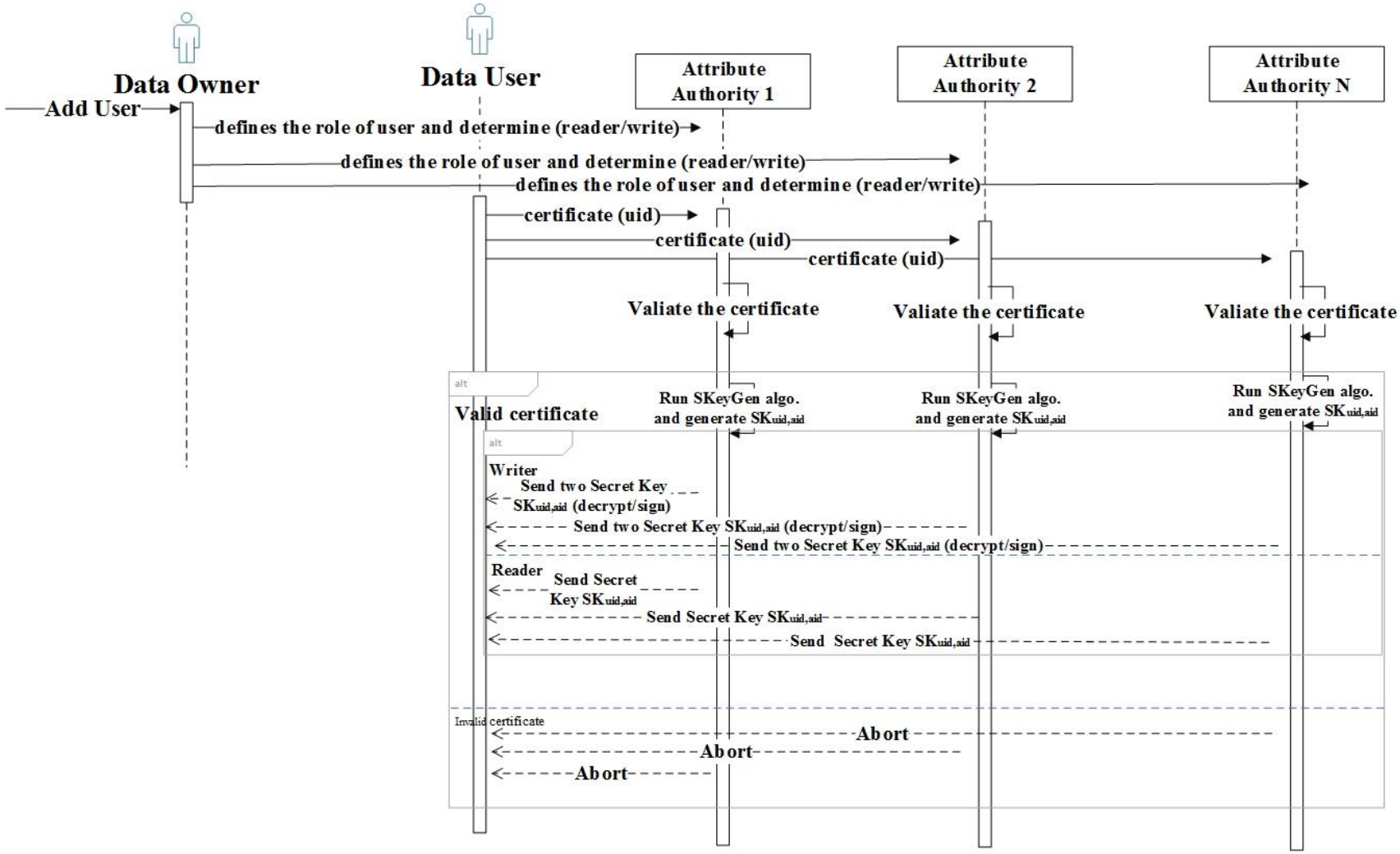

Figure 4-6: New User Grant

3. File Access(read/write)

Whenever a user wants to read the file, he processes as follows:

Step 1: The reader request the file from the CSP

Step 2: the cloud sends the file **{CT,**$\{DEK\}_{\{Pk_{aid}\}aid\in I_A}$**, δ ,** t **,** $\ddot{Y}$ **}** to the reader

Step 3: the user sends **{CT,**$\{DE\}_{\{Pk_{aid}\}aid\in I_A}$**, δ ,** t **,** $\ddot{Y}$ **}** to TTP

Step4: TTP request corresponding public keys from AA to verify signature (**δ)**

R1→**Verify (GPP, h(CT)|| t, δ,** $\bar{A}$ **,** $\{Pk_{aid}\}_{aid\in I_A}$**))**

Step 4: If the signature is valid, the TTP uses user's secret keys ($\{SK_{uid,aid_k}\}_{aid_k\in I_A}$) to decrypt attribute based encrypted decryption key$\{DEK\}_{MA-CP-ABE}$ and get symmetric decryption keyDEK. Otherwise, abort.

Step 5: The TTP decrypts encrypted file CT using symmetric decryption key DEK to obtain plaintext F.

Step 6: TTP send plaintext F to reader

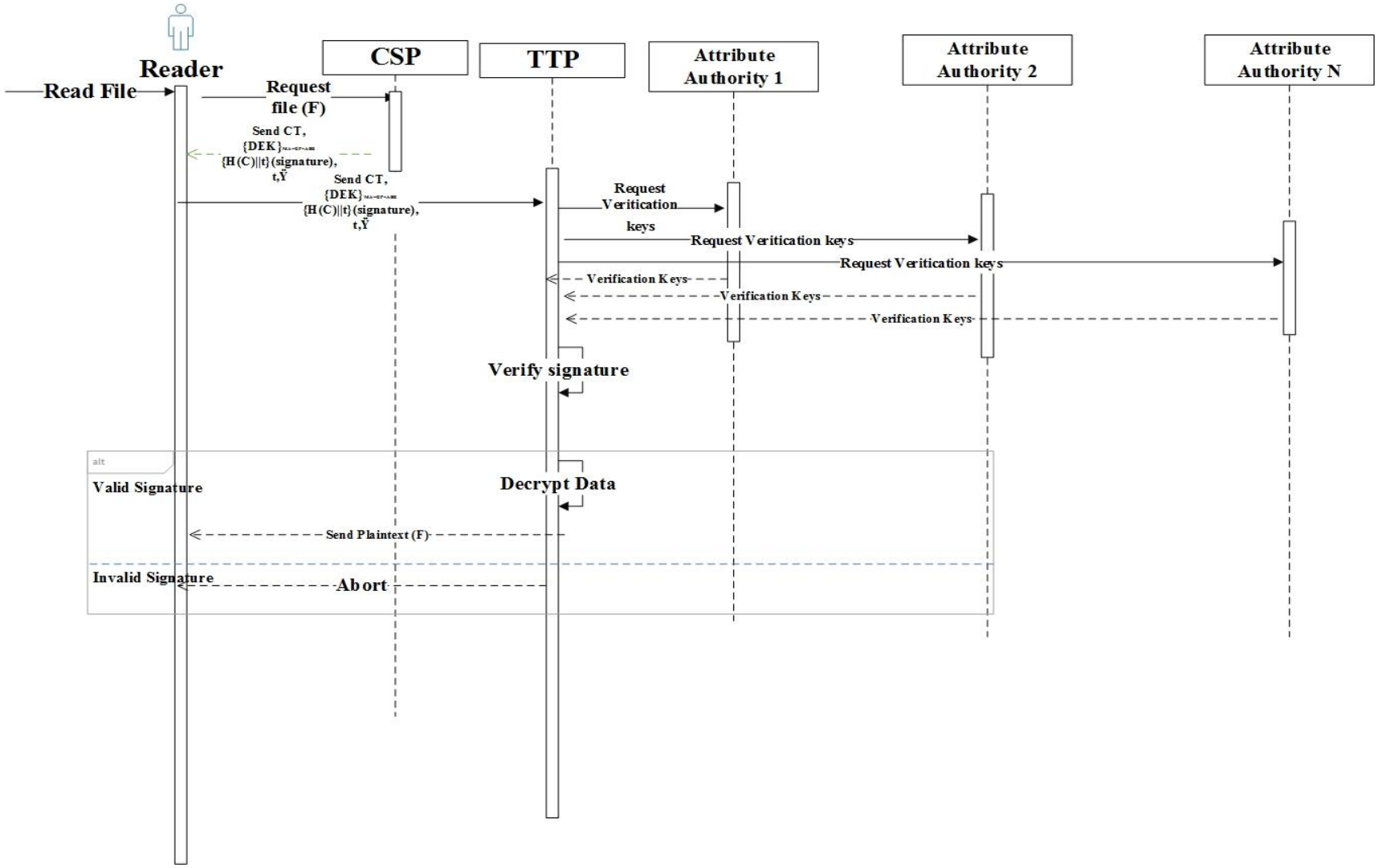

Figure 4-7: Read File

Whenever a user wants to update a file, he processes as follows:

Step 1: download the file as a reader (Same steps as stated above) to get plaintext.

Step 2: The user encrypts the plaintext F as in the encryption phase producing CT1

Step 3: Then the user sign the CT1 as in sign phase producing new signature **δ**1 with a new timestamp

Step 4: upload the updated encrypted file CT1, the attribute based encrypted decryption key$\{DEK\}_{MA-CP-ABE}$, the new signature **δ**1 with a new timestamp **t1**and claim predicate T_{sign} to the cloud storage provider (CSP).

Step 5: The cloud storage provider (CSP) will first check the validity of t1, then verify the **δ**1 by the **Ÿ** and verification keys to check if the user is able to update the file according to his secret keys or not.

Step 6: If the user is valid user, the updated file will be stored on the cloud otherwise the CSP will reject the update request.

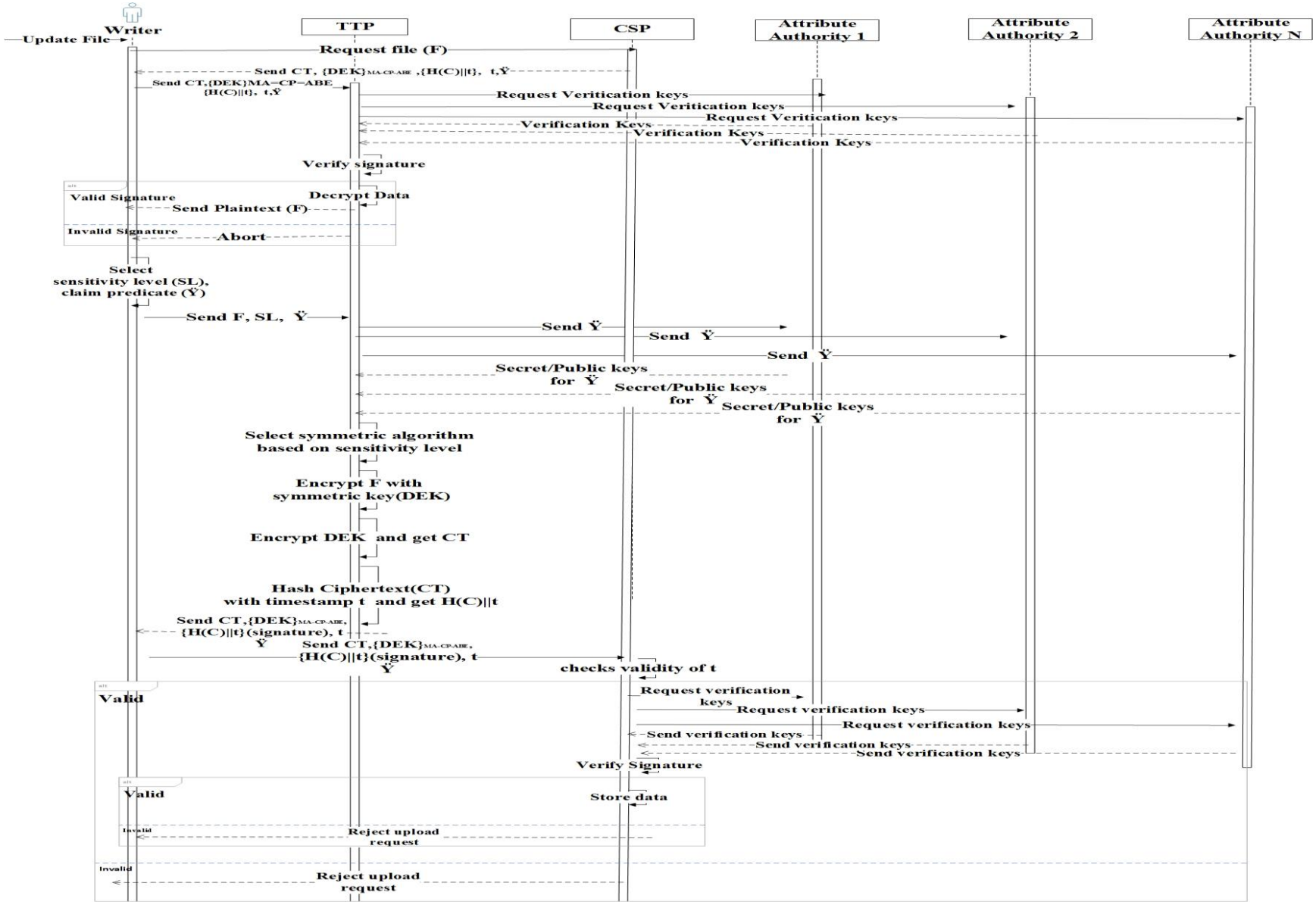

Figure 4-8: Write to file

- **Analysis**

The proposed scheme satisfies the security requirement in section 4.3. Fine-grain access control is achieved by employing multi-authority ciphertext policy attribute-based encryption (MA CP-ABE). MA CP-ABE defines each user access structure as a logic formula over data file attributes that represents any desired data file set. Instead of defining permissions based on roles as role based access control [118] or attach a list of privileged users to each file as in access control lists (ACL)[117], user files are described in terms of attributes in attribute-based encryption (ABE). Attributes are any bit of data, or label that describes a user, resource, target, object, environment, or action. In ABE, the file is encrypted once under the access policy and decrypted many by different users, each carrying a decryption key corresponding to access privileges (Appendix A). In this work, we have used a variant of attribute-based encryption (ABE) which is ciphertext policy attribute-based encryption (CP-ABE). In CP-ABE enables data owners to define their own access policies over user attributes and enforce the policies on the data to be distributed. In CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution. The authority can be the registration office in a university, the human resource department in a company, etc. The data owner defines the access policies and encrypts data under these policies. Each user will be issued a secret key according to his attributes. A user can decrypt the ciphertext only when his attributes satisfy the access policies. Moreover, in CP-ABE schemes, the access policy checking is implicitly conducted inside the cryptography. That is, there is no one to explicitly evaluate the policies and make decisions on whether allows the user to access the data [124], [125]. Therefore,

the data owners does not need to deal with any keys for data sharing. Despite the advantages associated with CP-ABE, one of the problems associated with CP-ABE is that the access structure, representing the security policy associated with the encrypted data, is not protected. Therefore, a curious storage provider might get information on the data by accessing the attributes expressed in the CP-ABE policies. The problem of having the access structure expressed in clear text affects in general all the CP-ABE constructions. Therefore, this mechanism is not suited for protecting the confidentiality of the access policies in an outsourced environment [206]. We solve this problem in our work [18] by implicitly associating the access structure inside the ciphertext. Furthermore, most of the existing CP-ABE schemes [26,114,127,129,131,132] have only one authority, which is responsible for issuing the secret keys for all users in the whole system. However, the applications in real world often require a user to obtain some attributes from different authorities (e.g., different government agencies, different commercial services he has subscribed to, different social networks he is registered with and so on) making it impractical to depend on a single authority. Moreover, single CP-ABE authority suffers from two drawbacks. The first one is that once the authority has been compromised, all secret keys are revealed. The second one is that it is difficult to find an authority trusted by all parties in large-scale network environment. Therefore, in this thesis, we used multi-authority CP-ABE. Several multi-authority attribute-based access control schemes [17,134-140] have been proposed. However, some of these [123,134,135] rely on a central authority (CA) to tie user's secret that are issued by different authorities for decrypting all ciphertexts. As a result, the CA has full control over user's data and is considered a

central point of failure. Another variant of MA CP-ABE, that removes the global authority is found in[139][140].Although[139], requires no global authority, it cannot scale well because authorities communicates with each other for creating the user's secret key leading to collusion attacks if the number of authorities is less than two. Therefore, our work is based on [18], because it is scalable, requires no global authority and avoids collusion attacks (details in system design section).

In addition, we delegate most of the computation load either to the cloud (as in revocation and write access enforcement) or to the trusted authorities (as it manages and distributes keys to users for data sharing). Furthermore, we provide a flexible many-write-many-read by combining and exploiting the CP-ABE and ABS. This many-write-many-read is designed to resist the replay attack by attaching a timestamp to the signature in order to prevent malicious reader from impersonating a valid writer by uploading an old version encrypted file with its old signature [205]. In addition, the insurance of data integrity is carried out by the cloud server whenever a user requests to update a file. This operation does not place any load on the owner because the ABS delegates the verification to the storage server. Moreover, users' data will not be leaked because it is stored in encrypted format no plain text is stored in the cloud.

- **Revocation**

Users may join and leave the system frequently, leading to constant key re-generation and re-distribution through additional communication sessions to handle user revocation. In a highly scalable system composed of thousands of users as the cloud, such events may occur at relatively high frequency. Therefore, revocation is

considered a challenging issue in this many one-to-many communication system with the data encrypted once and decrypted many. In MA CP-ABE, this issue is even more difficult since each attribute is conceivably shared by multiple users and the attributes comes from different authorities. Revocation in ABE systems come into two flavors: user revocation, and attribute revocation. User revocation takes place when the user's attributes donor satisfy the access structure. On the other hand, attribute revocation takes place when one attribute is revoked from user attributes. The removal of one or more attribute from the user does not mean that the user cannot decrypt ciphertexts. The user can still decrypt ciphertexts even after the removal of a subset of his/her attributes as long as the remaining attributes satisfy the access policy.

In this thesis, we propose a revocation approach for the multi-authority CP-ABE scheme. We want to support revocation process that removes the access from user either partially or totally without depending on the data owner for issuing new keys to other users or re-encrypting existing ciphertexts. Basing on the work in [19, 204, 18], we realize efficiently immediate attribute-level along with user revocation while achieving forward and backward secrecy. In our scheme, whenever an attribute revocation takes place, the corresponding attribute authority that possess this attribute generates a new version key for this revoked attribute and generates two update keys. One for non-revoked users whiles the other for updating ciphertext. By using their update key, the non-revoked users only updates the component associated with the revoked attribute in the secret key, while other components are kept unchanged. In this way, the used scheme can greatly improve the efficiency of attribute revocation. In addition, by this key updating process, the non-revoked users who hold the

revoked attributes update their secret key while revoked user do not. Hence forward security is achieved. On other hand, we make use of the abundant resources of the cloud to update the components associated with the revoked attribute in the ciphertext. We can do this by delegating the workload of updating attribute in the ciphertext to the server by using the proxy re-encryption. The goal of the proxy re-encryption is to securely enable the re-encryption of ciphertexts from one secret key to another key without worrying about illegal users who can see the data. By delegation the ciphertext update to proxy server, newly joined users are able to decrypt the data stored in the cloud before their participation, which was encrypted with the previous public keys, if they have attributes satisfying the access policy (backward security). Furthermore, by updating the ciphertexts, user do not need to keep records on all the previous secret keys because they hold the latest secret, unlike traditional methods that requires the user to keep record of re-generated keys distributed by the owner. In addition, delegation of ciphertext update to proxy eliminates the huge communication overhead between data owners and cloud server, and the heavy computation cost on data owners.

Whenever an attribute revocation take place: Update Key Generation by AAs, Secret Key Update by Non-revoked Users (those users who possess the revoked attributes ẍ$_{aid}$, but have not been revoked because their remaining attributes still satisfy the access structure) and Ciphertext Update by Server

Step 1: The AA that possesses the revoked attribute will run **UkeyGen** algorithm to produce two update keys.

UkeyGen($SK_{aid'}$, $VK_{\ddot{x}_{aid'}}$) → ($VK''_{\ddot{x}_{aid'}}$) ,$UK_{S,\ddot{x}_{aid'}}$, $UK_{c,\ddot{x}_{aid'}}$)

One for non-revoked uses to update their secret keys, while the other is send to proxy server to re-encrypt all ciphertexts that contain this revoked attribute.

Step 2: The non-revoked users run the SKUpdate algorithm to update his secret key to be able to decrypt the data files later after the revocation.

SKUpdate ($SK_{uid,aid'}$,$UK_{S,\ddot{x}_{aid'}}$) →$SK'_{uid,aid'}$

Step 3: The CSP will run CTUpdate algorithm to re-encrypt all ciphertexts that contain the revoked attribute.

CTUpdate(CT,$UK_{c,\ddot{x}_{aid'}}$) →CT"

User revocation is set of attribute revocation calls, the number of calls depends on the minimum number of attributes that make the user unable to satisfy the access structure. Whenever the data owner wants to revoke a user, the attribute authority determines the minimal subset of attributes γ'without which he cannot access the data. Then, for each attribute in the attribute setγ', the attribute authority performs an attribute revocation. Although, this method may incur huge computation overhead, however, the use of proxy servers in the cloud for updating ciphertext decreases much from this overhead.

Whenever a user revocation take place:

Step 1: The AAs will rum Aminimalset algorithm to get the minimum set of attributes that will make the user unable to decrypt the ciphertext.

$$\textbf{AMinimalSet(T)} \rightarrow \gamma'$$

Step 2: For each attribute in the minimum attribute set, we perform an attribute revocation process.

However, it is important to note that, whenever an attribute is removed from a user, all the ciphertext associated with this attribute shall be updated. Therefore, all re-encrypted data shall be signed with a new signature, since we do not trust the server for the signing process. The simple solution is to download all affected files and sign. However, this solution will incur high overhead on user. Therefore, we leave this problem as a future work and assume whenever a re-encryption takes place the signature is validated.

- **Analysis**

The proposed scheme satisfies the security requirement stated above in section 4.3.To achieve forward security, in the secret key update phase, the corresponding AA generates an update key for each non-revoked user that is associated with the user's global identity uid. Since each non-revoked user updated key include his global identity uid which is unique for each user, the revoked user cannot use update keys of

other non-revoked users to update its own secret key, even if it can compromise some non-revoked users. In addition, even if the revoked user was able to corrupt some other AAs (not the AA corresponding to the revoked attributes), the item in his secret key that distinguishes attributes from different AAs can prevent users from updating their secret keys with update keys of other users, since this item is only known by the authority and kept secret to all the users. On the other hand, the proposed scheme in [18] solves the collusion problem between users and attribute authorities and users and user. If a number of users collude together by combining their attributes to decrypt the ciphertext, they are not able to decrypt the ciphertext alone. Due to the random number t and the global identity of each authority id (aid) that is embedded in their secret key which makes each component associated with the attribute in the secret key is distinguishable from each other, although some AAs may issue the same attributes. Furthermore, each user secret key is also associated with his globally unique identity uid. Thus, users cannot collude together to gain illegal access by combining their attributes together.

4.4.3 Comparison with State-of-the-art Solutions

- **Data Confidentiality**

For data at rest, current cloud storage services provide their users with two solutions: server side encryption and client side encryption. For server side encryption, the data owner relies on the service for securing its data; however, this solution is not feasible for two reasons. The first reason is that the user will send his sensitive data in plaintext which exposes it to internal attacks where the attacker can exploit vulnerabilities of servers to achieve user's data. While for the second reason,

there is no guarantee that the service will encrypt the data before storing it in the cloud. On the other hand, in client side encryption, the data owner encrypts his data locally on his machine by using either client side cloud storage service as wuala [102] or encryption software such as TrueCrypt or BoxCryptor[103][201]. Although, these solutions seem to solve the problem, they do not address all aspects of the problem. In encryption software such as TrueCrypt, user's data is encrypted using full-disk encryption (FDE) methodology which encrypts virtual hard disks that can be mounted into the user's local file system and synchronized with. FDE is effective in protecting private data in certain scenarios such as stolen laptops and backup tapes; the concern is that it cannot fulfill data protection goals in the cloud, where physical theft is not the main threat. In addition, the encrypted data by itself is not feasible for sharing among users, these tools will not help much unless Encryption keys are shared among participating users over a secure out-of-band channel (or an elaborate PKI system is deployed). This is clearly a drawback in terms of usability [202]. Furthermore, these solutions have the drawback that the encryption and decryption process relies on software-based keys, which are stored on the respective client device and under some conditions could be accessible by unauthorized parties [103][201]. On the other hand, cloud storage services that provide client side encryption are not in a better situation, because the client software of these services may be exposed to the following threats: a) key disclosure: the client software uses the decryption key stored on user machine to decrypt the encrypted data send from the cloud storage provider to obtain the clear text. The client software might send this key to the provider or some other unauthorized party's; b) Manipulated file content: since these cloud services support

public key cryptography, the public keys of the users are known by some parties, including the provider. Server software may encrypt a malicious content making use of user's public key. The user can decrypt this content without detecting the fraud. This usually takes place because the data is usually not signed; c) the most dangerous threat that this solution exposed to is a secret agent working at the provider. This agent may be able to manipulate the client software. This agent can be used to inject malware in the customer's system [104,54]

Even the most secure client side storage service, Wuala, which supports convergent encryption [213] for securing files and optimizing storage by using de-duplication, suffers from confirmation of a file attack. In this attack, the attacker can effectively confirm whether a target possesses a certain file by encrypting an unencrypted, or plain-text, version and then simply comparing the output with files possessed by the target exposing user's data to dangers [203]. Therefore, most security experts advise cloud users who want to store their data in the cloud without any leakage to encrypt the data locally before uploading it to the cloud. Although, this method ensures that the data and the keys will not be leaked, it will not be feasible as it will incur too much burden to the client in terms of key management and maintenance, especially, if the user stores huge amount of data in the cloud.

From the previous, we note that the current solutions offered by cloud storage services for ensuring data confidentiality are not sufficient. We address the state of the art problem by transferring the trust to a trusted third party service (TTP). This service acts as middleware between the cloud and the user where it offer a level of trust between user and cloud. Although, it is not an optimal solution, but it tries to

achieve sufficient level of security while it maintains performance. This service can be managed locally on users' machine for those who do not trust anyone for their data and remotely for others. In addition, the user can encrypt each file separately taking an advantage over encryption software that encrypts the full disk. The user can also store his data keys on hardware devices that make it to great extent inaccessible to attacks taking advantage over client side storage service. Moreover, the service offers better performance by providing different levels of encryptions for data uploaded to the cloud. It also reduces the load for managing keys if the users encrypt the data by themselves. Furthermore, it can be used by devices with limited capabilities as mobiles. These devices can make use of the two varieties of service either by employing it or using it remotely.

- **Fine Grain Access Control**

Existing cloud storage services only provide basic access control mechanisms, and the limited research on secure, shared cloud repositories often require extensive deployment of infrastructure services that undermines their manageability. To prevent the un-trusted servers from accessing sensitive data, traditional methods usually rely on the data owners for encrypting files by using the symmetric encryption approach with content keys and then use every user's public key to encrypt the content keys and only users holding valid keys can access the data. These methods require complicated key management schemes and the data owners have to stay online all the time to deliver the keys to new user in the system. Moreover, these methods incur high storage overhead on the server, because the server should store multiple encrypted copies of the same data for users with different keys. This methodology

cannot be used with cloud storage sharing services that separates the roles of the data owner from the data service provider. In addition, it did not provide direct interaction between the data owner and the user directly for providing data access service [128,207-209]. Another prevalent methodology for enforcing access control policy is to provide the remote cloud server the power of key management and distribution under the assumption that the server is trusted or semi-trusted. However, the server cannot be trusted by the data owners in cloud storage systems and thus these methods cannot be applied to access control for cloud storage systems [210-211]

From the previous we note that, our proposed scheme provides a better way for ensuring the data confidentiality against service provider since cloud storage services have no access to plain text. Moreover, cloud storage services have no knowledge about user access privilege information, it only performs computational tasks. In addition, users (data owner/data users) are able to share data without direct interaction and without managing any data or control access keys. Furthermore, our approach allows only authorized users to have access to the shared files. In addition, our approach provides a flexible many-write-many-read method for data sharing where owners neither need to be always online nor need to distribute any credentials to other users individually. Compared with the current commercial cloud-based file sharing services, our solution provides a novel fine-grained access control mechanism to the file sharing services which enables real-life development of secure service. Since, the current cloud-based file sharing services offered by the cloud storage services did not provide a good means for securing data or ensuring data confidentiality against accesses beyond authorized rights. Most of these cloud storage services did not

include the file sharing feature [211,212]. However, those who support file sharing are able to view either the plaintexts of data files or user access structure or both. Even the most secure ones as wuala [102] still suffer from data leakage. In these secure cloud-based file sharing service (e.g. wualaa) the inviting user ask the service for the public key of the invitee. Then the inviting encrypts the decryption key by the invitee public key and sends this cryptogram to the provider which in turn sends the cryptogram to the invitee. However, this idea seems to be good, there is a problem associated with solution. The problem is that the inviting user can not verify the authenticity of invitee's key, because there is no independent public key infrastructure. So in the worst case, the inviting user encrypts the decryption key for another person that can decrypt the inviting user file [54].

- **Revocation**

The state-of-the-art MA CP-ABE schemes provide limited support for key revocation. Several multi-authority attribute-based access control schemes [134,139,140] lack revocation approach. On the other hand, others support revocation schemes that lack efficiency as in [137].In [137], the revocation can take place by any user with re-encryption privilege to re-define the access policy. He then recovers the plaintext message before re-encrypting. This method affects the confidentiality of the data and is not suitable and efficient in the cloud. Although, our proposed methodology is not the most efficient one to resolve the revocation problem, we try to propose a system that maintains the tradeoff between performance and security. In our scheme, we combine both user revocations along with attribute revocation

without leaking any information to server about revoked users and decreasing the computation cost, unlike [145,146] that delegate key updating to server which may not be appropriate for protecting the users' privacy and the data security because the cloud maintain user revocation lists containing information about revoked users. Our method only makes use of cloud for ciphertext update. However, our attribute as user revocation put some burden on attribute authorities for computing new update key for each unrevoked user even if the attribute is revoked from only one user. Our solution tries to maintain the tradeoff between performance and security.

4.5 Detailed Description of the Proposed Architecture Algorithms

Our framework for securing data in cloud storage while maintaining access control scheme is a collection of algorithms. These algorithms combine a set of multi-authority CP-ABE algorithms along with ABS algorithms. These algorithms are: CASetup, AASetup, AARegistration, UserRegister, KeyGen, Encrypt, Decrypt, Sign, and Verify.

General Overview

i. System Initialization

The system initialization phase contains Certificate Authority setup and Attribute Authorities setup

a) Certificate Authority Setup

CASetup (λ) → (GMSK, GPP, {(GPK_{uid},GPK'_{uid}), (GSK_{uid},GSK'_{uid}), SIG_{sk}(uid)})

The CA setup algorithm is run by the CA. It takes no input other than the implicit security parameter λ. It generates the global master key GMSK of the system and the global public parameters GPP. For each user uid, it generates the user's global

public keys (GPK_{uid},GPK'_{uid}), the user's global secret keys (GSK_{uid},GSK'_{uid}) and a certificate SIG_{sk}(uid) of the user.

b) Attribute Authorities Setup

AASetup(U_{aid})→(SK_{aid},PK_{aid},{{$VK_{x_{aid}}$, $PK_{x_{aid}}$}$_{x_{aid}\in U_{aid}}$)

The attribute authority setup algorithm is run by each attribute authority. It takes the attribute universe U_{aid}managed by the AA_{aid}as input. It outputs a secret and public key pair (SK_{aid},PK_{aid},) of the $AA_{aid}$and a set of version keys and public attribute keys $\{VK_{x_{aid}}, PK_{x_{aid}}\}_{x_{aid}\in U_{aid}}$ for all the attributes managed by the AA_{aid} .

ii. Secret Key Generation

SKeyGen(GPP,GPK_{uid}, GPK'_{uid},GSK_{uid} , SK_{aid} ,$S_{uid,aid}$,{$VK_{x_{aid}}$, Pk_{xaid}}$_{X_{aid}\in S_{aid}}$) →$SK_{uid,aid}$

The secret key generation algorithm is run by each AA. It takes as inputs the global public parameters GPP, the global public keys (GPK_{uid},GPK'_{uid}) and one global secret key $GSK_{uid}$of the user uid, the secret key of the attribute authority (SK_{aid}) , a set of attributes $S_{uid,aid}$ that describes the user uid in that attribute authority AA_{aid} and their corresponding version keys and public attribute keys $\{VK_{x_{aid}}, Pk_{xaid}\}$. It outputs a secret key $SK_{uid,aid}$for the user uid which is used for decryption.

iii. Data Encryption and Signature by Owners

Encrypt(GPP,{Pk_{aid}}$_{aid\in I_A}$, DEK, Ā)→CT.

The encryption algorithm is run by the third party software service to encrypt the content keys. It takes as inputs the global public parameters GPP, a set of public keys $\{Pk_{aid}\}_{aid\in I_A}\}$ for all the AAs in the encryption set I_A, the content key DEK and an access policy Ā. The algorithm encrypts DEK according to the access policy and outputs

a ciphertext CT. We will assume that the ciphertext implicitly contains the access policy A.

Sign(GPP,H(C)|| t , Ÿ ,$\{Pk_{aid}\}_{aid\in I_A}$},($GPK_{uid}$,GSK'_{uid}),$SK_{uid,aid}$)→δ

The signing algorithm is run by third party software service. It takes as inputs the global public parameters GPP, message M(hashed CT) , claim predicate **Ÿ**, set of public keys $\{Pk_{aid}\}_{aid\in I_A}\}$ for all the AAs in the signing set I_A, and all corresponding user keys as input. Then it returns signature δ if user's attribute set satisfies claimed access structure. Otherwise, it returns null.

iv. Data Decryption and Verification by Users

All the legal users in the system can freely query any interested encrypted data. Upon receiving the data from the server, the user runs the verify algorithm to verify the data, if the verifications succeed. The user runs the Decrypt algorithm to decrypt the ciphertext by using its secret keys from different AAs. Only if the attributes the user possesses satisfy the access structure defined in the ciphertext CT, the user can get the content key.

Verify (GPP, M, δ, Ÿ , $\{Pk_{aid}\}_{aid\in I_A}$})→{valid, invalid}

The verification algorithm is run by any user/server who wants to verify whether attribute set of user satisfies claimed access structure Ā. It takes as inputs the global public parameters GPP, message M , claim predicate, signature δ , and all public keys $\{Pk_{aid}\}_{aid\in I_A}\}$ as input, then returns one bit to tell whether the signature is valid or not.

Decrypt (CT,GPK_{uid} , GSK'_{uid},$\{SK_{uid,aid_k}\}_{aid_k\in I_A}$) →DEK.

The decryption algorithm is run by users to decrypt the ciphertext. It takes as inputs the ciphertext CT which contains an access policy A, a global public key GPK_{uid} and a global secret key GSK'_{uid} of the user uid, and a set of secret keys $\{SK_{uid,aid_k}\}_{aid_k \in I_A}$ from all the involved AAs. If the attributes $\{SK_{uid,aid_k}\}_{aid_k \in I_A}$ of the user uid satisfy the access policy A, the algorithm will decrypt the ciphertext and return the content key DEK

v. User/Attribute Revocation

The revocation comes in two forms: user revocation and attribute revocation. Attribute revocation consists of three steps: Update Key Generation by AAs, Secret Key Update by Non-revoked Users(those users who possess the revoked attributes ẍ$_{aid'}$ but have not been revoked because their remaining attributes still satisfy the access structure) and Ciphertext Update by Server. On the other hand, user revocation has the same three steps in attribute revocation, in addition to, minimalset which determining the minimal set of attributes with which the users cannot access the data.

$$\textbf{UkeyGen}(SK_{aid'}, \ddot{x}_{aid'}, VK_{\ddot{x}_{aid'}}) \rightarrow (VK''_{\ddot{x}_{aid'}}), UK_{S,\ddot{x}_{aid'}}, UK_{c,\ddot{x}_{aid'}})$$

The update key generation algorithm is run by the corresponding $AA_{aid'}$ that manages the revokedattribute ẍ$_{aid'}$. It takes as inputs the secret key $SK_{aid'}$ of $AA_{aid'}$, the revoked attribute ẍ$_{aid'}$ and its current version key $VK_{\ddot{x}_{aid'}}$. It outputs a new version key $VK''_{\ddot{x}_{aid'}}$ and the update key $UK_{S,\ddot{x}_{aid'}}$ (for secret key update) and the update key $UK_{c,\ddot{x}_{aid'}}$.(for ciphertext update).

$$\textbf{SKUpdate}\ (SK_{uid,aid'}, UK_{S,\ddot{x}_{aid'}}) \rightarrow SK'_{uid,aid'}$$

The secret key update algorithm is run by each non-revoked user uid. It takes as inputs the current secret key of the non-revoked user $SK_{uid,aid'}$ and the update key $UK_{S,\ddot{x}_{aid'}}$. It outputs a new secret key $SK'_{uid,aid'}$ for each non-revoked user uid.

CTUpdate(CT,$UK_{c,\ddot{x}_{aid'}}$) →CT"

The ciphertext update algorithm is run by the cloud proxy server. It takes as inputs the ciphertexts which contain the revoked attribute $\ddot{x}_{aid'}$, and the update key $UK_{c,\ddot{x}_{aid'}}$. It outputs new ciphertexts CT" which contain the latest version of the revoked attribute $\ddot{x}_{aid'}$.

AMinimalSet(T) →γ'

The AMinimalSet is run by AAs. It takes as input an access tree T. It finds a minimal subset of attributes without which T will never be satisfied. It outputs the minimum set of attributes that restrict user access to the data fileγ'.

Details of the construction

I. System Initialization

a) Certificate Authority Setup

Let S_A and S_u denote the set of attribute authorities and the set of users in the system respectively. Let G and G_T be the multiplicative groups with the same prime order p and e: G×G→$G_T$be the bilinear map. Let g be the generator of G. Let H {0, 1}∗→G be a hash function that matches the string to an element in G, such that the security will be modeled in the random oracle.

The certificate authority (CA) sets up the system by running the CA setup algorithm (CASetup), which takes a security parameter λ as input. The CA first chooses two multiplicative groups G and GT with the same prime order p and a bilinear map e such that e:G×G→GT. It also chooses a hash function H: {0,1}∗→G that matches the string to an element in G. Then, the CA chooses two random numbers a, b∈Z_P as the global master key GMSK=(a,b) of the system and computes the global public parameters as GPP = (g, g_a, g_b, ,H).

After creation of the system public/secret keys, the CA is ready to accept both User Registration and AA Registration.

1) User Registration: When a user joins the system, the CA first authenticates this user. If the user is an authorized user in the system, the CA will assign him a globally unique user identifier uid .After that, the CA generates two random numbers u_{uid}, u'_{uid}∈Z_P in order to create the user's global secret keys as (GSK_{uid},GSK'_{uid}). It then generates the user's global public keys as (GPK_{uid},GPK'_{uid}) for each user with unique identifier (uid). In addition, the CA generates a certificate SIG_{sk}(uid) for each user. Then, it sends one of the user's global public keys GPK_{uid} , one global secret key GSK'_{uid} and the Certificate SIG_{sk}(uid) to the user.

2) AA Registration: Each AA should register itself to the CA during the system initialization. If the AA is a legal authority in the system, the CA will assign it a global attribute authority identifier aid. After that, the CA sends the other global public/secret key of each user (GPK'_{uid} ,GSK_{uid}) to the AA_{aid}together with the

system global public parameter GPP. It also sends a verification key to theAA_{aid} which can be used to verify the certificates of users issued by the CA.

b) Attribute Authorities Setup

Let S_{aid} represent the set of attributes managed by each attribute authority AA_{aid} ($S_{aid} \in U_{aid}$). Each AA_k (k $\in S_A$) runs the authority setup algorithm AASetup. It chooses three random numbers $\alpha_{aid}, \beta_{aid}, \gamma_{aid} \in Z_P$as the authority secret keySk_{aid}.It also generates Pk_{aid} as its public key. In addition, for each attribute $X_{aid} \in S_{aid}$, the AA_{aid}generates a public attribute key $Pk_{xaid}$by implicitly choosing an attribute version keyVk_{xaid}. All the public attribute keys $\{Pk_{xaid}\}_{X_{aid} \in S_{aid}}$ along with the public key $Pk_{aid}$of the $AA_{aid}$are published on the public bulletin board of the AA_{aid} .

II. Secret Key Generation

For each user U_{uid} (uid$\in S_u$) and each authority AA_{aid}(aid$\in S_A$), the user has to authenticate himself to an AA_{aid} to prove that he is a legal user before he can be entitled some attributes from thatAA_{aid}. The user authenticate himself as a legal user by submitting his certificate SIG_{sk}(uid) to the AA_{aid}. The AA_{aid} then authenticates the user by using the verification key issued by the CA. If the user is a legal one, the AA_{aid} entitles a set of attributes $S_{uid,aid}$to the user uid according to its role or identity in its administration domain. Otherwise, it aborts. After that, the AA_{aid} generates the user's secret key $SK_{uid,aid}$

III. Data Encryption and Signature by Owners

Before uploading the data (m) to cloud servers. Data owner first encrypts the data (m) with content keys (DEK) by using symmetric encryption algorithm, then he runs the encryption algorithm Encrypt to encrypt the content key. It takes as inputs the as inputs the global public parameters GPP, a set of public keys $\{Pk_{aid}\}_{aid\in I_A}\}$ for all the AAs in the encryption set I_A, the content key DEK and the access policy (M,ρ) over all the involved attributes.

Let M be a ∫ × n matrix, where ∫ denotes the total number of all the attributes. The function ρ maps each row of M to an attribute. To encrypt the content key κ, the encryption algorithm first chooses a random encryption exponent s∈$Z_P$and chooses a random vector v = (s, y_2,....,y_n) ∈$Z_P{}^n$, where y_2,....,y_n are used to share the encryption exponent s. For i = 1 to ∫, it computes λ_i= v · M_i, where M_i is the vector corresponding to the i-th row of M. Then, it randomly chooses r_1,....,$r_{\int}$∈Z_P and computes the ciphertext CT.

It is important to note that, the encryption set I_A consists of a set of AAs from the S_A that are involved in the encryption, because not all the attributes of the access structure come from all AAs. In addition, we assume that the ciphertext implicitly contains the access policy in order prevent malicious CSP from deducing any confidential information about the data if the access policy is attached to the ciphertext.

IV. User/Attribute Revocation

Attribute Revocation

Let ẍ$_{aid'}$denotes the revoked attribute and uid' denotes the revoked user. Suppose an attribute ẍ$_{aid'}$is revoked from the user uid', the revocation method performs the following three steps:

1) Update Key Generation

In this step, the $AA_{aid'}$ generates new version key for revoked attribute, update key for non-revoked users and an update key of ciphertext update to enable the users access the file after and the revocation. By theses update keys the revoked user is not able to access the file because his keys are obsolete and are cannot to decrypt the ciphertext. On the other hand, non-revoked users can access the file after updating their secret keys.

When an attribute ẍ$_{aid'}$is revoked from the useruid', the corresponding authority $AA_{aid'}$that governs this attribute runs the update key generation algorithm UKeyGen to compute the update keys. The algorithm takes as inputs the secret key $SK_{aid'}$of$AA_{aid'}$, the revoked attributeẍ$_{aid'}$. It then generates a unique update key $UK_{s,\ddot{x}_{aid'}}$for secret key update by each non-revoked user uid and generates the update key $UK_{c,\ddot{x}_{aid'}}$ for ciphertext update.

Next, the $AA_{aid'}$sends the $UK_{s,\ddot{x}_{aid'}}$ to non-revoked user uid and sends$UK_{c,\ddot{x}_{aid}}$. To the cloud proxy server.

Then, the $AA_{aid'}$updates the public attribute key of the revoked attribute ẍ$_{aid'}$ and publishes it on its public bulletin board. Then, the $AA_{aid'}$broadcasts a message for all the owners that the public attribute key of the revoked attribute ẍ$_{aid'}$is updated.

2) Secret Key Update by Non-revoked Users

In this step, the non-revoked user receives the update key $UK_{S,\ddot{x}_{aid',uid}}$ from the corresponding$AA_{aid'}$. After that, he updates his/her secret key by running the new secret key update algorithm SKUpdate. The algorithm takes the non-revoked user's current secret key $SK_{uid,aid'}$ and the update key $UK_{S,\ddot{x}_{aid'}}$ as its input and outputs a new secret key $SK'_{uid,aid'}$for the non-revoked user uid.

It is important to know that, this algorithm only updates the component associated with the revoked attribute $\ddot{x}_{aid'}$in the secret key, while other components are kept unchanged.

3) Ciphertext Update by Cloud Server

In this step, the cloud server receives the update key $UK_{c,\ddot{x}_{aid'}}$.from the corresponding$AA_{aid'}$. Next, it forwards it to the proxy server, which in turn, runs the ciphertext update algorithm CTUpdate to update the ciphertext associated with the revoked attribute $\ddot{x}_{aid'}$. The algorithm takes the current version of ciphertext (CT) that is associated with the revoked attribute $\ddot{x}_{aid'}$and the update key $UK_{c,\ddot{x}_{aid'}}$ as its input and outputs a new ciphertexts CT" which contain the latest version of the revoked attribute $\ddot{x}_{aid'}$.

User Revocation

The user revocation is set of attribute revocation calls with the outputs minimum number of attributes that make the user unable to satisfy the access structure. Whenever the data owner wants to revoke a user, the $AA_{aid'}$ runs the AMinimalSet algorithm to get minimal subset of attributes γ'without which he cannot access the

data. This algorithm takes as input an access tree T and outputs minimal subset of attributes γ'without which T will never be satisfied. For each attribute in the attribute setγ', the AA_{aid}, perfroms an attribute revocation.

4.6 Security Analysis

To validate the conceptual design presented earlier, a security proof (analysis) has been constructed. Through this security proof (analysis), we shall investigate the possibility of attacks from unauthorized users and cloud service provider to gain access to the outsourced data that they are not allowed to access. The security proof investigates the proposed scheme on each step of its fundamental steps.

1. Data initialization and Key generation

In multi-authority CP-ABE, user keys come from different authorities. Therefore, user's secret keys must be tied together for the same user without exposing it to any collusion attacks. The collusion attacks in multi-authority environment appear in the following cases: users collude with each other or with attribute authorities. In this thesis, based on [18], these issues are resolved. This work is able to tie secret keys using by employing a certificate authority (CA), which is not involved in any creation of secret keys or management of attributes. The certificate authority (CA) is responsible only for issuing global keys and global unique identities to legal users and authorities along with global master key GPMK. Since each user has a unique global identifier uid and, secret keys issued by different AAs for the same uid can be tied together for decryption without the need for a central authority as in [134]. Therefore, a colluding user cannot combine his secret keys from a certain set of authorities with another user who has enough keys

from the other authorities to decrypt the ciphertext, because each key has its user identity (uid) embedded inside so different key from different users cannot make up user secret key. In addition, each user key contains a random number t for randomizing the key. Due to this random number t and the AA global identifier aid, each component associated with the attribute in the secret key is distinguishable from each other. Therefore, users and authorities cannot collude by combine their keys to get access to user's data, even if some AAs may issue the same attributes. Furthermore, the CA do not have full control over encrypted data, because its GPMK is a share of the key not the whole key as in [134].In[134], user's data are encrypted with system unique public key (generated by the unique master key) that is owned by central authority. Therefore, the central authority [134] has full control over encrypted data. However, CA it is considered a single point of failure and if it is corrupted, the whole system will be totally down. Therefore, we assume it is it fully trusted and we can use a backup for the CA, to avoid the single point of failure issue.

Moreover, each user is issued a certificate from the CA that it is presented to AA for requesting the secret keys. The AA validates this certificate using the verification keys issued from CA before issuing any keys to users. By doing his validation step, we prevent any user from using a fake uid to request a decryption keys from AAs.

In addition, this certificate prevents attribute authorities from colluding with each other, because, users do not present their unique identifiers to every authority for requesting the key. However, they just submit their certificate. This certificate is a pseudonym based on user unique identifier that proves to the attribute authority that he has this uid, without revealing the uid itself.

2. File creation

Initially, the trusted third party (TTP) service encrypts users data using a symmetric key selected by the user according to sensitivity of data either locally or remotely. The keys are stored on a hardware device which makes them it hard to for attackers to break.

3. File access

Our design offers two-layer encryption for data before outsourcing it to the cloud. The data is encrypted according to the level of sensitivity chosen using a symmetric key algorithm by TTP. Then, the encryption key is encrypted with MA- CP-ABE secret key. After that, the encrypted outsourced data$\{F\}_{DEK}$, the attribute based encrypted decryption key$\{DEK\}_{MAEK\}ttri}$, encrypted decryption key, signature and claim predicate $T_{sign}$are uploaded to the cloud. In order for the adversary to extract any information about F, he has to decrypt DEK firstly in order to extract any information about F. However, such session key (DEK) is encrypted with the access control policy (τ) it would further require MA-CP-ABE secret key (SK) that can satisfy (τ). Since, SK is only shared with the legitimate users by the data owner, the computational complexity for an attacker would be equal to deciphering CP-ABE without SK. Actually, MA-CP-ABE used in this thesis is provably secure under [125] given the decisional q-parallel Bilinear Diffie-Hellman Exponent (q-parallel BDHE) problem is hard. Therefore, the intuitive scheme is secure under the same model. Since the ciphertext implicitly contains the access policy (τ), the attackers cannot track the user or infer the sensitivity of ciphertext by eavesdropping the access policies. Furthermore, unauthorized users cannot update any file, because any user must be authenticated he to the cloud by providing the secret keys that satisfy its claim predicateT_{sign}. Since these users cannot present these credentials to the cloud, they are not allowed to update the file. Therefore the message integrity with non-repudiation can be

provided by our proposed scheme. Moreover, our proposed scheme is resistant to replay attacks, because, whenever a revoked writer or a reader tries to write data to file by uploading an old version encrypted file with an old signature which was signed by a former writer to the cloud storage server, they are not able to replace data with stale information from previous writes. This is because of the period of validity t (time stamp) associated with each file. So, any write operation has to attach a new time stamp τ and sign the message $H(C) \| \tau$ again. Since they do not have valid attributes, they are not able to create a valid signature.

4. User revocation

Our scheme can achieve forward security whenever attribute or user revocation takes place. Upon the revocation of an attribute, the attribute authorities generate update keys for non-revoked users to update their current key with the new version key for the revoked attribute. Since the updated key includes the user's global identity uid, only non-revoked users can update their keys. On the other hand, the revoked users cannot make use of these keys to update their secret key as it does not include their (uid) s. Moreover, the revoked user cannot collude with other attribute authorities to get the updated key, because each attribute authority has a secret random number γ_{aid} that is used for attribute revocation embedded in its secret key. Therefore, whenever, an attribute revocation take place, the AA generates update keys to non-revoked including this random number making colluding an authority with another an impossible task.

The proposed scheme achieves backward security by utilizing the attribute version key. After each attribute revocation process, the revoked attribute version key is updated. Any new user joining the system will be assigned secret keys associated with these

attributes with the latest version. By using proxy server, the previously published ciphertexts that was encrypted under attributes with old version with be re-encrypted into the latest attribute version. Therefore, any new user who joins the system can still decrypt previously published ciphertexts, if their attributes can satisfy access policies associated with ciphertexts. Consequently, backward security satisfied.

4.7 Discussions

We compare our scheme with the two other access control schemes (in the table below) that are similar to ours. We will show that our scheme supports many features that the other schemes did not support. The three schemes support fine grain access control and multi-read-multi writes. However, they differ in that [167] and ours support MA- CP-ABE, while [205] support single authority CP-ABE. Although [167] support decentralized CP-ABE, the decentralized CP-ABE algorithm that [167] is based on incurs a very significant loss in efficiency. This loss of efficiency is costly enough to limit the potential applications of fully secure system. In addition, it did not consider the collusion attack that takes place between authorities. Our scheme support revocation operation that both [205] [167] did not consider. Moreover, both of the two schemes depend on owner for managing data encryption and decryption, unlike our scheme that rely on TTP to remove this burden from data owner.

Scheme	Fine-grained access control	Type of authority	Write and read access	Type of access control	Privacy preserving authentication	Encryption	Revocation	Collusion Resistance
[205]	Yes	Single	Yes	CP-ABE	Yes	Owner	No	Partially
[167]	Yes	Multi	Yes	CP-	Yes	Owner	No	Partially

				ABE				
Ours	Yes	Multi	Yes	CP-ABE	Yes	TTP	Yes	Fully

In the following lines, we shall investigate and analyze the requirements traceability.

To achieve data confidentiality in transit in our design, we employed a SSL/TLS protocol in the communication channel between the cloud users and cloud service provider. Data confidentiality against cloud service provider is maintained by using a trusted third party service which encrypts the data before uploading it to the cloud and decrypt it upon user's request to access. Therefore, we are able to achieve data confidentiality at rest. In addition, the trusted third party stores user's encryption / decryption keys on hardware device to protect it for key snooping or stealing.

For data integrity, the trusted third party service signs user's data with private keys issued from attribute authorities according to a claim predict defined by the data owner. The cloud service provider checks data integrity before storing it in the cloud by requesting the public keys from designated attribute authority in each upload process. By doing so, the data owners or writers are the only ones that have the privilege to modify the data. In addition, whenever, the data is downloaded from the cloud, the trusted third party service requests the public keys from designated attribute authority to verify consistent of updated data and detect any unauthorized modification.

To achieve data confidentiality against accesses beyond authorized rights, trusted third party service encrypts user's data with public keys issued from attribute authorities according to the access policy defined by the data owner. The access policy is defined

at attribute level using multiple authority ciphertext policy attribute based encryption. Therefore, we achieve fine grained access control over encrypted data and allow authorized users only to have access to the data according their assigned attributes.

To achieve effective and efficient revocation, the AAs generate update keys to update the keys of non-revoked users and update keys to re-encrypt data associated with revoked attributes. We delegate the task of re-encrypting the data to proxy servers that reside on the cloud so that we can use the abundant resources of the cloud.

5 Conclusion and Future Work

Cloud storage services provide a cost effective solution to deal with the problem of on demand data accessibility. These services enable their subscribers to share, collaborate, archive and synchronize data across different devices and domain, without the concerns of data provisioning and availability. Cloud infrastructure associated with these services is owned, managed and operated by an un-trusted entity called cloud service provider (CSP). Since, CSP is in-charge of processing, persisting and provisioning of outsourced data there is a great deal of privacy concerns when confidential data is outsourced to such services.

To ensure data privacy and confidentiality often cryptographic methodologies are employed (i.e., encryption algorithms); however, these methodologies are not enough to achieve fine-grained access control. Access control policies ensure fine-grained access control. However, conventional methodologies were designed to restrain illegal data access in a trusted domain in which only user accessing data could behave maliciously. Contrary to that, cloud storage services were provisioned from public domain by an un-trusted entity. Thus, conventional access control policy could be exploited by a cloud service provider to compromise privacy of the outsourced data.

In this dissertation we address these problems by designing a secure file sharing service. The proposed service transfers the trust from the cloud to a trusted third party service. It also provides security for users' data with minimal overhead on cloud users. Particularly, this service ensures data confidentiality against cloud and unauthorized users. Data confidentiality against cloud can be achieved by storing data in an encrypted

format in cloud storage so that malicious insiders are not able to view/decrypt it. Since the currently deployed encryption services in either the cloud or inside client side cloud storage services are vulnerable to security attacks, we address these vulnerabilities by using a trusted third party (TTP) service. This TTP service has encryption/decryption service that can be employed either locally or remotely according to level of severity of the data. This service shall remove the burden of key management and maintained from data owners. Moreover, this service takes advantage over the current software encryption/decryption service that offers full disk encryption. For achieving data confidentiality against unauthorized users, the TTP service collaborates with a number of attribute authorities to achieve fine grained access control. By doing so, we prohibit the cloud and unauthorized users from getting access to owner's plaintext or credentials, unlike most of the currently available cloud storage services that either do not provide file sharing services or give the cloud provider full power over access control .In addition, we support user and attribute revocation without depending on the data owner for re-encrypting the affected files or regenerating system parameters and users' keys. Moreover, we provide read or write or both accesses to a file stored in the cloud, unlike most of the systems that supports 1- write-many-read. Last but not least, we shift most of the heavy computations such as verification and re-encryption from the owner/user to the cloud. We believe that our proposed scheme combine different algorithms to form a larger and more generic solution that supports the needs of a cloud-based collaboration environment.

We validate our design by security analysis. The analysis demonstrates the feasibility of the entire design as it is described above. Through this security proof (analysis), we

investigate the possibility of attacks from unauthorized users and cloud service provider to gain access to the outsourced data that they are not allowed to access.

Contribution

1) We defined a secure storage system that utilizes a trusted third party service that enables users to share data over any web-based cloud storage platform while data security is preserved. This service protects the confidentiality of the communicated data and it can be employed locally or remotely. We take advantage over almost all existing work that depends on data owner, client side cloud storage services, or encryption software tools for encrypting users' data, which expose users' data to leakage.

2) We constructed a new multi-authority CP-ABE scheme that achieves fine grained access control. Based on multi-authority CP-ABE [18], we realize efficient fine grained access control. Different from [18], we support many-write-many-read for users(which means after the owner creates one encrypted file on the storage server, other users with appropriate attributes can also update the encrypted file at a later time without any help from files' original owners) instead of 1-write-many-read. By employing multi-authority CP-ABE, we take a step on most of the existing work that are based on single authority CP-ABE for issuing private access keys. Moreover, almost all existing multi-authority CP-ABE based cloud storage systems did not consider the insurance of outsourced data integrity, unlike our scheme that supports data integrity checking. Furthermore, we support many-write-many-read for users that is almost lacked by all systems that support multi-authority CP-ABE schemes.

3) We proposed an efficient revocation approach for the proposed multi-authority CP-ABE scheme. Basing on the revocation method [19], we realize efficiently immediate user/attribute-level revocation while achieving both the backward and forward secrecy. In addition, we delegate the re-encryption right to cloud proxy servers to make use of cloud abundant resources. We take advantage over systems that support either do not support revocation in multi-authority CP-ABE or support either user or attribute revocation not both.

Future Work

There are several directions of future work related to this thesis. In this section we list a number of interesting topics that need further research. In this thesis we have proposed two methodologies to realize data confidentiality so that the cloud storage cannot disclose user's data and data confidentiality by implementing a fine-grained access control mechanism with privacy considerations. In this work, we focus on data confidentiality for data stored on cloud storage in two states: data in transit and at rest while did not consider data in use. Data in use refers to processing of data that is stored on a remote server.

However, the system may suffer from scalability issue if the number of requests exceeds millions requests, therefore, the scalability issue would be an interesting future research topic. In addition, we shall work on providing formal approaches for validating our proposed architecture along with a prototype. Another direction for future work could be ensuring data confidentiality while the data is used for processing purposes. For cryptographic-based data access control, user access is enabled by possessing the corresponding data decryption key(s). This opens the door for an authorized but malicious user to share her secret key with unauthorized users. More seriously, in

copyright-sensitive applications pirates may take this advantage to make profits by selling their secret keys (and hence access privileges) to others. These kinds of attacks are extremely harmful for copyright-sensitive applications since it very easy for key abusers to duplicate and distribute data decryption keys to others by ways such as email. Another threat that comes from attribute authorities in CP-ABE is their engagement in any malicious activities (as signing messages, generating and distributing keys) without the threat of being detected as they are treated as trusted entities. This is still a new area of research as almost work done in CP-ABE treats attribute authorities as trusted entities. Therefore, another direction for future work could be key abuse resistance. Last but not least, updating the signature whenever a revocation operation takes place can be a good candidate for future work.

6 References

[1] L. Badger, R. Patt-corner, J. Voas, L. Badger, R. Patt-corner, and J. Voas, "DRAFT Cloud Computing Synopsis and Recommendations of the National Institute of Standards and Technology," National Institute of Standards and Technology (NIST), May ,2011 . [Online] Available: http://csrc.nist.gov/publications/drafts/800-146/Draft-NIST-SP800-146.pdf.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009.

[3] CircleID Reporter, " Survey: Cloud computing 'no hype', but fear of security and control slowing adoption," Feb. 26, 2009, [Online]. Available: http://www.circleid.com/posts/20090226_cloud_computing_hype_security/.

[4] T. Velte, A. Velte, and R. C. Elsenpeter, Cloud Computing, A Practical Approach, First Edition. McGraw Hill Professional, 2009.

[**5**] B. Sosinsky, Cloud Computing Bible, First Edition. John Wiley & Sons,2010.

[**6**] P. Sprenger. Sun on privacy: 'get over it'. Wired, January 26, 1999. Available at: http://www.wired.com/politics/law/ news/1999/01/17538.

[**7**] R.Esguerra. Google CEO Eric Schmidt dismisses the importance of privacy. Electronic Fronteer Foundation, 10 December 2009. Available at: https://www.eff.org/deeplinks/2009/12/google-ceo-eric-schmidt-dismisses-privacy.

[8] W. Halton , "Security Issues and Solutions in Cloud Computing", June 25,2010,[Online], Available : http://wolfhalton.info/2010/06/25/security-issues-and- solutions-in- cloud-computing/

[9] "Dropbox authentication: insecure by design", Derek Newton, April 7, 2011, [Online]. Available: http://dereknewton.com/2011/04/dropbox-authentication-static-host-ids/.

[10] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, " Dark clouds on the horizon: Using cloud storage as attack vector and online slack space," In *Proceeding of the 20th USENIX Conference on Security*, 2011.

[11] http://techlogon.com/2012/03/09/box-com-security-issues-for-personal- accounts/

[12] https://spideroak.com/

[13] W.Hu, T. Yang, and J. N. Matthews. "The good, the bad and the ugly of consumercloud storage." ACM SIGOPS Operating Systems Review, pp. 110–115, 2010.

[14] X. Yu and Q. Wen, "A View about Cloud Data Security from Data Life Cycle," *Computational Intelligence and Software engineering*, no. 4072020, pp. 0–3, 2010.

[15] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring data storage security in Cloud Computing,” *2009 17th International Workshop on Quality of Service*, pp. 1–9, Jul. 2009.

[16] A. Juels and J. Burton S. Kaliski, “PORs: Proofs of Retrievability for Large Files,” In *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 584–597, 2007.

[17] K. Yang, X. Jia, K. Ren, and B. Zhang, "DAC-MACS: Effective data access control for multi-authority cloud storage systems," *2013 Proceedings IEEE INFOCOM*, pp.2895-2903, April 2013.

[18] K. Yang, and X. Jia, "Expressive, Efficient and Revocable Data Access Control for Multi-Authority Cloud Storage," *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[19] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," 2010 *Proceedings IEEE INFOCOM*, pp.1-9, March 2010.

[20] R. L. Krutz and R. D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, First Edition. John Wiley & Sons, Jul 15, 2010.

[21] I. T.Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, First Edition. O’Reilly Media, Inc., 2009.

[22] R. Buyya, J. Broberg, and A. M. Goscinski, Cloud Computing: Principles and Paradigms, First Edition. John Wiley & Sons, 2010.

[23] KPMG, "The Cloud: Changing the Business Eco System," KPMG, 2011, [Online]. Available: http://www.kpmg.com/IN/en/IssuesAndInsights/ThoughtLeadership/The_Cloud_Changing_the_Business_Ecosystem.pdf.

[24] M. Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, First Edition. Que Publishing, 2008.

[25] F.Liu, J. Tong, J. Mao, R. B. Bohn, J. V. Messina, M. L. Badger, and D. M. Leaf, "NIST Cloud Computing Reference Architecture," National Institute of Standards and Technology(NIST), September 08, 2011,[Online]. Available: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505

[26] Storage Networking Industry Association (SNIA), "Implementing, Serving, and Using Cloud Storage. Cloud Storage Initiative," SNIA, October, 2010, [Online]. Available: http://www.snia.org/sites/default/files/2010-10-WP-ImplementingServingandUsingTheCloud.pdf

[27] Storage Networking Industry Association (SNIA), "Cloud Data Management Interface (CDMI). Technical Position Version 1.0.1," SNIA, September 15, 2011, [Online]. Available: http://snia.org/sites/default/files/CDMI_SNIA_Architecture_v1.0.1.pdf

[28] "Amazon S3," [Online]. Available: http://aws.amazon.com/s3/

[29] "Google Cloud Storage," [Online]. Available: http://code.google.com/apis/storage/docs/getting-started.html

[30] "Dropbox," [Online]. Available: https://www.dropbox.com/

[31] "How secure is Dropbox?" [Online]. Available: https://www.dropbox.com/help/27

[32] R. Bhadauria, " a Survey on Security Issues in Cloud Computing," In *Computing Research Repository (CoRR)*, 2011.[Online] abs/1109.5388, Avaiable: http://dblp.uni-trier.de/db/journals/corr/corr1109.html#abs-1109-5388

[33] B. Furht and A. Escalante, Handbook of Cloud Computing, First Edition, Springer, 2010.

[34] M. Carroll, P. Katz, and A. V. Mere, "Going virtual: popular trend or real prospect for enterprise information systems," In *Proceedings of 12th International Conference on Enterprise Information Systems*, p. 214 - 222, June 2010.

[35] A. Singh and G. Tech, "Server-Storage Virtualization : Integration and Load Balancing in Data Centers," *In International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1-12, Nov. 2008.

[36] "European Network and Information Security Agency (ENISA)," [Online]. Available:www.enisa.europa.eu/

[37] "Cloud Security Alliance(CSA)," [Online]. Available:https://cloudsecurityalliance.org/

[38] D. Chatted and G. Hagen, "Cloud Computing: Benefits, risks and recommendations for information security," Tech. rep. The European Network and Information Security Agency (ENISA). 2009. [Online] Available: http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment

[39] Cloud Security Alliance (CSA), "Top Threats to Cloud Computing v1.0,"Tech. rep. v1.0. Cloud Security Alliance (CSA), March, 2010. [Online] Available: http://www.cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf

[40] J. E. Vascellaro, "Google Discloses Privacy Glitch", Wall Street Journal, March, 2009, [Online]. Available: http://blogs.wsj.com/digits/2009/03/08/1214/

[41] C. Foresman, "Brief Facebook glitch sent private messages to wrong users", Ars Technical, February, 2010, [Online]. Available: http://arstechnica.com/tech-policy/news/2010/02/brief- facebook-glitch-sent-private-messages-to-wrong- users.ars

[42] P. Wong, "Conversations about the Internet #5: Anonymous Facebook Employee", the Rumpus, January, 2010, [Online]. Available: http://therumpus.net/2010/01/conversations-about-the-internet-5-anonymous-facebook- employee/

[43] R. Thompson, "Hacked Facebook Applications reach out to Exploit Sites in Russia", AVG, October, 2009, [Online]. Available: http://blogs.avg.com/news-threats/hacked- facebook-applications-reach-out-to-exploit-sites-in-russia/

[44] S. Perez, "How Safe are Facebook Applications?" Read Write Social, October, 2009, [Online]. Available: http://readwrite.com/2009/10/16/how_safe_are_facebook_applications

[45] M. Jensen, N. Gruschka, and N. Luttenberger, "The Impact of Flooding Attacks on Network-based Services," *2008 Third International Conference on Availability, Reliability and Security*, pp. 509–513, Mar. 2008.

[46] M. Chew, D. Balfanz and B. Laurie, 2008, (Under)mining Privacy in Social Networks, *2008 IEEE Symposium on Security and Privacy Workshop: Web 2.0 Security and Privacy - W2SP* 2008,[Online]. Available: http://w2spconf.com/2008/papers/s3p2.pdf

[47] M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "On Technical Security Issues in Cloud Computing," *2009 IEEE International Conference on Cloud Computing*, pp. 109–116, 2009.

[48] M. Arrington, "Celebrity Twitter Accounts Hacked ", TechCrunch, January, 2009, [Online]. Available: http://techcrunch.com/2009/01/05/either-fox-news-had-their-twitter-account-hacked-or-bill-oreilly-is-gay-or-both/

[49] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage " Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds", In *Proceedings of the 16th ACM conference on Computer and communications security*, pp.199-212, Nov. 2009.

[50] R. Hasan, S. Myagmar, A. J. Lee, and W. Yurcik, "Toward a threat model for storage systems," *Proceedings of the 2005 ACM workshop on Storage security and survivability - StorageSS '05*, p. 94, 2005.

[51] D. Chen and H. Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," *2012 International Conference on Computer Science and Electronics Engineering*, no. 973, pp. 647–651, Mar. 2012.

[52] M. Kurta, "On the Road Towards Cloud Computing Services", August, 2010, [Online]. Available:http://oaithesis.eur.nl/ir/repub/asset/7791/M901%20IENE-Kurta_333055.pdf

[53] European Commission,"ARTICLE 29 DATA PROTECTION WORKING PARTY", July, 2012, [Online].Available:http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp196_en.pdf

[54] M. Borgmann, T. Hahn, M. Herfert, T. Kunz, M. Richter, U. Viebeg, and S. Vowe, "On the Security of Cloud Storage Services," Fraunhofer Institute for Secure Information Technology SIT, March, 2012, [Online]. Available: http://www.sit.fraunhofer.de/en/cloudstudy.html

[55] E. Hamburger ,"Google Drive vs. Dropbox, SkyDrive, SugarSync, and others: a cloud sync storage face-off," The verge, April 24, 2012, [Online]. Available: http://www.theverge.com/2012/4/24/2954960/google-drive-dropbox-skydrive-sugarsync- cloud-storage-competition

[56] W. Mossberg, "Many Devices, Many Files and Four Ways to Share Them," AllThings, July 31, 2012, [Online]. Available: http: http://allthingsd.com/20120731/many-devices- many-files-and-four-ways-to-share-them/

[57] "Open Source vs. Proprietary Software," bloomtools,January 31, 2012.[Online]. Available: http://www.bloomtools.com/articles/open-source-vs-proprietary- software.html

[58] S. Lesem, "Understanding Cloud Storage APIs: Standards, Functions, Lock-in, and What's Next," November 17, 2009, [Online]. Available: http://cloudstoragestrategy.com/2009/11/understanding-cloud-storage-apis.html

[59] D. Floyer, "Integration of the Storage Optimization Stack," Wikibon, June 09, 2010, [Online]. Available:

http://wikibon.org/wiki/v/Integration_of_the_Storage_Optimization_Stack#Storage_Optimization_Techniques

[60] Y. V. Lokeshwari, B. Prabavathy, and ChitraBabu, "Optimized Cloud Storage with High Throughput De-duplication Approach," *International Conference on Emerging Technology Trends*, 2011.

[61] T. Dierks, and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008, [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt

[62] M. Williams, "New security concerns with cloud storage," Faronics, August 23, 2012, [Online]. Available:http://www.faronics.com/2012/new-security-concerns-with-cloud-storage-2/

[63] T. Hahn, T. Kunz, M. Schneider, and S. Vowe, "Vulnerabilities through Usability Pitfalls in Cloud Services: Security Problems due to Unverified Email Addresses," *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 850–856, Jun. 2012.

[64] "Encryption for cloud storage," The University of Edinburgh, Information Services November 14, 2011, [Online]. Available: http://www.ed.ac.uk/schools- departments/information-services/services/computing/desktop- personal/security/encryption/cloud

[65] "About data encryption for cloud storage," Symantec, March 17, 2011, [Online]. Available: http://www.symantec.com/business/support/index?page=content&id=HOWTO45099

[66] "Online Storage Services Review," TopTenREVIEWS[Online]. Available: http://online- storage-service-review.toptenreviews.com/

[67] "Secure Cloud-based File Sharing and Collaboration," Intel, [Online]. Available: http://cloudsecurity.intel.com/solutions/secure-file-sharing

[68] A. Sumner, "Alternatives to Dropbox for cloud-based file syncing and sharing," Stratepedia Blog, May 18, 2011, [Online]. Available: http://blog.stratepedia.org/2011/05/18/alternatives-to-dropbox-for-cloud-based-file- syncing-and-sharing/

[69] R. Lepofsky, "File Sharing or Privacy Breaching Service? Beware," Infosec Island, May 23, 2011, [Online]. Available: http://www.infosecisland.com/blogview/13890-File- Sharing-or-Privacy-Breaching-Service-Beware.html

[70] G.-Z. Sun, Y. Dong, D.-W. Chen, and J. Wei, "Data Backup and Recovery Based on Data De-Duplication," *2010 International Conference on Artificial Intelligence and Computational Intelligence*, pp. 379–382, Oct. 2010.

[71] T. Wu, W. Lee, and C. F. Lin, "Cloud storage performance enhancement by real-time feedback control and de-duplication,."*Wireless Telecommunications Symposium (WTS), 2012* , pp.1-5, 18-20 April 2012.

[72] Q. He, Z. Li, and X. Zhang, "Data deduplication techniques," *2010 International Conference on Future Information Technology and Management Engineering*, pp. 430– 433, Oct. 2010.

[73] O. Heen, C. Neumann, L. Montalvo, and S. Defrance, "Improving the Resistance to Side-Channel Attacks on Cloud Storage Services," *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, May 2012.

[74] D. Harnik, B.Pinkas, and A. Shulman-Peleg, "Side Channels in Cloud Services: Deduplication in Cloud Storage," *IEEESecurity & Privacy*, vol.8, no.6, pp.40-47, Nov.- Dec. 2010.

[75] M. Alhamad, T. Dillon, and E. Chang, "Conceptual SLA Framework for Cloud Computing," *2010 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pp. 606–610, 2010.

[76] B. R. Kandukuri, R. P. V., and A. Rakshit, "Cloud Security Issues," *2009 IEEE International Conference on Services Computing*, pp. 517–520, 2009.

[77] X. Zhang, H. Du, J. Chen, Y. Lin, and L. Zeng, "Ensure Data Security in Cloud Storage," *2011 International Conference on Network Computing and Information Security*, pp. 284–287, May 2011.

[78] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, "Security SLAs for Federated Cloud Services," *2011 Sixth International Conference on Availability, Reliability and Security*, pp. 202–209, Aug. 2011.

[79] S. A. de Chaves, C. B. Westphall, and F. R. Lamin, "SLA Perspective in Security Management for Cloud Computing," *2010 Sixth International Conference on Networking and Services*, pp. 212–217, Mar. 2010.

[80] C. Almond, "A Practical Guide to Cloud Computing Security," August 27, 2009, [Online]. Available:http://www.avanade.com/Documents/Research%20and%20Insights/practicalguidetocloudcomputingsecurity574834.pdf

[81] W. Itani, A. Chehab, and A. Kayssi, "Energy-efficient platform-as-a-service security provisioning in the cloud," *2011 International Conference on Energy Aware Computing*, pp. 1–6, Nov. 2011.

[82] Y. Tan and X. Wang, "Research of cloud computing data security technology," *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 2781–2783, Apr. 2012.

[83] M. Ahmed, Y. Xiang, and S. Ali, "Above the Trust and Security in Cloud Computing: A Notion Towards Innovation," *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, pp. 723–730, Dec. 2010.

[84] A. J. Choudhury, P. Kumar, M. Sain, H. Lim, and H. Jae-Lee, "A Strong User Authentication Framework for Cloud Computing," *2011 IEEE Asia-Pacific Services Computing Conference*, pp. 110–115, Dec. 2011.

[85] S. S. L. Tls, R. Oppliger, R. Hauser, P. Ag, D. Basin, and E. T. H. Zurich, "SSL/TLS Session-Aware User Authentication," *Computers & Security*, pp. 64–70, March, 2008.

[86] K. Sarikaya and A. B. Can, “Password-based client authentication for SSL/TLS using ElGamal and Chebyshev polynomials,” *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–5, Oct. 2011.

[87] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” "Practical techniques for searches on encrypted data," *2000 IEEE Symposium on Security and Privacy*, pp.44-55, 2000.

[88] E. Goh, “Secure indexes,” in Cryptology ePrint Archive, Technical Report 2003/216, 2003, [Online]. Available: http://eprint.iacr.org/2003/216

[89] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” *In Proceedings of the 13th ACM conference on Computer and communications security(CCS '06)*, pp. 79 - 88, 2006.

[90] R. Koletka and A. Hutchison, “An architecture for secure searchable cloud storage,” *2011 Information Security for South Africa*, pp. 1–7, Aug. 2011.

[91] M. Kuzu, M. S. Islam, and M. Kantarcioglu, “Efficient Similarity Search over Encrypted Data,” *2012 IEEE 28th International Conference on Data Engineering*, pp. 1156–1167, Apr. 2012.

[92] M. Li, S. Yu, N. Cao, and W. Lou, “Authorized Private Keyword Search over Encrypted Data in Cloud Computing,” *2011 31st International Conference on Distributed Computing Systems*, pp. 383–392, Jun. 2011.

[93] J. Li, C. Jia, J. Li, and Z. Liu, “A Novel Framework for Outsourcing and Sharing Searchable Encrypted Data on Hybrid Cloud,” *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*, pp. 1–7, Sep. 2012.

[94] C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. dissertation, Stanford University, 2009, [Online]. Available: http:// crypto.stanford.edu/craig

[95] M. Brenner, J. Wiebelitz, G. Von Voigt, and M. Smith, “Secret Program Execution in the Cloud Applying Homomorphic Encryption,” *2011 Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST),*vol. 5, no. June, pp. 114–119, 2011.

[96] M. Tebaa, S. E. L. Hajji, and A. E. L. Ghazi, “Homomorphic Encryption Applied to the Cloud Computing Security,” *2012 National Days of Network Security and Systems(JNS2),* vol. I, pp. 8–11, 2012.

[97] R. Kui, W. Cong, and W. Qian, "Security Challenges for the Public Cloud," *IEEEInternet Computing*, vol.16, no.1, pp.69-73, Jan.-Feb. 2012.

[98] J. Chen, Y. Wang, and X. Wang, “On-Demand Security Architecture for Cloud Computing," *Computer* , vol.45, no.7, pp.73,78, 2012.

[99] G. Kulkarni, J. Gambhir, T. Patil, and A. Dongare, “A security aspects in cloud computing,” *2012 IEEE International Conference on Computer Science and Automation Engineering*, pp. 547–550, Jun. 2012.

[100] W.-G. Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 995–1003, Jun. 2012.

[101] S. Kamara and K. Lauter, "Cryptographic Cloud Storage Architecture of a Cryptographic Storage Service," *in Financial Cryptography and Data Security, ser. Lecture Notes in Computer Science. Springer*, pp. 1–14 , 2010.

[102] "Wuala," [Online]. Available: http://www.wuala.com

[103] R. Freudenreich, "Boxcryptor," [Online]. Available: http://www.boxcryptor.com/

[104] P. Deniability, P. Gasti, G. Ateniese, and M. Blanton, "Leakage-Resilient Client-side: sharing files via public-key deniability," *In Proceedings of the 9th annual ACM workshop on Privacy in the electronic society (WPES '10)* ,pp. 31–42, 2010.

[105] J. Hwang and H. Chuang, "A Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service," *2011 International Conference on Information Science and Applications*, pp. 1–7, Apr. 2011.

[106] V. Joshi, P. Sanghavi, , "Three tier data storage security in cloud using Face fuzzy vault," *Computing, Communication and Applications (ICCCA), 2012 International Conference on* , vol., no., pp.1-6, 22-24 Feb. 2012.

[107] L. Hao and D. Han, "The study and design on secure-cloud storage system," *2011 International Conference on Electrical and Control Engineering*, pp. 5126–5129, Sep.2011.

[108] F. Rocha and M. Correia, "Lucy in the sky without diamonds: Stealing confidential data in the cloud," *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 129–134, Jun. 2011.

[109] J. M. Rosenbaum, "In Defense of the delete Key," 2000, [Online]. Available: http://simson.net/ref/2005/csci_e-170/ref/rosenbaum_deletekey.pdf

[110] " Disposition of Computer Hard Drives Attachment 2 Specifications for Sanitization of Hard Drives," OSS-SPECTRUM PROJECT, 2008, , [Online]. Available: http://www.spectrumwest.com/Attach2.htm

[111] P. F. Bennison, H. St, P. J. Lasher, C. Street, and R. Only, "Data security issues relating to end of life equipment*," 2004 Conference Record. 2004 IEEE International Symposium on Electronics and the Environment*, pp. 317–320, 2004.

[112] S. M. Diesburg and A. Andy Wang."A survey of confidential data storage and deletion methods," *ACM Computer Survey (CSUR)*, Article 2 pp. 24-25 , December 2010.

[113] Z. N. J. Peterson, R. Burns, J. Herring, A. Stubblefield, and A. D. Rubin, " Secure deletion for a versioning file system," *In Proceedings of the 4th conference on USENIX Conference on File and Storage Technologies - Volume 4 (FAST'05)*, pp. 143–154, 2005.

[114] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 903–916, 2012.

[115] R. Perlman, “File System Design with Assured Delete Radia Perlman,” *In Proceedings of the Third IEEE International Security in Storage Workshop (SISW '05). IEEE Computer Society*, 2005.

[116] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, “A Secure Cloud Backup System with Assured Deletion and Version Control,” *2011 40th International Conference on Parallel Processing Workshops*, pp. 160–167, Sep. 2011.

[117] M. Farhatullah, "ALP: An authentication and leak prediction model for Cloud Computing privacy," *2013 IEEE 3rd International Advance Computing Conference (IACC)*, pp.48, 51, 22-Feb. 2013.

[118] W. Tsai, and S. Qihong, "Role-Based Access-Control Using Reference Ontology in Clouds," *2011 10th International Symposium on Autonomous Decentralized Systems (ISADS)*, pp.121, 128, 23-27 March 2011.

[119] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable secure file sharing on untrusted storage,” in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, Berkeley, CA, USA: USENIX Association, pp. 29–42, 2003.

[120] A. Sahai and B.Waters, "Fuzzy Identity Based Encryption," *In Advances in Cryptology - Eurocrypt*, volume 3494 of LNCS, pp. 457-473, Springer, 2005.

[121] A. Shamir, "Identity Based Cryptosystems and Signature Schemes," In *Advances in Cryptology-CRYPTO*, volume 196 of LNCS, pp. 37-53.Springer, 1984.

[122] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” In *Proceedings of the 13th ACM conference on Computer and communications security* (CCS '06). ACM, New York, NY, USA, pp. 89–98, 2006.

[123] R. Ostrovsky, A. Sahai, and B. Waters," Attribute-based Encryption with Non-monotonic Access Structures.", In *Proceedings of the 14th ACM conference on Computer and communications security (CCS '07)*. ACM, New York, NY, USA, pp. 195–203, 2007.

[124] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute based encryption,” In *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[125] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” In *Proceedings of the 4th International Conference on Practice and Theory in Public Key Cryptography (PKC’11)*. Springer, pp. 53–70, 2011.

[126] S. Tu, S. Niu, H. Li, Y. Xiao-ming, and M. Li, “Fine-grained Access Control and Revocation for Sharing Data on Clouds,” *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pp. 2146–2155, May 2012.

[127] R. Z. - and P. C. -, “A Dynamic Cryptographic Access Control Scheme in Cloud Storage Services,” *International Journal of Information Processing and Management*, vol. 4, no. 1, pp. 104–111, Jan. 2013.

[128] Y. Ming, L. Fan, H. Jing-li, and W. Zhao-li, “An Efficient Attribute based Encryption Scheme with Revocation for Outsourced Data Sharing Control,” 2011.

[129] J. Hur and D. K. Noh, “Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, Jul. 2011.

[130] J.-M. Do, Y.-J. Song, and N. Park, “Attribute Based Proxy Re-encryption for Data Confidentiality in Cloud Computing Environments,” *2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, pp. 248–251, May 2011.

[131] R. Lu and X. Shen, “ESPAC : Enabling Security and Patient-centric Access Control for eHealth in cloud computing Mrinmoy Barua , Xiaohui Liang ,” 2011.

[132] L. Zhiquan, C. Hong, M. Zhang and D. Feng, "A secure and efficient revocation scheme for fine-grained access control in cloud storage,", *2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp.545,550, 3-6 Dec. 2012.

[133] D. Jiazhu, “A privacy-preserving access control in outsourced storage services,” *2011 IEEE International Conference on Computer Science and Automation Engineering*, pp. 247–251, Jun. 2011.

[134] M. Chase, " Multi-authority attribute-based encryption," In *The Fourth Theory of Cryptography Conference (TCC 2007)*, 2007.

[135] S. Muller, S. Katzenbeisser, and C. Eckert, "On multi-authority ciphertext-policy attribute-based encryption," *In Bulletin of the Korean Mathematical Society* 46, pp. 803-819, 2009.

[136] M. Chase and S.M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," *In Proceedings of the 16th ACM conference on Computer and communications security (CCS '09). ACM*, New York, NY, USA, pp. 121-130, 2009.

[137] T. Jung, X. Li, Z. Wan, and M. Wan, "Privacy preserving cloud data access with multi-authorities," *2013 Proceedings IEEEINFOCOM*, pp.2625-2633, April 2013.

[138] H. Lin, Z. Cao, X. Liang, and J. Shao, “Secure threshold multi-authority attribute based encryption without a central authority,” in *Proceedings: International Conference on Cryptology in India-INDOCRYPT’08*, vol. 5365 of *Lecture Notes in Computer Science*, (Kharagpur, India), pp. 426–436, Springer, December 2008.

[139] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," In *EUROCRYPT*, pp. 568–588, 2011.

[140] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, “Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles,” in *Proceedings: European Symposium on Research in Computer Security-ESORICS’11* , vol. 6879 of *Lecture Notes in Computer Science*, p. 278297, Springer, September 12-14 2011.

[141] J. Hur, "Improving Security and Efficiency in Attribute-Based Data Sharing," IEEE Transactions on Knowledge and Data Engineering, vol.25, no.10, pp.2271-2282, Oct. 2013.

[142] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient".In *Proceedings of the 1st International Conference on Cryptology in Africa – AFRICACRYPT 2008*, volume 5023 of LNCS, Springer, 2008.

[143] M. Naor and B. Pinkas, "Efficient trace and revoke schemes," In *Proceedings of the 4th International Conference on Financial Cryptography – FC2000*, volume 1962 of LNCS, pp. 1–20. Springer, 2000.

[144] N. Attrapadung and H. Imai, “Conjunctive Broadcast and Attribute-Based Encryption,” *In Pairing ’09: The 3rd International Conference on Pairing-Based Cryptography*, volume 5671 of Lecture Notes in Computer Science, pp. 248–265. Springer-Verlag, 2009.

[145] Z. Xu and K.M. Martin, " Dynamic User Revocation and Key Refreshing for Attribute-Based Encryption in Cloud Storage," In*2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 844–849. IEEE, 2012.

[146] Q. HUANG, Z. MA, J. FU, X. NIU, and Y. YANG, "Attribute Based DRM Scheme with Efficient Revocation in Cloud Computing," Journal of Computers, North America, November, 2013.

[147] A. Boldyreva, V. Goyal, and V. Kumar, “Identity-Based Encryption with Efficient Revocation,” *Proc. ACM Conf. Computer and Comm. Security*, pp. 417-426, 2008.

[148] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, “Secure Attribute-Based Systems,” Proc. ACM Conf. Computer and Comm. Security, 2006.

[149] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W.Jonker, " Mediated Ciphertext-Policy Attribute-based Encryption and its Application," In *Information Security Applications*, pp. 309–323. Springer, 2009.

[150] J. Camenisch, " Efficient and generalized group signatures," In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques, EUROCRYPT’97*, pp. 465–479, Berlin, Heidelberg, Springer-Verlag, 1997.

[151] R. L. Rivest, A. Shamir, and Y. Tauman, " How to leak a secret," In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pp. 554–567. Springer-Verlag, 2001.

[152] X.Boyen,"Mesh signatures," In *Proceedings of the 26th annual international conference on Advances in Cryptology, EUROCRYPT ’07*, pp. 210–227, Berlin, Heidelberg, 2007.

[153] D. Cao, B. Zhao, X. Wang, J. Su, and G. Ji, "Multi-authority Attribute-Based Signature," *2011 Third International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pp. 668-672, 2011.

[154] D. Cao, B. Zhao, X. Wang, and J. Su, "Multi-authority Attribute-based Signature Scheme with Constant-length Signature", In *Proceedings International Conference on Computer and Applications (CCA 2012)*, 2012.

[155] X. Liu, R. Zhang, and R. Xue, "Multi-Central-Authority Attribute-Based Signature,*" 2012 International Symposium on Information Science and Engineering (ISISE)*, pp.173-178, Dec. 2012.

[156] H. Maji, M. Prabhakaran, M. Rosulek, "Attribute-Based Signatures," *CT-RSA*, volume 6558 of Lecture Notes in Computer Science, pp. 376–392, 2011.

[157] T. Okamoto, and K.Takashima, " Efficient attribute-based signatures for non-monotone predicates in the standard model," In *PKC 2011*, vol. 6571,Springer Heidelberg, pp. 35–52, 2011.

[158] M.Mambo, E.Okamoto, "Proxy cryptosystems: delegation of the power to decrypt

ciphertexts," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, pp. 54–63,1997.

[159] M. Blaze, G.Bleumer, M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," In *EUROCRYPT 1998*, vol. 1403, Springer, Heidelberg, pp. 127–144, 1998.

[160] M. Jakobsson, "On Quorum Controlled Asymmetric Proxy Re-Encryption," In *Proceedings of Public Key Cryptography*, Lecture Notes in Computer Science(LNCS), volume 1560, Springer, Heidelberg, pp. 112–121, 1999.

[161] L. Zhou, M. A. Marsh, F. B. Schneider, A.Redz, " Distributed blinding for ElGamal re-encryption," In *Proceedings of 25th IEEE International Conference on Distributed Computing Systems*, IEEE Computer Society pp. 815–824, 2005.

[162] A. Ivan, Y. Dodis, "Proxy Cryptography," In *Proceedings of the Network and Distributed System Security Symposium*, the Internet Society, 2003.

[163] T. Matsuo," Proxy Re-encryption Systems for Identity-Based Encryption," In *Proceedings of Pairing 2007*, Lecture Notes in Computer Science(LNCS), volume 4575, pp. 247–267, Springer, Heidelberg, 2007.

[164] M. Green, G.Ateniese, " Identity-Based Proxy Re-Encryption," In *Proceedings of Pairing 2007*, Lecture Notes in Computer Science(LNCS), volume 4575, pp. 288–306, Springer, Heidelberg, , 2007.

[165] J. Li, G. Zhao, X. Chen, D. Xie, C. Rong, W. Li, L. Tang, and Y. Tang, "Fine-Grained Data Access Control Systems with User Accountability in Cloud Computing," *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pp. 89–96, Nov. 2010.

[166] W. Lafayette, E. Bertino, and W. Lafayette, "Privacy Preserving Delegated Access Control in the Storage as a Service Model," *2012 IEEE 13th International Conference on Information Reuse and Integration (IRI)*, pp. 645–652, 2012.

[167] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 556–563, May 2012.

[168] Y. Zhu, H. Hu, G.-J. Ahn, D. Huang, and S. Wang, "Towards temporal access control in cloud computing," *2012 Proceedings IEEE INFOCOM*, pp. 2576–2580, Mar. 2012.

[169] G. Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang, "Trusted Data Sharing over Untrusted Cloud Storage Providers," 2010 *IEEE Second International Conference on Cloud Computing Technology and Science*, pp. 97–103, Nov. 2010.

[170] M. Nabeel, N. Shang, E. Bertino, "Privacy Preserving Policy-Based Content Sharing in Public Clouds," *IEEE Transactions on Knowledge and Data Engineering*, vol.25, no.11, pp.2602-2614, Nov. 2013.

[171] Y. Yang and Y. Zhang, "A Generic Scheme for Secure Data Sharing in Cloud," 2011 *40th International Conference on Parallel Processing Workshops*, pp. 145–153, Sep. 2011.

[172] B. K. Samanthula, G. Howser, Y. Elmehdwi, and S. Madria, "An Efficient and Secure Data Sharing Framework using Homomorphic Encryption in the Cloud," In *Proceedings of the 1st*

International Workshop on Cloud Intelligence (Cloud-I '12), ACM, New York, NY, USA, 2012.

[173] P. Deniability, P. Gasti, G. Ateniese, and M. Blanton, "Deniable cloud storage: sharing files via public-key deniability," In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society (WPES '10)*, pp. 31–42, 2010.

[174] B. Priyadharshini, "Data Integrity in Cloud Storage," *2012 International Conference Advances in Engineering, Science and Management (ICAESM)*, pp. 261–265, 2012.

[175] J. Feng, Y. Chen, W.-S. Ku, and P. Liu, "Analysis of Integrity Vulnerabilities and a Non-repudiation Protocol for Cloud Data Storage Platforms," 2010 39th International Conference on Parallel Processing Workshops, pp. 251–258, Sep. 2010.

[176] Z. Xiao and Y. Xiao, "Security and Privacy in Cloud Computing," *Communications Surveys & Tutorials*, IEEE, pp.1-17.

[177] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," In Proceedings of the 14th ACM conference on Computer and communications security (CCS '07), pages 598-609, 2007.

[178] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. of SecureComm'08*, 2008.

[179] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia "Dynamic Provable Data Possession," *Proc. 16th ACM Conf, Computer and Comm. Security (CCS '09)*,2009.

[180] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," Proc. 16th ACM conference on Computer and communications security, pp. 187-198, 2009.

[181] R. S. Kumar and A. Saxena, "Data integrity proofs in cloud storage," 2011 *Third International Conference Communication Systems and Networks (COMSNETS)*, pp.1-4, 4-8 Jan. 2011.

[182] L. Wenjun and B. Guojing , "Ensuring the data integrity in cloud data storage," Cloud Computing and 2011 *IEEE International Conference on Intelligence Systems (CCIS)*, pp.240-243, 15-17 Sept. 2011.

[183] W. Shao-hui, C. Su-qin, C. Dan-wei, and W. Zhi-wei, -wei, S)cloud data storage," Cloud Data Storage Security With Zero Knowledge Privacy" [Online]. Available: http://eprint.iacr.org/2012/365.pdf.

[184] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," Network, IEEE , vol.24, no.4, pp. 19–24, 2010.

[185] K. Yang, S. Member, X. Jia, and S. Member, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, pp. 1–11, 2012.

[186] Q. Wang, S. Member, C. Wang, and K. Ren, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing, "*IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[187] L. Li, L. Xu, J. Li, and C. Zhang, "Study on the third-party audit in cloud storage service," 2011 *International Conference on Cloud and Service Computing*, pp. 220–227, Dec. 2011.

[188] "Cloud services outage report," [Online]. Available: http://bit.ly/cloud outage.

[189] "Amazon s3 July 2008 outage," [Online]. Available: http://www.networkworld.com/ news/2008/072108-amazon-outages.html.

[190] H. Xia and A. A. Chien, "RobuSTore: a distributed storage architecture with robust and high performance," *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*, pp.1-11, November 2007.

[191] H. Weatherspoon and J. D. Kubiatowicz, "Erasure Coding Vs. Replication: A Quantitative Comparison," In Revised Papers from *the First International Workshop on Peer-to-Peer Systems (IPTPS '01)*, pp. 328-338, 2002.

[192] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, " RACS: a case for cloud storage diversity," In *Proceedings of the 1st ACM symposium on Cloud computing* (SoCC '10), pp. 229-240, 2010.

[193] Y. Singh and F. Kandah, "A secured cost-effective multi-cloud storage in cloud computing," 2011 *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 619–624, April 2011.

[194] C.-W.Chang, P. Liu, and J.-J. Wu, "Probability-Based Cloud Storage Providers Selection Algorithms with Maximum Availability," 2012 *41st International Conference on Parallel Processing*, pp. 199–208, Sep. 2012.

[195] M. a. Alzain, B. Soh, and E. Pardede, "MCDB: Using Multi-clouds to Ensure Security in Cloud Computing," 2011 *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, pp. 784–791, Dec. 2011.

[196] A. Bessani, M. Correia, B. Quaresma, F. André and P. Sousa, "DepSky: dependable and secure storage in a cloud-of-clouds", *EuroSys'11:Proc. 6th Conf. on Computer systems*, pp. 31-46, 2011.

[197**]** J. Shin, Y. Kim, W. Park, and C. Park, "DFCloud: A TPM-based secure data access control method of cloud storage in mobile devices," 2012 *IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp.551-556, Dec. 2012.

[198] I. Chuang, S. H. Li, K. C. Huang, Y. H. Kuo, "An effective privacy protection scheme for cloud computing,"*2011 13th International Conference on Advanced Communication Technology (ICACT)*, pp.260, 265, Feb. 2011.

[199] H. R. Patel, D. Patel, J. Chaudhari, S. Patel, and K. Prajapati, "Tradeoffs between performance and security of cryptographic primitives used in storage as a service for cloud computing," ACM, pp. 557-560, 2012.

[200] S. Malik and A. Sardana, "Secure Vault: A privacy preserving reliable architecture for Secure Social Networking," 2011 *7th International Conference on Information Assurance and Security (IAS)*, pp.116-121, Dec. 2011.

[201] "TrueCrypt," [Online]. Available: http://www.truecrypt.org/

[202] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," *Computer*, vol. 45, no. 1, 2012.

[203] tahoe-lafs.org (2008-08-20).Retrieved on 2013-09-05.

[204] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol.24, no.1, pp.131-143, Jan. 2013.

[205] F. Zhao, T. Nishide, and K. Sakurai, "Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems," in Information Security Practice and Experience, ser. Lecture Notes in Computer Science, F. Bao and J. Weng, Eds. Springer Berlin / Heidelberg, vol. 6672, pp. 83–97, 2011.

[206] B. Crispo, M. Ion, and M. Asghar," ESPOON: Enforcing Encrypted Security Policies in Outsourced Environments", 2011 *Sixth International Conference on Availability, Reliability and Security (ARES)*, pp.99-108, August, 2011.

[207] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology–CRYPTO 2001*.Springer, 2001, pp. 41–62, 2001.

[208] S. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: management of access control evolution on outsourced data," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, pp. 123–134, 2007.

[209] S. D. C. d. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "A data outsourcing architecture combining cryptography and access control," in *Proceedings of the 2007 ACM workshop on Computer security architecture*, ser. CSAW '07. New York, NY, USA, ACM, , pp. 63–69, 2007.

[210] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and efficient access to outsourced data," in *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM, 2009, pp. 55–66.

[211] "Mozy," [Online]. Available: http://www.mozy.com

[212] "CrashPlan," [Online]. Available: http://www.crashplan.com

[213] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributedfile system," In *ICDCS*, pp. 617–624, 2002.

[214] G. Shaniqng, Z. Yingpei, "Attribute-based Signature Scheme," *International Conference on Information Security and Assurance, 2008ISA* , pp.509,511, April 2008.

[215] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations,"NIST Special Publication 800-162 DRAFT, December, 2013.
[Online] Available : csrc.nist.gov/publications/drafts/800-162/sp800_162_draft.pdf

APPENDIX A

Background on CP-ABE Systems

Our proposed framework is build upon the ciphertext policy attribute-based encryption (CP-ABE) scheme. The concept of ABE was introduced along with another cryptography called fuzzy identity-based encryption (FIBE). Therefore, we can best understand CP-ABE by first introducing bilinear maps and LSSS. Knowledge of finite fields and elliptic curves is required to better explain the actual implementation of cyclic groups in bilinear maps. In this section, we give required background material on bilinear maps, LSSS, the formal definition of a CP-ABE scheme. We also present the formal definitions for multi-authority CP-ABE schemes.

I. Bilinear Maps

There are several definitions of bilinear maps, or pairings, in a cryptography paradigm. The differences between the expressions of these definitions are subtle. Generally speaking, bilinear maps associate pairs of elements from two algebra group to yield an element of a third algebra group that is linear in each of its arguments. According to Ben Lynn [1], the essential property of bilinear maps is that they give cyclic groups additional properties. We outline three definitions of bilinear maps in order of restrictiveness.

The General Bilinear Pairing

Definition: Let G_1, G_T be cyclic group of prime order p. Let G_2 be a group where

each element has order dividing r. In particular G_2 is not necessarily cyclic. A bilinear pairing or bilinear map e is an efficiently computable function

$$e: G_1 \times G_2 \longrightarrow G_T$$

such that

(i) (Non-degeneracy) $e(g_1; g_2) \neq 1_{G_T}$ for all $g_2 \in G_2$ if and only if $g_1 = 1_{G_1}$, and $e(g_1; g_2) \neq 1_{G_T}$ for all $g_1 \in G_1$ if and only if $g_2 = 1_{G_2}$.

(ii) (Bi-linearity) for all $g_1 \in G_1$ and $g_2 \in G_2$ and $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $a, b \in Z$.

The general bilinear pairing is the most flexibility. It solve the hash problem by reduce G_2 to be not necessarily cyclic. However, in this setting, the hardness assumption must be changed. Based on different scheme, we have to assume certain problems are hard in both groups. For example, we can assume that given $g_1, g_1^x \in G_1$ and $g_2 \in G_2$, there is no efficient algorithm to compute g_2^x.

II. .Linear Secret-Sharing Scheme

The idea of linear secret-sharing scheme (LSSS) and monotone span programs was discussed by Amos Beimel [2]. In a LSSS, dealer holds a secret and distributes the shares of the secret to parties. Parties can reconstruct the secret from a linear combination of the shares of any authorized set. A famous example of LSSS is the Shamir t-out-of-n threshold scheme[3]. In that scheme, the hardness of secret reconstruction depends on the hardness of polynomial reconstruction.

III. CP-ABE Definition

Ciphertext policy attribute-based encryption (CP-ABE) is becoming a promising cryptographic solution. It enables data owners to define their own access policies over user attributes and enforce the policies on the data to be distributed. In CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution. The data owner defines the access policies and encrypts data under the policies. Each user will be issued a secret key according to its attributes. A user can decrypt the ciphertext only when its attributes satisfy the access policies. Moreover, in CP-ABE schemes, the access policy checking is implicitly conducted inside the cryptography. That is, there is no one to explicitly evaluate the policies and make decisions on whether allows the user to access the data.

Basic Ciphertext-Policy Attribute-Based Encryption (CPABE)[4]scheme consists of four algorithms: *Setup, Encrypt, KeyGen,* and *Decrypt*.

Setup(λ ,U). The setup algorithm takes security parameter λ and attributes universe description as input *U*. It outputs the public parameters *PK* and a master key *MK*. The public key is used for encryption. The master key, held by the central authority, is used to generate user secret keys.

KeyGen(MK,S,PK). The key generation algorithm takes as input the master key *MK,* the public parameters PK, and a set of attributes *S* that describe the key. It outputs a private key *SK* associated with *S*.

Encrypt(PK,M,A). The encryption algorithm takes as input the public parameters *PK,* a message *M,* and an access structure *A* over the universe of attributes. 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. We assume that A is implicitly included in CT.

Decrypt (PK, CT, SK). The decryption algorithm takes as input the public parameters *PK*, a ciphertext *CT*, which contains an access policy *A*, and a private key *SK*, which is a private key for a set *S* of attributes. If the set *S* of attributes satisfies the access structure *A* then the algorithm will decrypt the ciphertext and return a message *M*.

A CP-ABE system is said to be correct if whenever PP, MSK are obtained by running the setup algorithm, CT is obtained by running the encryption algorithm on PP, M, A, SK is obtained by running the key generation algorithm on MSK, PP, S and S satisfies A, then Decrypt(PK, CT, SK) = M.

IV. Multi-Authority CP-ABE Definition

We now present formal definitions for multi-authority CP-ABE systems . A multi-authority Ciphertext-Policy Attribute-Based Encryption system is comprised of the following five algorithms:

Global Setup (λ) $\rightarrow$ GP The global setup algorithm takes in the security parameter λ and outputs global parameters GP for the system.

Authority Setup(GP) $\rightarrow$ SK, PK Each authority runs the authority setup algorithm with GP as input to produce its own secret key and public key pair, SK, PK.

Encrypt (M, A, GP, $\{$PK$\}$) $\rightarrow$ CT the encryption algorithm takes in a message M, an access structure A, the set of public keys for relevant authorities, and the global parameters. It outputs a ciphertext CT.

KeyGen (GID, GP, i, SK) $\rightarrow$ $K_{i,GID}$ The key generation algorithm takes in an identity GID, the global parameters, an attribute i belonging to some authority, and the secret key SK for this authority. It produces a key $K_{i,GID}$ for this attribute, identity pair.

Decrypt(CT, GP, $\{$PK$\}$, $\{K_{i,GID}\}$) $\rightarrow$ M The decryption algorithm takes in a ciphertext, the global parameters, the public keys for the relevant authorities, and a collection of keys corresponding to attribute, identity pairs all with the same fixed identity GID. It outputs the message M when the collection of attributes i satisfies the access structure corresponding to the ciphertext.

A multi-authority CP-ABE system is said to be correct if whenever GP is obtained from the global setup algorithm, $\{$PK, SK$\}$ are obtained by running the authority setup algorithm, CT is obtained from the encryption algorithm on the message M using the public keys $\{PK\}$, and $\{K_{i,GID}\}$ is a set of keys obtained from running the key generation algorithms with $\{$SK$\}$ for the same identity GID and for a set of attributes satisfying the access structure of the ciphertext, *Decrypt*(CT, GP, $\{$PK$\}$, $\{K_{i,GID}\}$) = M.

References

[1] B. Lynn. On the implementation of pairing-based cryptosystems. PhD thesis, Citeseer, 2007.

[2] A. Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[3] A. Shamir, " How to share a secret," *Commun. ACM*, pp. 612-613, November 1979.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption,"

In *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.