CLOUD-BASED DOCUMENT MANAGEMENT FRAMEWORK FOR BUSINESS PROCESS OPTIMIZATION

A. Nalaka Arjuna Premathilaka

(158240U)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

February 2019

CLOUD-BASED DOCUMENT MANAGEMENT FRAMEWORK FOR BUSINESS PROCESS OPTIMIZATION

A. Nalaka Arjuna Prematilaka

(158240U)

Thesis submitted in partial fulfillment of the requirements for the degree

Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

February 2019

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ……. Date: …….

Name: A. Nalaka Arjuna Premathilaka (158240U)

The supervisor/s should certify the dissertation with the following declaration.

The above candidate has carried out research for the Master's Dissertation under my supervision.

Signature of the supervisor: ……. Date: ……..

Name: Dr. Indika Perera

ABSTRACT

Nowadays, most of the corporates and private users are tend to use the Cloud services because of its benefits such as cost reduction, flexibility & scalability, high availability, accessibility and many more. When organizations upload their sensitive data to the public cloud storage in plain text, the main concern is data security & privacy. Cloud data must be protected from the external attackers, intruders and from Cloud storage owners as well. There are few examples that even the cloud storage providers were involved in the security breaches of the data in their storages. Therefore, In order to achieve the data security and privacy in the public cloud storages, usually data will be encrypted prior upload to the cloud. However, in public cloud storages such as Dropbox, Amazon S3, Mozy, and others, perform data deduplication to save space by removing the repeated chunks of the files or data blocks. This will help to reduce storage usage and has a cost benefit as well. But when the data is encrypted, the de-duplication process is not working as expected and storage space savings are lost.

This research focuses on identifying a method to overcome these issues in public cloud storages when storing sensitive data. This research implements a solution/framework for corporate and individual users to use public cloud storage by ensuring data security and space saving as well. Implementation of this research will introduce an additional application layer between public cloud storage and cloud users. It handles communication between the user and the cloud storage and performs the data de-duplication and encryption prior to uploading data chunks to the public cloud.

Keywords: Cloud storage, Data security, Data de-duplication, Data encryption, Public cloud, Cryptography, Access control, Data Privacy, Authorization

ACKNOWLEDGMENTS

I take this opportunity to express my sincere gratitude to my supervisor, Dr. Indika Perera, for his invaluable thoughts, encouragement, supervision, and guidance throughout this research work. He offered his generous guidance every time I reached him especially when I was stuck with my personal matters.

Further, I use this opportunity to thank Prof. Gihan Dias, for his thoughts, encouragement, supervision, and guidance throughout the initial phase of this research.

Further, I use this opportunity to thank my MSc lecturers who guided me throughout my MSc (Computer Science) course and shared their valuable knowledge with dedication.

Finally, I express my gratitude to my parents, my wife, and my friends for the support and encouragement throughout my life to bring me up to this level.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

QOS	Quality of Service
GB	Giga Byte
KB	Kilo Byte
Db	Database
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
CE	Convergent Encryption
AES	Advanced Encryption System
CBC	Cipher Block Chaining
ECB	Electronic Code Book

CHAPTER 1

INTRODUCTION

1.1. Background

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".[6]

Cloud computing model is one of the most successful computing service models which was introduced in the last decades. It has become an emerging trend in the IT sector and most of the companies are moving their architecture towards Cloud Computing paradigm because of its easy usage, high availability, cost efficiency, and many other advantages. Cloud computing paradigm became famous in October 2007 when IBM and Google announced collaboration [7] and currently it is one of the hottest topics and a rapidly developing area in the vast field of Information Technology. The National Institute of Standards and Technology (NIST) has provided a definition for cloud computing and mentioned in the preface of the chapter.

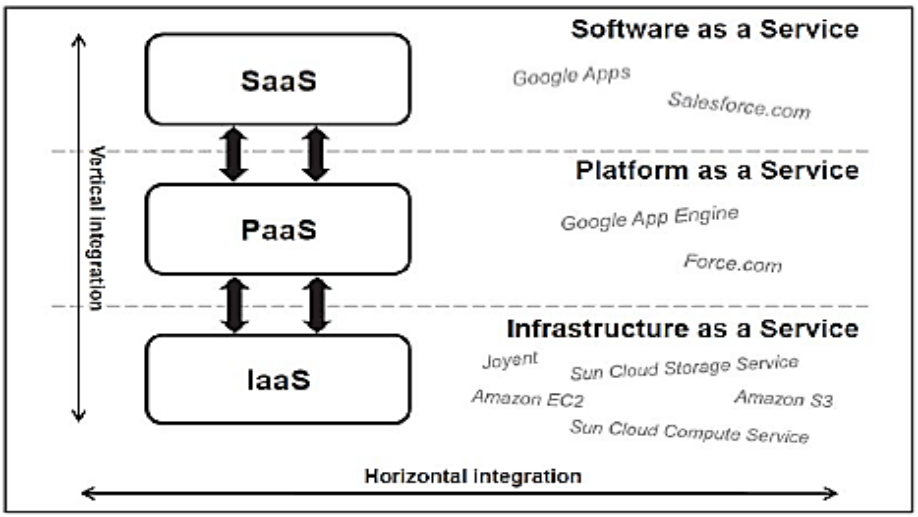

Figure 1 : The Three layers of Cloud Computing: SaaS, PaaS and IaaS
Source: [5]

Cloud computing paradigm is an extension of grid computing, distributed computing, and parallel computing. Therefore some of the main characteristics of cloud computing are (a) virtualization, (b) distribution, (c) On-demand self-service, (d) Broad network access, (e) Resource pooling(Multi-Tenancy), (f) Rapid elasticity, (g) Measured service and (h) dynamically extendibility [8, 9].

Mell & Grance [6] have proposed three types of service models that could be available in the cloud computing; They are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Overview of the three service layers and sample products are shown in

Figure 1. Based on those three service models, Cloud services make the IT operations' life easier and eliminates the need for local data centers and server infrastructures hence reducing the operational and maintenance cost.

One of the primary uses of cloud computing is for data storage[10]. Cloud storage provides services on both Infrastructure as a Service (IaaS) and Software as a Service (SaaS). Currently, there is a trend on Storage as a Service as well for cloud storage. Cloud storages stores data on multiple third-party server farms, rather than on the dedicated servers as used in traditional networked data storage. When storing data in the cloud, the user experiences virtual storage and it appears as the data is stored in a specific place with a specific name. But the actual implementation is different; it is a combination of various processes such as Compressing, Encrypting, De-duplicating, backing-up data and etc.

According to Vangie Beal [11], there are four types of cloud storages; they are Personal cloud storage, Public cloud storage, Private cloud storage, and Hybrid cloud storage. Personal cloud storage is a subset of public cloud storage that applies to store an individual's data in the cloud and providing the individual with access to the data from anywhere. It also provides data syncing and sharing capabilities across multiple devices. Public cloud storage is where the enterprise and storage service provider are separate and data are stored in the outside of the enterprise's data center. The cloud storage provider fully manages the enterprise's public cloud storage. Private Cloud storage is a form of cloud storage where the enterprise and cloud storage provider is

integrated into the enterprise's data center. In private cloud storage, the storage provider has the infrastructure in the enterprise's data center that is typically managed by the storage provider. Private cloud storage helps resolve the potential for security and performance concerns while still offering the advantages of cloud storage. Hybrid cloud storage is a combination of public and private cloud storage where some critical data resides in the enterprise's private cloud while other data is stored and accessible from a public cloud storage provider.

Apart from the personal cloud storages, many enterprises are tended to move their data storages to Public Cloud storage. In Microsoft research[12], they have identified four main advantages to move data to the public cloud. Those are; Strong consistency, Global & scalable storage space, proper disaster recovery, and Multi-tenancy & low-cost storage service.

1.2. Motivation

With the growth of the data produced in corporate environments, cloud storage systems are becoming attractive due to their accessibility and low cost. A recent survey by Gartner[13] shows that data growth forms a higher cost for hardware infrastructure in the data center. Data de-duplication has been performed by commercial cloud storage services such as Google Drive, Dropbox, and Bitcasa across users to save space. However, concerns on Cloud data security is the main obstacle to preventing many users to migrate into the cloud. In order to achieve data security, the foremost solution is to encrypt the data before leaving out from the corporate network. Even though this is good from the security perspective, it limits the other cloud-related functionalities such as space/bandwidth saving functionalities. Compression and the de-duplication are the main functions used to improve the storage efficiency and high compression ratio and deduplication ratio allow optimal usage of the resource of the cloud storage provider and consequently lower cost for the users[14].

Data deduplication is the process which a Cloud storage provider only stores a single copy of a file or data block that is owned by several users. In the industry there are four strategies for deduplication; depending on where the deduplication perform: client side or server side deduplication, and which level deduplication happens: block level or file level deduplication. Generally, the client side deduplication is more useful as it also saves data upload bandwidth. According to the Stanek, et al. [14], deduplication is a critical factor for a number of popular and successful storage services (e.g. Dropbox, Memopal) that offer cheap, remote storage to the public by performing client-side deduplication, thus saving both the network bandwidth and the storage costs associated with processing the same content multiple times. We can ensure that the data deduplication is the main actor for providing low-cost services on cloud storage and cloud backup services.

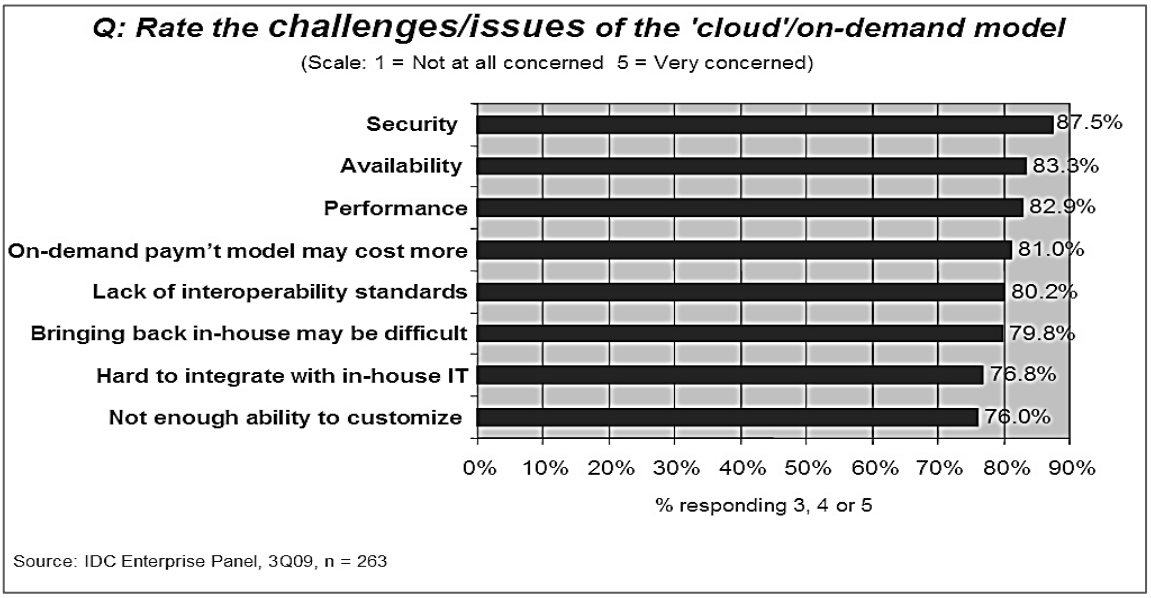

Figure 2 : Challenges for Cloud computing
Source: [2]

Based on the corporate policies and the legal regulations, cloud storage users need to encrypt their data before storing in the cloud. Usually, the user encrypts the file using the user's private key before upload it to the cloud storage. However common encryption methods used in the industry are randomized and it makes cloud

storage de-duplication impossible because cloud sees the ciphertext regardless of the actual data. Therefore end to end encryption using general encryption methodology is not suitable for deduplication. According to Li, et al. [15] we can use convergent encryption methodology to eliminate this issue. Convergent encryption method generates identical ciphertext for identical plain text using different keys. However convergent encryption in cloud storage is also vulnerable to offline brute-force dictionary attack[16].

According to Figure 2, Security and the cost for usage in cloud resource have become one of the major challenges on moving to the cloud. Therefore it is identified that there is a prominent issue when using public cloud storage for corporate clients, regarding the data privacy and the security while assuring the efficient use of cloud storage with low cost.

1.3. Objectives

- The main objective of this research is to develop a solution for corporates or individuals to use public cloud storages such as Dropbox to store mission critical/ sensitive data without worrying about security.

- This solution will ensure the privacy and the security of the data uploaded to the cloud storage by using data encryption.

- This solution will allow customers to enjoy the benefits of efficient usage of the cloud storage by implementing data de-duplication methodology

- Users will be able to upload data files through the proposing application and application will handle the deduplication and encryption before uploading them to cloud storage. Only the encrypted data will be stored in the cloud and it will ensure the privacy of the data.

- This application is independent of the cloud storage service. Therefore, this application will allow customers to use any type of public cloud storages without worrying about the security of cloud platforms. All the encryption and de-duplication is independent of the cloud storage service.

- Users will be able to retrieve back the uploaded data in the cloud storage through the proposed application. It will get the encrypted chunks from the cloud storage and build the complete data file for the customer.

1.4. Blueprint of Proposed Solution

Figure 3 depicts the blueprint design of the proposed application.

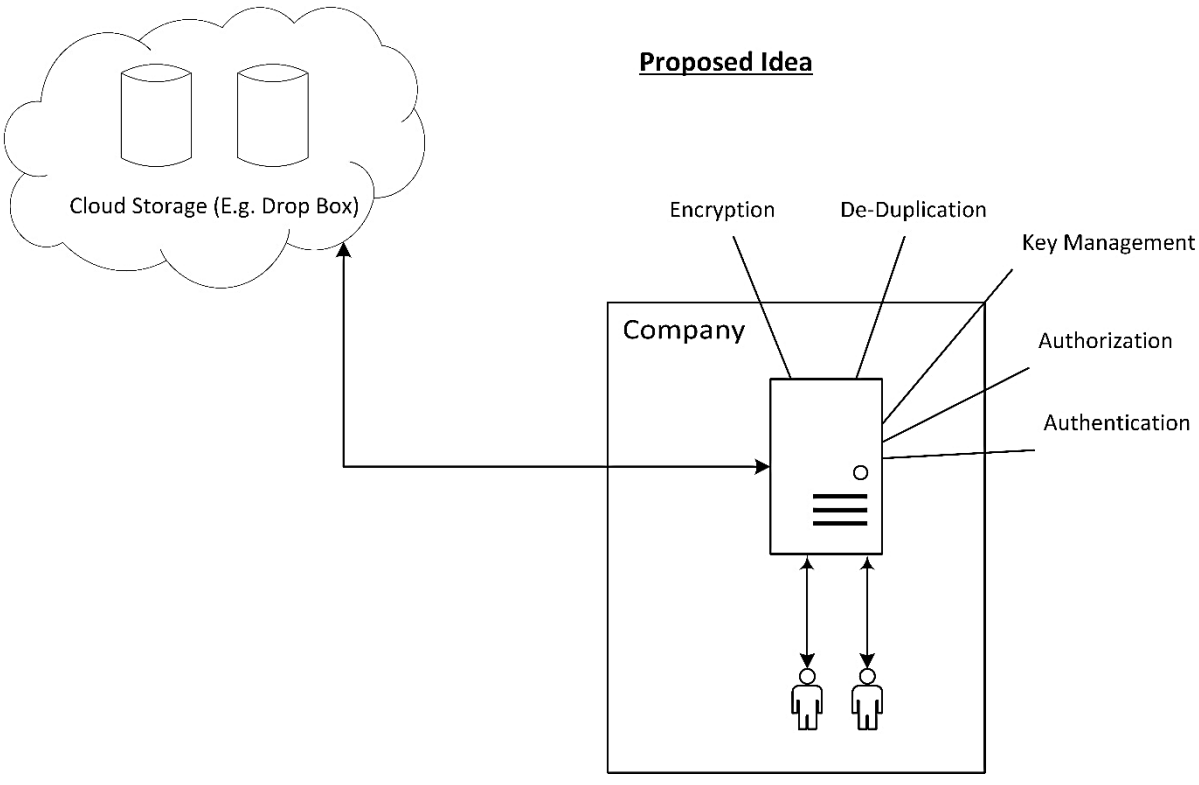

Figure 3 : Blue print of the proposed solution

1.5. Overview of the Dissertation

The rest of this report is organized as follows. Chapter 2 discusses the existing literature related to Cloud storage security and data deduplication. Chapter 3 presents the design and implementation of related details of the system. Chapter 4 presents the implementation details of the proposed system. Chapter 5 presents the evaluation conducted and results. Chapter 6 discusses how this proposed system addresses the problems discovered during the literature review. Finally, Chapter 7 represents how this system can be further extended and what could be the potential improvements.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

The main objective of this project is to build a software system that communicates with public cloud storage and it should capable of encrypting, decrypting and de-duplication of data upload to the cloud storage. During the communication process, the intermediate application performs authentication and authorization with the cloud. Further, it is focused only on encryption & decryption methods which support for deduplication.

We reviewed several related problems addressed by previous work, as well as their approaches to the problem is considered. Several ideas from these prior work formed the basis of this research. We discuss these ideas under the following major class of problems in an attempt to place our work in the context of previous studies. Mainly this research focuses on Cloud storage security and encryption and Data de-duplication.

2.2. Data Deduplication

2.2.1. Deduplication approaches

De-duplication[15] is a process of identifying redundancy in data content and denying this incoming data if it matches an existing record. Hence, only a unique single copy of the data is stored and will be made available to all the authorized users. According to the Harnik, et al. [17] data deduplication strategies can be categorized based on the data units they handle. There are two main data deduplication strategies based on the data units: (1) File-level deduplication, in which only a single copy of each file is stored. If there are multiple identical files, the additional file will be omitted. (2) Block-level deduplication, which segments files into blocks and stores only a single copy of each block and deduplication process executes on the block level. The system could either use fixed-sized blocks or variable-sized chunks.

There are another two deduplication strategies based on the approach. Those are; (1) target-based approach, in which the deduplication is handled by the target data storage device or service, while the client is unaware of any deduplication that might occur. This approach improves storage utilization but does not save bandwidth. (2) Source-based approach, in which the deduplication perform at the client side before it is uploaded. The advantage of this approach is that it improves both storage and bandwidth utilization. While data deduplication at the client side can achieve bandwidth savings, unfortunately, it is prone to side-channel attack [17].

2.2.2. Cross-User Deduplication

Based on the research done by Harnik, et al. [17], it is identified that the Cross-user deduplication in the cloud storage as a crucial point that is vulnerable to attacks. In the process of cross-user deduplication, each file or block is compared with data of all the other users in the cloud storage. Even though this method has advantages, it can be used as a side channel which reveals information about the contents of files of other users. This is a major concern for data privacy on cloud storage. It is mentioned that even the major service providers such as Dropbox, Mozy, and Memopal also perform cross-user deduplication. In the research they have proven how the cross-user deduplication approach allows attackers to identify files in the cloud storage, Learning the content of the files and how to create a covert channel for attacks. In order to overcome those security issues, Harnik, et al. [17] has discussed the implementation with the convergent encryption. But it still vulnerable to attacks because the attackers are still able to identify the occurrence of deduplication. The solution that they are suggesting is to perform deduplication on both target and source-based approaches using a random logic. However, it uses the convergent encryption which makes the key management additional overhead and also the bandwidth saving is not considerable.

Rashid, et al. [18] proposed a framework that implements block-level data de-duplication in which files are divided into blocks and de-duplicated. In this solution, in order to fully utilize the benefit of data de-duplication, cross-user de-duplication is used in practice. It identifies redundant data across different users and then removes the redundancy and therefore saving storage space. The authors also pointed out that an average of 60% of data can be de-duplicated for an individual using a cross-user deduplication technique. Thus, proving that data de-duplication is capable of supporting the integration with cloud storage to provide space-efficient storage on a lower cost and bandwidth consumption. However, there are several security drawbacks in data deduplication such as the issue of data privacy and integrity. Deduplication cross users can potentially lead to information leakage to malicious users through side channel attacks.

2.2.3. Data Deduplication on distributed Storage

Storer, et al. [4] Introduced a solution that provides both data security and space efficiency in single-server storage and distributed storage systems. Even this is not specific for the cloud environment, the solution is applicable for distributed storage systems. In this solution Encryption keys are generated in a consistent manner from the chunk data; thus, identical chunks will always encrypt to the same ciphertext, which is also called convergent encryption.

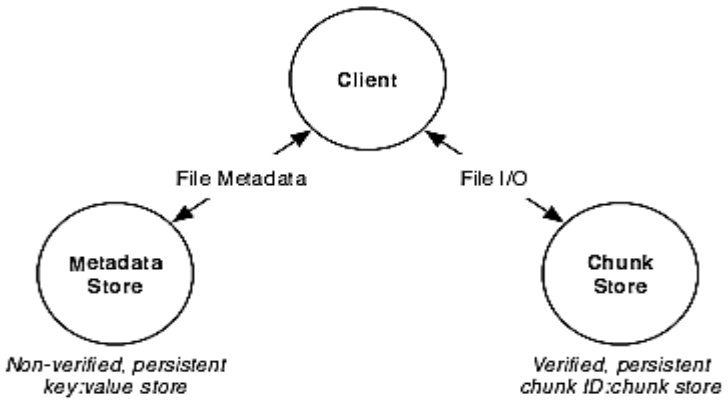

Figure 4 : Three primary players in the storage model [4]

Furthermore, the keys cannot be deduced from the encrypted chunk data. Since the information each user needs to access and decrypt the chunks that make up a file is encrypted using a key known only to the user, even a full compromise of the system cannot reveal which chunks are used by which users.

According to Storer, et al. [4], there are three primary players in the solution as shown in figure 4. Users interact with the system through the client, which is the starting point for both upload and download. It is the central contact point between the other components in the storage model. The metadata store is responsible for maintaining the information that users require in order to rebuild files from chunks, such as maps and encryption keys. We model this persistent storage using a simple, key: value architecture. In such a system, when the user submits a key: value pair to the metadata server. The role of the third player, the chunk store, is to persistently store data chunks and to fulfill requests for chunks based on their ID. The chunk store is also modeled as a key: value store, however, unlike the metadata store, the chunk store must be able to verify the correctness of the key with regards to the value.

In this solution, both file chunking and encryption occur on the client component. Encryption is done using convergent encryption, so that deduplication make easier. There are a number of benefits to performing these tasks on the client. First, it reduces the amount of processing that must occur on the server. Second, by encrypting chunks on the client, data is never sent in the clear, reducing the effectiveness of many passive, external attacks. Third, a privileged, malicious insider would not have access to the data's plaintext because the server does not need to hold the encryption keys. This solution has not handled the data compression for space saving, because the client cannot access encrypted chunks.

2.2.4. Data Deduplication on Cloud

Anderson and Zhang [1] have proposed an algorithm that can use to back up the laptop data to the cloud environment. This implementation also used the convergent encryption, and generate a key for each data block. Then the generated key is used as the index for storing data blocks. Using that key, any attempts to store multiple copies of the same block will be detected immediately. But in this research also it is mentioned that there is considerable overhead on maintaining the keys.

Figure 5 shows the proposed architecture of the backup system. Our main focus related to current research is on the backup module component. The backup module is handling data compression and data encryption (convergent). Data compression prior to encryption allows further reduce the size of data upload and this will save the data bandwidth. However, the data uploaded to the cloud is vulnerable to attacks because the data blocks are encrypted using a deterministic approach.

According to the Rahumed, et al. [19] research, They also introduced an idea of secured cloud storage backup system. In this implementation, they just use the hash function of the data block to check duplicates. From a security perspective, there are lots of disadvantages.

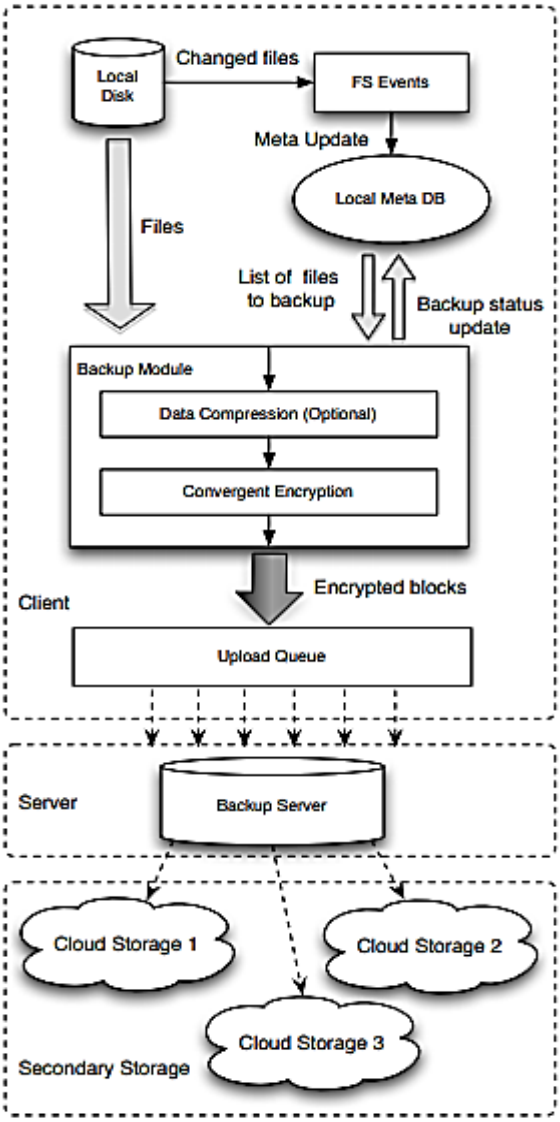

Figure 5 : Proposed architecture of backup system [1]

"ClouDedup" [3] is another solution proposed to perform secure deduplication with encrypted data for cloud storage. It provides a secure and efficient storage service which assures block-level deduplication and data confidentiality at the same time. It is also based on convergent encryption. This scheme consists of two basic components: a server that is in charge of access control and that achieves the main protection against attacks; another component, named as metadata manager (MM), is in charge of the actual deduplication and key management operations. Figure 6 shows the high-level view of the proposing system.

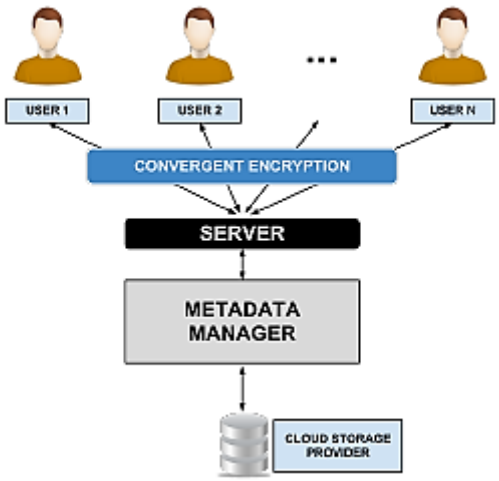

Figure 6 : High level view of the ClouDedup [3]

The server provides a simple solution to prevent the attacks against convergent encryption (CE) consists of encrypting the ciphertexts resulting from CE with another encryption algorithm using the same keying material for all input. The server has three main roles: authenticating users during the storage/retrieval request, performing access control by verifying block signatures embedded in the data, encrypting decrypting data traveling from users to the cloud and vice versa.

Metadata Manager (MM) is the component responsible for storing metadata, which includes encrypted keys and block signatures, and handling deduplication. When data upload MM checks if that block has already been stored by computing its hash value and comparing it to the ones already stored. User interaction in the ClouDedup has shown in Figure 7.

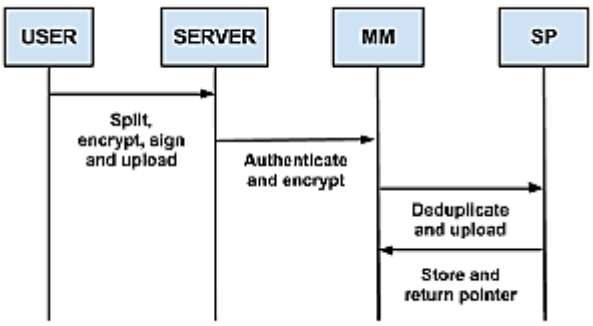

Figure 7 : Data storage protocol [3]

The main advantage in ClouDedup is the data are no longer vulnerable to convergent encryption weakness. The server takes care of adding an additional layer of encryption to the data (blocks, keys, and signatures) uploaded by users. Before being forwarded to MM, data are further encrypted in order to prevent MM and any other component from performing dictionary attacks and exploiting the well-known weaknesses of convergent encryption. During file retrieval, blocks are decrypted and the server verifies the signature of each block with the user's public key. If the verification process fails, blocks are not delivered to the requesting user.

2.3. Cloud Storage Security & Data encryption

Since this research is mainly focused on data de-duplication, traditional encryption methods are not suitable at all. Traditional encryption requires different users to encrypt their data with their own keys which convert identical plain text into different ciphertext and making deduplication impossible.

Kamara and Lauter [20] have provided a review of recent advances in the cryptography of cloud storages. They have discussed the six main advantages in cryptographic storage systems; (a) Regulatory compliance – support to law enforcement of data security (b) Geographic Restrictions – low restriction based on the storage location not affected (c) Subpoenas – secured from legal requests (d) Security breaches – Secured from the attacks (e) Electronic discovery – Integrity is verifiable (f) Data retention and destruction - can be easily managed with the master key.

Further, that research has discussed two encryption methods used in the cloud storages. Those two are Searchable Encryption and Attribute-based Encryption. At a high level, a searchable encryption scheme provides a way to "encrypt" a search index so that its contents are hidden except to a party that is given appropriate tokens.

Using a searchable encryption scheme, the index is encrypted in such a way that (1) given a token for a keyword one can retrieve pointers to the encrypted files that contain the keyword; and (2) without a token the contents of the index are hidden. There are many types of searchable encryption schemes, each one appropriate to particular application scenarios. For example, the data processors in our consumer and small enterprise architectures could be implemented using symmetric searchable encryption (SSE), while the data processors in the large enterprise architecture could be based on asymmetric searchable encryption (ASE).

2.3.1. Convergent Encryption

According to Li, et al. [15], convergent encryption is an option to enforce data security while implementing deduplication. It encrypts data copy with a convergent key, which is derived by computing the cryptographic hash value of the data copy itself[21]. After key generation and encryption, users retain the key and send the ciphertext to the cloud. Since the convergent encryption is a deterministic, identical data copies will generate the same convergent key and same ciphertext. This allows the cloud to perform deduplication on the ciphertext. Decryption can be performed with the convergent keys itself. In the convergent encryption, for each data block of the file, the convergent key is generated and user needs to store it for later use. According to Li, et al. [15], Key management in convergent encryption is an additional overhead to the application. Further, it is inefficient, as it generates an enormous number of convergent keys with an increasing number of users and files. This key management overhead becomes more prominent if we perform block-level deduplication.

As an example to show the level of overhead in key management, suppose that a user stores 1 TB of data with all unique blocks of size 4 KB each and that each convergent key is the hash value of SHA-256, which is used by Dropbox for deduplication. Then the total size of the keys will be 8 GB[15]. The number of keys is further multiplied by the number of users.

The resulting intensive key management overhead leads to the huge storage cost, as users must be billed for storing a large number of keys in the cloud under the pay-as-you-go model.

However, Li, et al. [15] has proposed a reliable convergent key management scheme for secure deduplication called “DeKey”. Dekey applies deduplication among convergent keys and distributes convergent key shares across multiple key servers while preserving semantic security of convergent keys and confidentiality of outsourced data. They have implemented Dekey using the “Ramp” secret sharing scheme and shows that it incurs small encoding/ decoding overhead compared to the network transmission overhead in the regular upload/ download operations.

According to the Chuan, et al. [13], this Convergent encryption scheme suffers from (1) Confirmation of File attack (CoF), where an attacker who has already known the full plain text of the data, he or she is able to verify if a copy of that file has already been stored. (2) Learn-the-Remaining-Information (LRI) attack, where the attackers already owned a big part of the original data and tried to guess the unknown parts by checking if the result of the encryption matches the observed ciphertext. And lastly, (3) Dictionary Attack, an attacker who is able to guess or predict the original file can easily derive the potential encryption key and verify whether the file is already stored in the cloud storage provider or not. In order to avoid the disadvantages of the convergent encryption, it is suggested that add random & unique secret value to the encryption key. But this will limit the effectiveness of data deduplication.

CHAPTER 3

METHODOLOGY

This chapter describes the context of experimental design and implementation. This is to discuss architectural and design aspects of the proposed solution in detail. First part of this chapter discusses the high-level architecture of the proposed secured proxy system for cloud storage with data- deduplication and the rationale for choosing that architectural model. The latter part of this chapter discusses how this proposed architecture was converted to the design and the justification for choosing the different technological components in the system design.

3.1. Requirement Elicitation Process

In order to finalize the proper solution, requirement elicitation process plays a major role. Though there are lots of requirement identification methods, this research has used only a few methods such as Literature Review, Naturalistic observation and introspect & personal experience.

3.1.1. Literature review

A literature review was conducted to discover the existing products and state-of-art techniques used to secure the public cloud storages with data deduplication. The author gained a wide and solid understanding of the problem domain and the solutions proposed by various researchers so far. Therefore it could be utilized to build a better solution based on the work done by others.

In other words, this technique is more familiar with the methodology, content, and conclusions of other researches and it can cover a huge area of requirements up to a large extent and can familiar with potential successful responses to a chosen problem which others have attempted and evaluated for their effectiveness. A literature review is considered as one of the most credible sources for requirement analysis for the proposing solution for public cloud storages.

3.1.2. Naturalistic observation

The author visited the selected general corporate users of the public cloud storages and observed their day-to-day tasks in its natural setting. Observation methods are truly helpful to understand how real users interact with cloud storage and their expectations. And it could be used to find out some details that simply did not come out of the other investigations.

3.1.3. Introspect and personal experience

This method has been used to gather information by using personal experience and knowledge on Public cloud storages to identify the requirements. Some demo applications were tried and techniques to identify the requirements that those applications have catered.

3.2. Previous solutions

Based on the literature review, few components were identified that makes a positive and negative impact on the public cloud storage usage for corporates. Those two areas are De-duplication and Security.

3.2.1. De-duplication

Facts related to data deduplication on existing systems have elaborated in following Table 1.

	Advantages	Disadvantages
File Level deduplication	Deduplication processing overhead is low.	Storage space saving is low
Block Level deduplication	Storage space saving ratio is higher	Deduplication processing overhead is high.
Source-Based deduplication	Network bandwidth will be saved	Vulnerable to attacks such as side channel attack
Target Based deduplication	Vulnerability to attacks is low.	Network bandwidth usage is high.

Table 1 : Facts for deduplication on previous solutions

3.2.2. Data Security

Facts related to storage security on existing systems have elaborated in following Table2.

	Advantage	Disadvantage
Non-deterministic encryption for the data, prior deduplication check	High secured	Storage saving from the deduplication is not applicable
Deterministic(Convergent) encryption for the data, prior deduplication check	Secured and storage saving from the deduplication is high	Still vulnerable for some attacks. Key management overhead is high.

	Advantage	Disadvantage
Non-deterministic encryption for the data, prior to storing on public cloud	High secured and high privacy	Cross user deduplication on cloud storage is not applicable
Deterministic(Convergent) encryption for the data, prior to storing on public cloud	-	Vulnerable to attacks on cloud storage. Key management overhead is high

Table 2 : Facts for security on previous solutions

3.3. Architectural Goals and Motivation

Proposing a secured proxy system to access storage with data-deduplication is specially designed for cloud computing architecture. Therefore it is required to concentrate and adhere to some main features of the cloud computing paradigm and those features was discussed by the Mell and Grance [9] and Zhang, et al. [8] on their researches. Therefore it has followed architectural best practices and Quality of Service (QoS) factors to produce a good architectural design for the proposed solution. Major goals are listed below in Table3.

Constraints	Description
Scalability	Cloud computing environment is highly scalable because of its usage increases day by day. Therefore, this proposing system also should be able to scale-up

Constraints	Description
	based on the usage and data volume with the minimum change of code.
Extendibility	The system should be designed; such that it should be able to add new features to the core system easily. This would support to add future enhancements.
Availability	The proposed solution should have 100% availability and failure of any component can lead to demote the purpose whole system.
Performance	Because of the cloud computing produce huge data sets, the proposed solution should have efficient algorithms for deduplication and retrieval.
Security	Cloud storage system is mainly concern about security and privacy. Therefore the proposing system should be able to store data in the public cloud without compromising the security and privacy of the data.

Table 3 : Feature of the system that should match to the cloud

3.4. Proposing Solution

Basic architectural overview of the proposing secured system to access the public cloud with data optimization is shown in Figure 7. Proposing solution is applicable for the corporate environment which has a requirement to store and retrieve data in the public cloud storage. Even though the internal users can access the public

cloud directly, the proposing system provides separate internal application as a proxy to access the public cloud storage.

As shown in Figure 8, internal users directly access the new proxy server. The proxy server contains three main components; they are (a) Access Controller, (b) Data de-Duplicator and (c) Data Processor. Based on the internal user's request proxy server communicate with the public cloud storage.

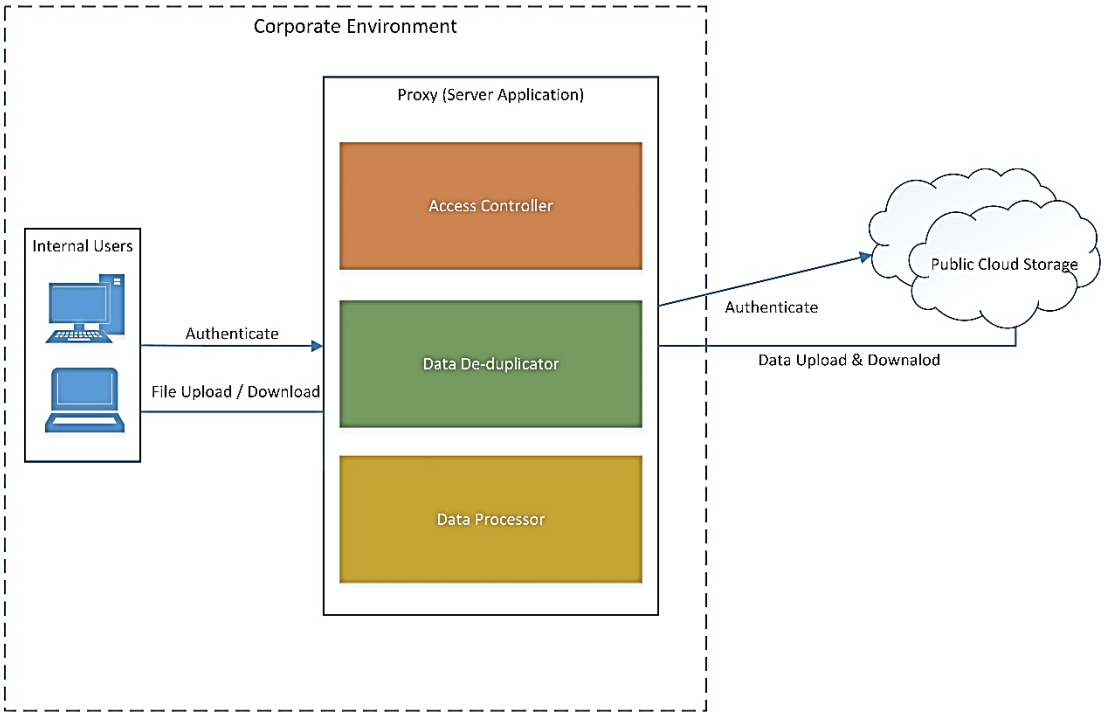

Figure 8 : Architectural overview of the Proposing system

3.4.1. Access Controller

The proxy server provides a web application to perform tasks related to public cloud storages such as Upload files, Download files, list all the files &, etc. This application is available only to the internal network and all the users should authenticate prior to performing any task. This access controller module handles user authentication and authorization. Any authentication method such as Kerberos, OAuth, OpenID & LDAP can be implemented to authenticate the internal users.

Further, it will track the file and its ownership. When the user requests a file, then this component validates the ownership of the file and show only the authorized files for that particular user.

3.4.2. Data De-duplicator

Data de-duplicator is one of the main components in the proposing solution. That will take an incoming file and perform block-level deduplication using the metadata database available in the application server itself.

3.4.3. Data Processor

Data processor performs a major role in both file upload and download. In file upload, once the de-duplicator eliminated the duplicate blocks, the rest of the data will be transferred to the data processor. Then the data processor will do the data compression and encryption. Encrypted blocks will be uploaded to the Public cloud storage.

3.5. Solution Architecture

3.5.1. User Authentication and Authorization

The proposing system is allowed to access only for permitted users in the internal corporate network. Proposing solution will handle the user management and any type of authentication method can be plugged-in to the system. In the current implementation, it has used a simple user name/ password authentication via https connection. User Authentication model is shown in Figure 9.

The proposed solution keeps track of the file level ownership. When a user uploads a file, Ownership is tracked against each file. That information is stored in the

SQL database. When a user requests back the files, based on the existing information, it will send only the authorized content only.

Even though there are multiple users in the system, for the proposed solution doesn't require to maintain a separate account in cloud storage for each user. Proposed solution maintain only one account for cloud storage, which is used to store data belongs to all the users in the company. From the user's perspective, its look like each user has an individual account in the cloud storage.

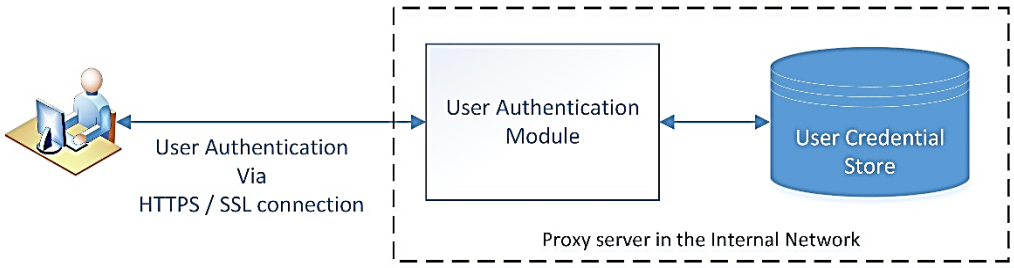

Figure 9 : User Authentication process

3.5.2. Data De-duplication

Data de-duplication is the main function of the proposing system. This proposing system uses target based & block level deduplication mechanism. So that deduplication is happened on the server application, not in the client machine. Once the user uploads a file to the server, it will break into the blocks in a fixed size. That block size is predefined in the system (Currently it is set as 2 Kb). Optimum block size depends on the average size of the file uploaded to the system. If the file size is too large and the block size is too small, then the number of blocks per each file is increasing and it will add additional overhead to the system; but that will increase the deduplication ratio. If the file size is small and block size is larger, then it will behave the same as the file level de-duplication and it will reduce the deduplication ratio.

Each block gets a unique id and it will be stored in the database along with it sequence within the file. Those IDs and sequences will be used to merge the blocks

together and regenerate the file for the user. In the deduplication process, the hash value for each block will be derived and stored.

When a new block arrived, the hash value of the new block will be compared with the existing hash values stored in the database. If there is matching hash value for the block, that block will be identified as a duplicate block and it will not be stored in the cloud storage. If none of the existing hash values matched with the new block, the hash value of the new block will be stored in the database and data block will be ready to upload to the Cloud storage. This deduplication process is shown in Figure 10.

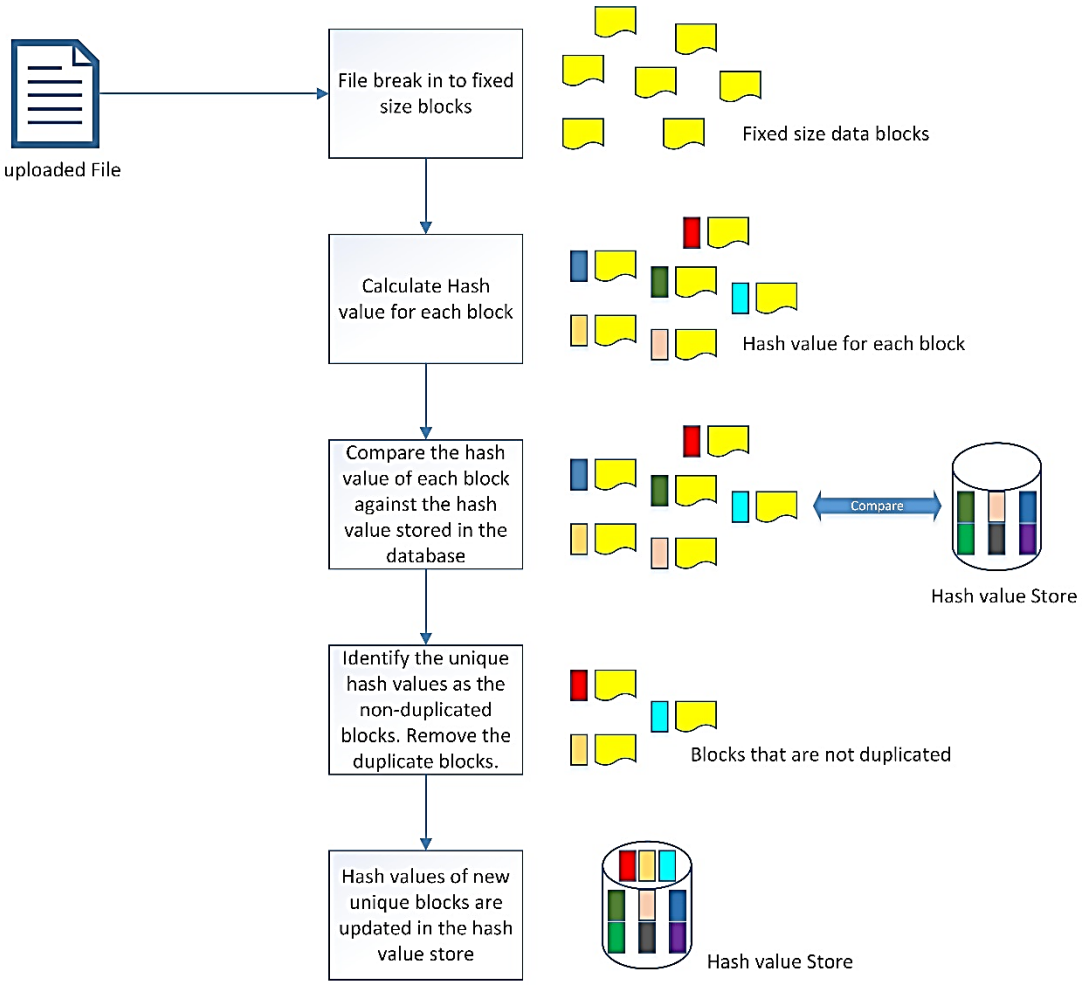

Figure 10 : Deduplication Process

This proposing architecture of the deduplication has reduced the additional key management overhead in convergent encryption as it in the other existing systems. Since this system used in the internal corporate environment, it is not necessary to encrypt data prior to deduplication.

3.5.3. Data Processing and Access Cloud storage

Data Processing module will perform two major tasks while data uploading. Those tasks are data compression and data encryption. Data compression is performed on the new data blocks prior to upload the cloud storage. That will helps to save the cloud storage space and reduce the cost. After the data compression, each data block will be encrypted using a symmetric encryption method. Then encrypted data block will be uploaded to the public cloud storage. Each block uploaded as a single file and block id use as the file name. Therefore, once the data uploaded to the cloud storage,

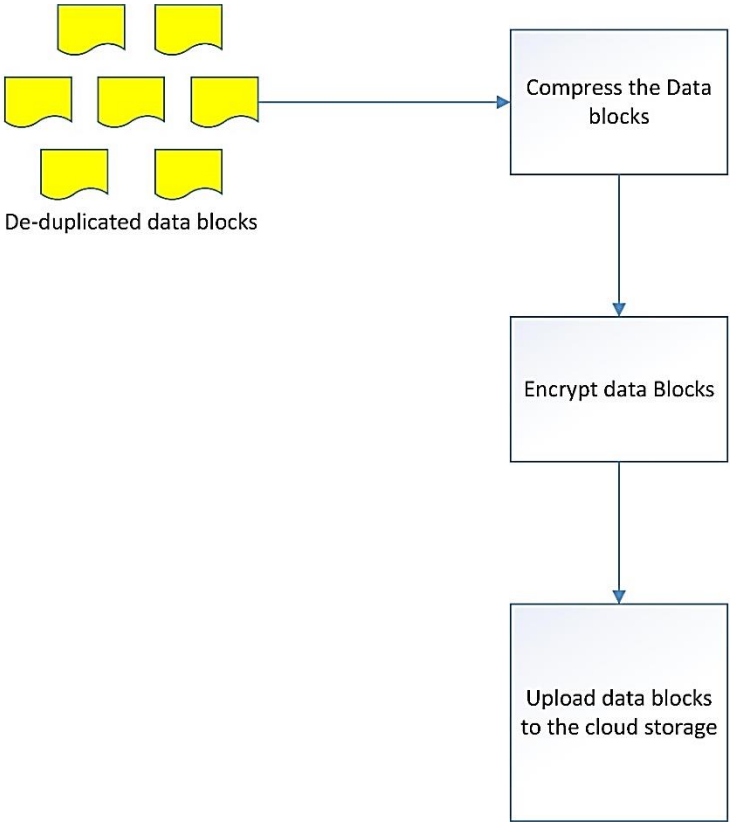

Figure 11 : Data Processing flow while data upload

all the data are encrypted and no way of relating each block to identify the single file content. Data Processing flow when data upload is shown in Figure 11.

Data processing component plays a major in data downloading as well. When a user requests a file to download, all the tasks such as Data fetching from the cloud, decrypting and merging are performed at the data processing component and had described further in the latter part of this chapter. Apart from that, Data processing unit is the only part that makes the connection with the public cloud storage. It keeps the corporate public cloud account credentials and makes a secured connection with the store directly.

3.6. Functionality

Current solution support three main functionalities; (a) File upload, (b) File Download and (c) List available files.

3.6.1. File Upload

The file upload process is shown in Figure 12. When an authenticated user uploads a file through a web interface, User Authentication module take the input stream and track the information required to file authorization such as file name, auto-generated file ID, user ID &, etc. After the user data tracking, Data De-duplicator module takes over the process.

As a first step, data de-duplicator module breaks the input file into fixed-size data blocks. That is because we are following block level deduplication; not the file level deduplication. Then each block is assigned to a unique ID and the hash value is generated for each data block. That hash value will be used to compare both new and existing blocks uploaded to the cloud storage in order to check for duplicates. Then

the proposing system will track the composition (how the file is made from the block) of the file using block ID. Those block related information will be stored in the database and the hash value of each block & the sequence number of the block within the file are also stored against the block ID. The system will compare the hash value of each block related to the newly uploaded file with the hash values in the database. If the hash value already exists in the database, that block will be identified as a duplicate block and that will not be uploaded to the cloud storage. Blocks which are related to non-duplicated has values make ready to upload to the cloud. Finally, the system will generate unique file names for each the non-duplicated upload ready data blocks and update the file name in the database against each block record.

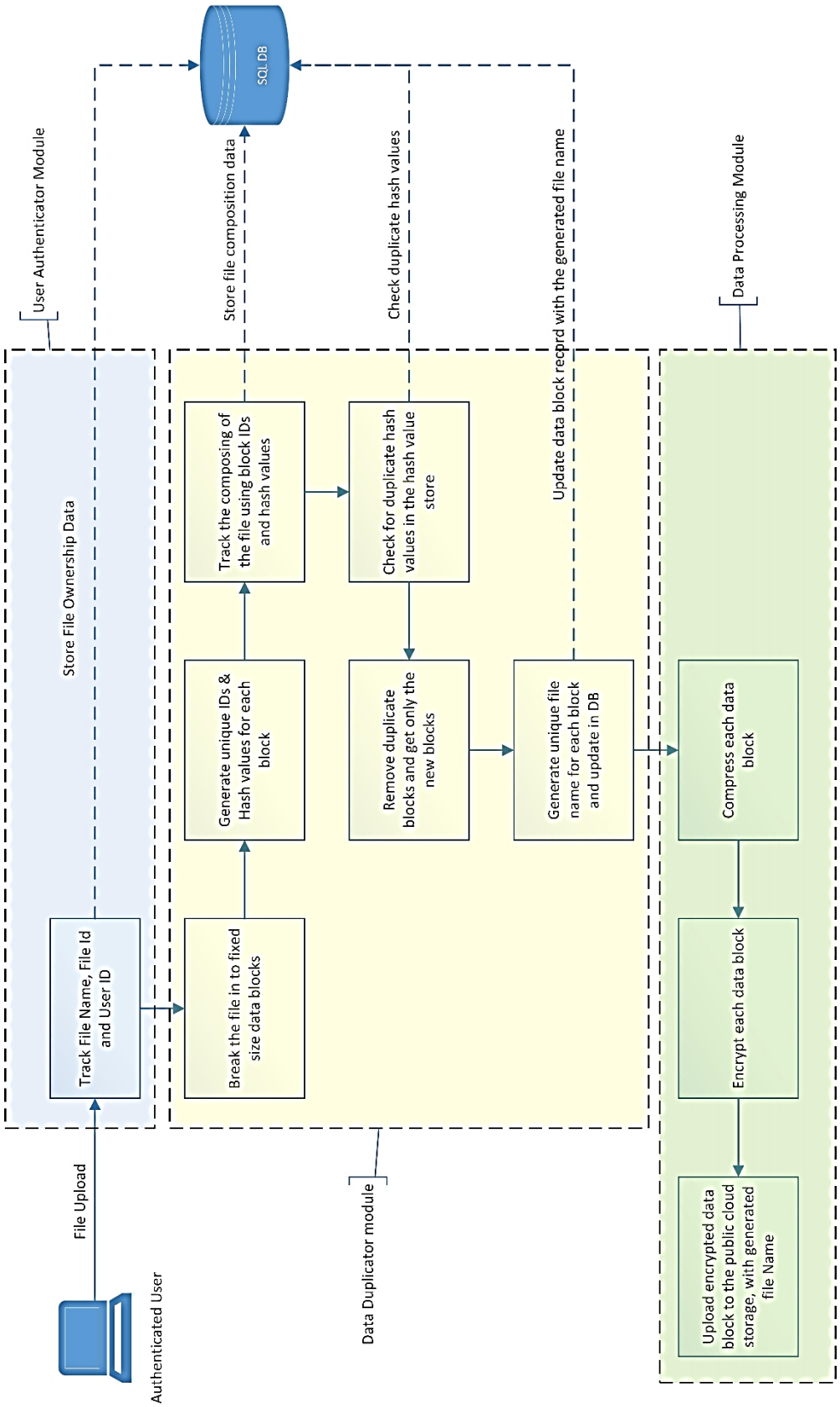

Figure 12 : File Upload Process

Then the process will take over by Data processing module of the system. First, it will compress all individual data blocks which are ready to upload. That will help to reduce the storage usage in the cloud. Then the compressed data blocks will be encrypted using a symmetric algorithm. Finally, encrypted data blocks will be uploaded to the public cloud storage using the generated file names against each data blocks. Single public cloud storage account will be used to upload all the data related to all the users in a corporate environment. That will help to eliminate purchasing individual accounts for each user in the corporate environment and it will have a huge cost saving. Cloud storage contains only the encrypted data blocks as files and the individual file doesn't have any meaning at all.

3.6.2. File Download

When an authenticated user requests a file for download, which will reach to the user authentication/authorization module. It will validate the user request and check the requested file access is authorized to the requested user. If the file is accessible to the user, then the request goes to the data processing module.

Initially, it will get the details of the data blocks which require for the file composition such as block ID, block name, Block sequence number &, etc. Then using those records, all the data blocks will be downloaded from the public cloud storage. Once the download is completed, it will decrypt all the data blocks and decompress after that. Finally, all the data blocks ordered according to the sequence number and merge into a one data stream and generate the file for the user. The detailed file download process is shown in Figure 13 below.

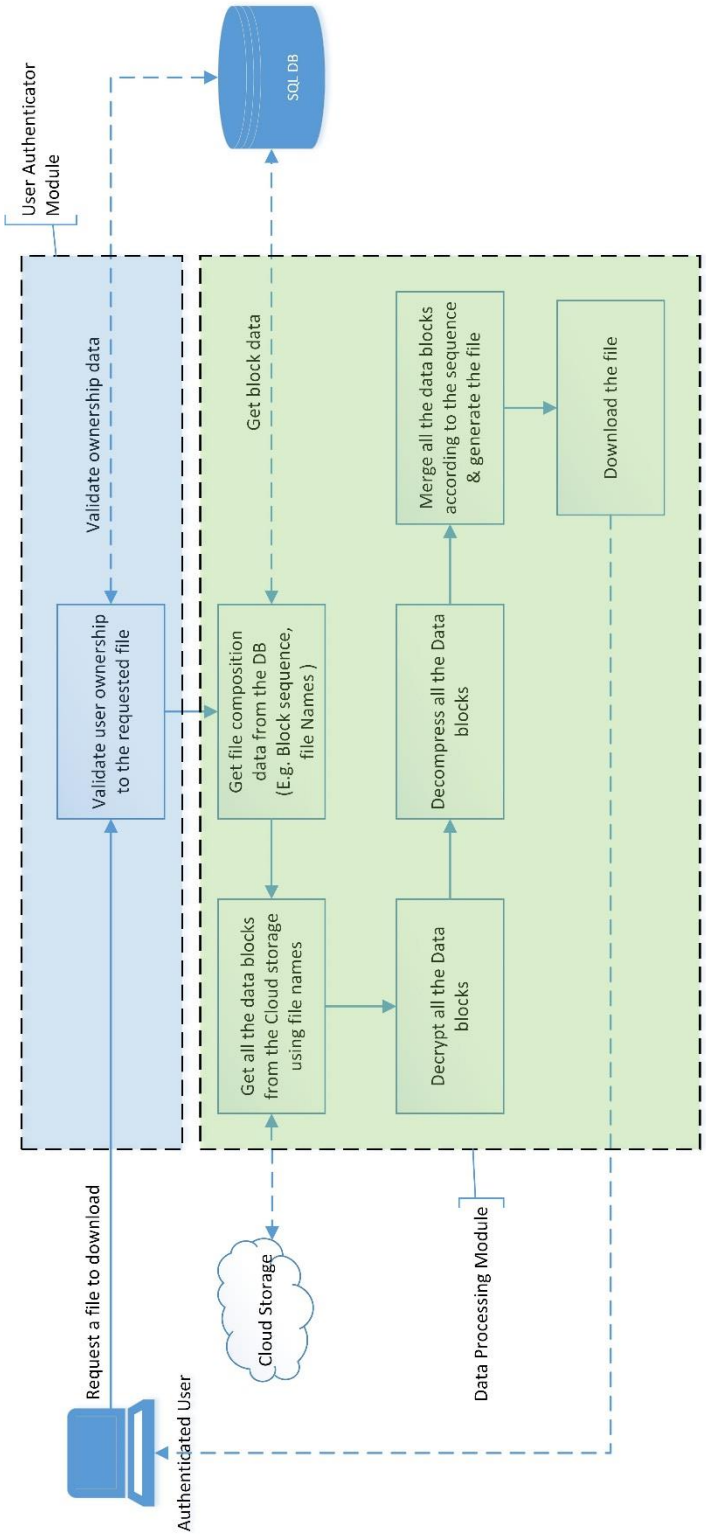

Figure 13 : File Download Process

3.6.3. List available files

When authenticated user request to list all the accessible files for their account, the request will handle only by the user authentication and authorization module. Proposing application database maintains the list of files and their ownership information. Using an internal database, the application will determine the list of files that are accessible to the user and will send those details to the user. In this process, the system will not communicate with the Cloud storage and communications happen within the internal network only.

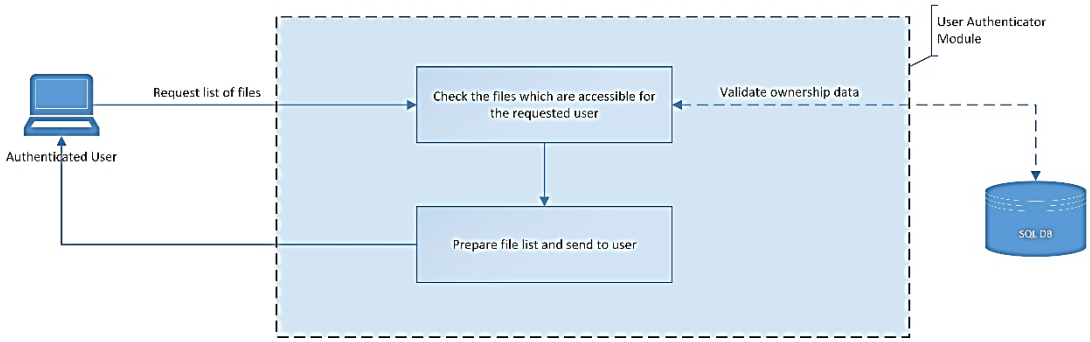

Figure 14 : List files process

3.7. System Component Architecture

Multi-tier architecture has been adapted to the development of the proposed secured storage solution for cloud computing. There are many advantages to use in the n-tier architecture such as flexibility, low dependency on technology, loosely coupled layers and etc. The component architecture is shown in Figure 15.

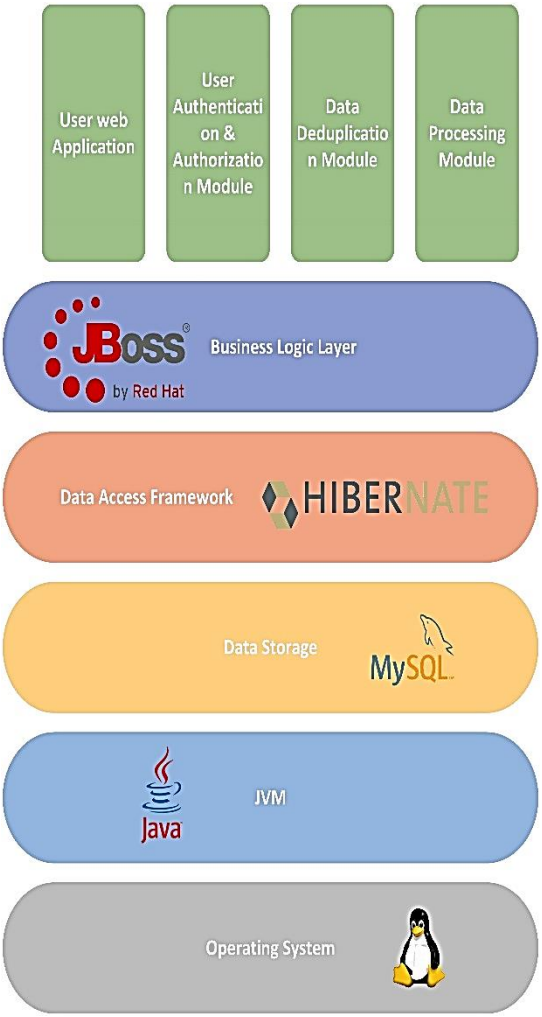

Figure 15 : Component Architecture

CHAPTER 4

IMPLEMENTATION

4.1. Implementation Rationale

Several Cutting-edge technologies are used in the proposed solution for cloud storage. Findings of the literature review chapter have elaborated these technologies and this section describes the rationale of selecting those technologies/methodologies and internals of them.

4.1.1. Public cloud storage – Dropbox

Dropbox is one of the most common cloud storage in the market. Known for its convenience and easy to use interface, Dropbox, in particular, has garnered more than 500 million users worldwide by May 2018. This is expected to increase up to 540 million by the end of the year 2018[22]. Dropbox has one of the top network storage services and it has since been providing personal data storage and data sharing among multiple users. Dropbox has good data sharing facility and those sharing methods are as follows: (a) Public Sharing, data is intended for the public, so there's no access control. A link to the shared folder (called the sharing URL) can be published, giving anyone on the Internet access to the shared documents. (b) Secret URL Sharing, file owner shares the data with others by sending them a sharing URL generated by the cloud storage provider. Anyone with this URL can access the data without further authentication or authorization. The data owner is responsible for identifying the URL receivers. This is only applicable to shared files and not folders. (c) Private Sharing, file owner must explicitly specify who can access the shared data. The cloud storage providers then authenticate the identity of the named users, usually by requesting that they sign into their account before accessing the data.

In October 2014, IT news giant, Engadget reported that Dropbox has been under massive hacks which compromises 7 million Dropbox accounts with their credential leaked online [23]. Despite Dropbox quickly implemented counter-measures to detect the account compromised, they are undeniably a single point of

failure in the face of all the sophisticated attacks online. Dropbox does not have control over how the secret URL links are shared after they are generated. This can lead to unauthorized re-sharing of the secret URL link. For instance, owner A sends a secret URL link to user B, B although is not capable of inviting others to view this file, but is capable to re-sharing this URL to others without the acknowledgment of the file owner. This would mean that if an unauthorized user got their hands on the URL, they can potentially access the shared file with the URL. The convenience of file sharing is performed on the cost of data confidentiality and privacy, therefore there is a need for data encryption and access control to overcome the vulnerabilities of Dropbox.

Even though the security of the Dropbox cloud storage is compromised, according to our implementation, it will not be an issue since we are storing encrypted blocks in the cloud. Even the attacker will not be able to read the content and unable to identify the relationship between data blocks to prepare the complete file. Further, Dropbox provides comprehensive developer API, which can be used to develop third-party applications to use Dropbox with full features.

4.1.2. Check Duplicates – SHA-512 Hashing

In order to check for duplicates, the proposing solution uses a hash function on each data blocks to do the comparison. A cryptographic hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but the output is always of fixed length. In the industry, there are few popular Hash functions such as MD5, SHA1, SHA256, SHA512 &, etc. The MD5 algorithm is the most widely used hash function. However, these hashes are not always unique and rarely there can be the same hash value for two different inputs. This is called “collision” and chances of collision in SHA family is less than MD5 and also SHA family algorithms generate more strong hashes than MD5.

Java has 4 implementations of the SHA algorithm. They generate the following length hashes in comparison to MD5 (128-bit hash). Table 4 depicts the comparison of SHA family algorithms.

	Length	Comparison
SHA - 1	160 bits	The simplest one. Stronger than the MD5
SHA - 256	256 bits	Stronger than SHA-1
SHA - 384	384 bits	Stronger than SHA-256
SHA - 512	512 bits	Stronger than SHA - 384

Table 4 : SHA Family Hash comparison

According to the findings, the SHA-512 algorithm would be one of the strongest hash algorithms in Java. Therefore it will use to generate the hash values of the file blocks that required to check for duplicates.

4.1.3. Compress/Decompress Data Blocks – Java Deflater

Java Deflater is actually a wrapper around the zlib library, which implements the deflate algorithm. Java deflator is a combination of two essential algorithms; (1) dictionary-based compression & (2) Huffman encoding. Dictionary-based compression works within a "window" of data. It looks at the upcoming data to be compressed and looks to see if it already encountered a matching sequence in the data recently processed. How exactly the compressor matches upcoming input against previous sequences is something that is up to the Deflater (in fact the zlib) implementation, and it turns out that it can be configured to some extent.

Specifically, we can configure a Deflator's compression level to indicate how far the deflator looks for matching sequences, resulting in a tradeoff between CPU and compression ratio.

After applying dictionary compression, Huffman encoding is applied to the output. Huffman encoding is one of the most famous types of compression system. Even if it is not used on its own, it is often used in conjunction with another compression scheme (as indeed in the case here). Huffman encoding works on a stream of bits. It takes an "alphabet" of symbols (for example, the possible "symbols" could be the bytes 0-255, but it could be some arbitrary range of numbers/values), and re-codes each symbol in the alphabet as a sequence of bits, so that the sequence corresponding to each symbol reflects the frequency of that symbol[24].

4.1.4. Encrypt / Decrypt Data Blocks – AES

AES stands for Advanced Encryption System and it’s the symmetric encryption algorithm that uses a single key known as a private key or secret key to encrypt and decrypt sensitive information. It is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES is a block cipher that encryption happens on fixed-length groups of bits. AES supports key lengths of 128, 192 and 256 bit. Every block goes through many cycles of transformation rounds. The important part is that the key length does not affect the block size but the number of repetitions of transformation rounds (128-bit key is 10 cycles, 256 bit is 14). In the proposing system, AES is using with CBC mode to encrypt a message as ECB mode is not semantically secure.

4.2. Database Design

Proposing solution uses simple DB design as a proof of concept. MySQL relational database has been used for the actual implementation of the system. Even though there are many tables used in the solution, Figure 16 depicts the ER diagram of the main tables used within the solution.

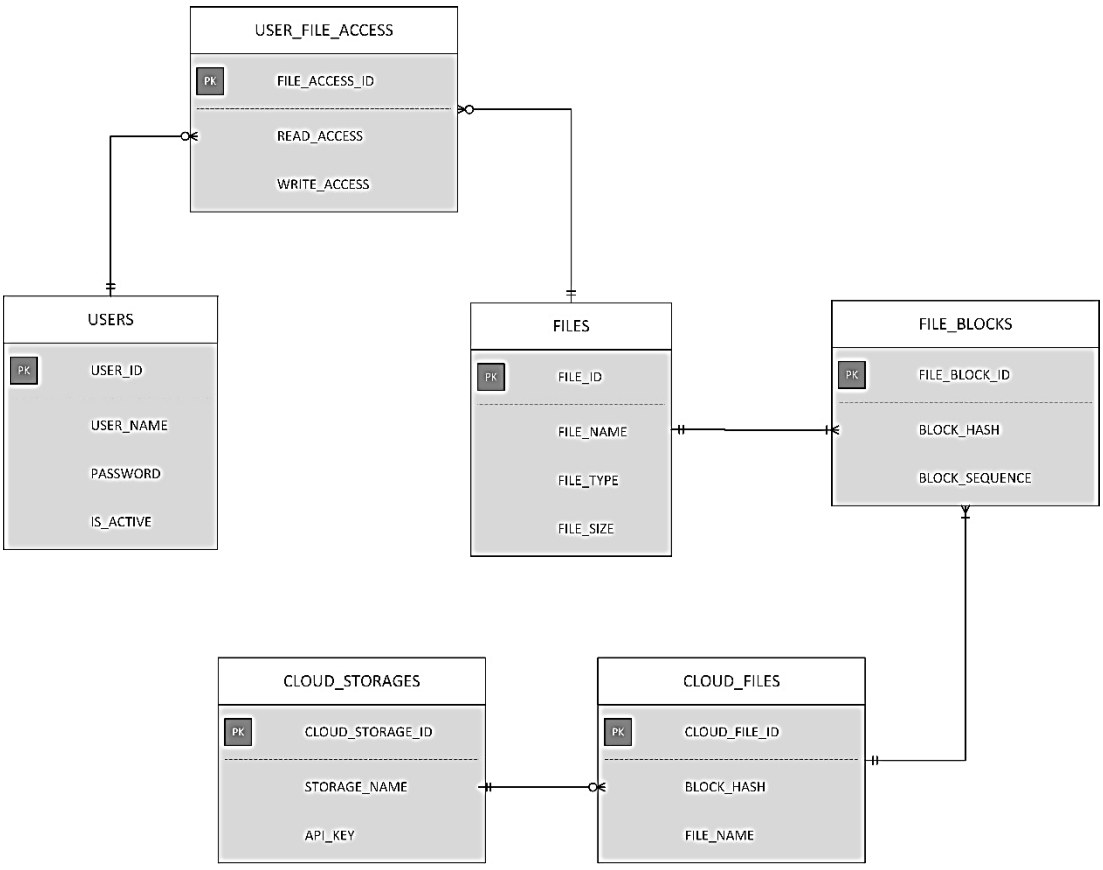

Figure 16 : ER diagram of the master tables of the solution

Mainly six tables involved in the solution. USER table contains the user related information. FILES tables contain information about the files that are uploaded to the system. USER_FILE_ACCESS table contains the file ownership details. Each USER can be related to many USER_FILE_ACCESS entries and each file can be related to multiple USER_FILE_ACCESS entries. This is because the system is designed with

the idea of file sharing with multiple users. FILE_BLOCKS table contains the details of file blocks that are broken out form the file. Therefore, Each FILE record is related to one or many FILE_BLOCKS records. Each FILE_BLOCKS record related only to one FILE record. CLOUD_FILES table contains the information of the blocks that are uploaded to the cloud storage. Each CLOUD_FILES record can be related to many FILE_BLOCKS records. Each FILE_BLOCKS should have only one CLOUD_FILES. This is because duplicate blocks won't be uploaded to the cloud storage two times.

4.3. Flow of Sequence

Data file upload to the public cloud storage and download files from the cloud are the main function of the proposing solution. Following Figure 17 depicts the sequence diagram of the file upload process. Implementation of this solution is done using the Java language.

For the file upload process, mainly eight java classes were in use. FileUploader, FileTracker, FileTrackeDAO, FileProcessor, BlockProcessor, FileBlockDao, DropBoxCleintManager and MySQLConnectionmanager are those classes involved in for implementation. File upload request initially received to the FileUploader class. Then it will create DB entry for the file tracking. After that FileProcessor class perform the rest of the upload tasks. It breaks the file into the blocks, generates hash value and stores the data block details in the Database. Then duplicate data blocks identified and compression and encryption performed on the unique data blocks. After that unique file name is generated for each block and update the database. Finally, non-duplicate data blocks uploaded to the public cloud storage in an encrypted format.

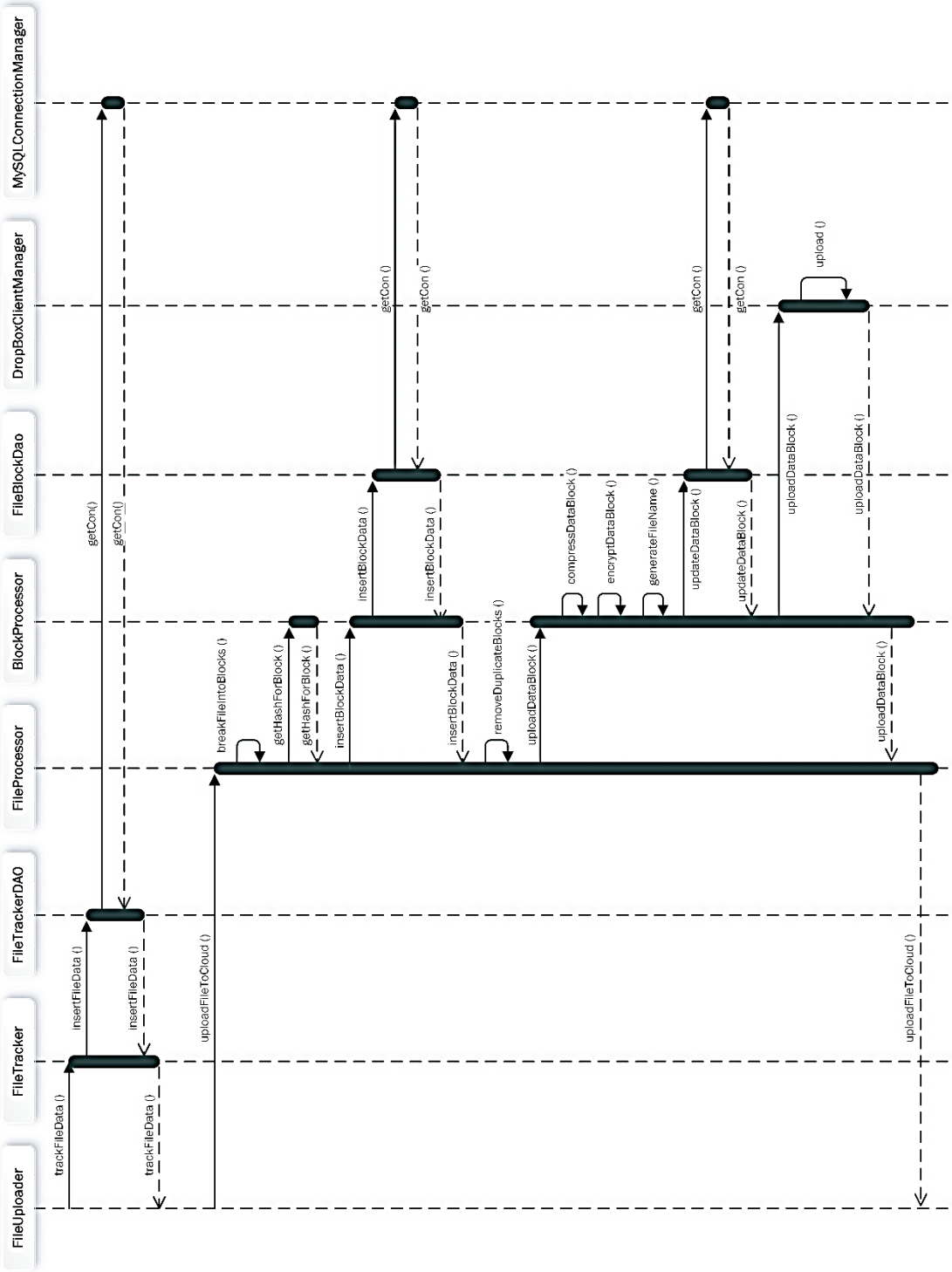

Figure 17 : Sequence diagram for file upload

CHAPTER 5

OBSERVATION RESULT &

EVALUATION

To assess the successful research project, it is important to carry out the proper critical evaluation. The complete project is needed to evaluate based on both qualitative and quantitative factors. This section elaborates on the evaluation process of the proposed secured system to access public cloud storages.

5.1. Evaluation Methodology

In order to assess the proposing solution; “Secured way to access the public cloud storage”, it is mainly performed an empirical evaluation on the solution. This evaluation is done in two steps.

1. Quantitative Evaluation

Quantitative analysis of the prototype was carried to ensure that the project implementation has met the required properties of the secured system. Quantitative evaluation was done based on the two methods of data gathering. They are

- Statistical analysis
- Existing research findings

2. Evaluation by the author

A self-evaluation to critically asses the status of the project is done by the author. In the self-evaluation, the main focus is on qualitative factors of the solution.

5.2. Evaluation Criteria

Evaluation of the proposing solution was conducted based on the pre-defined criteria and that has ensured all the aspects of the project were covered through the evaluation process. Those evaluation criteria are as follows.

- Validating the approach
- Algorithms and Techniques
- Strengths and weaknesses
- Future work and suggestion

5.3. Quantitative Evaluation of the prototype

Quantitative evaluation of the proposed solution is very crucial to identify the product suitability to work with the real-life environment. Quantitative analysis provides statistical data that can be used to take a definitive idea on the product flow. Quantitative evaluation of the proposed solution is conducted in the controlled environment and those statistical data can be changed in a different environment.

5.3.1. Cloud Storage Saving

Cloud storage saving is the most vital feature in the proposing framework for a secured public cloud storage solution. The solution has provided deduplication mechanism along with the data compression, in order to improve the cloud storage saving. Since single cloud storage account stores data belongs to all the users in the company, an efficient storage saving mechanism will give considerable cost saving to the company.

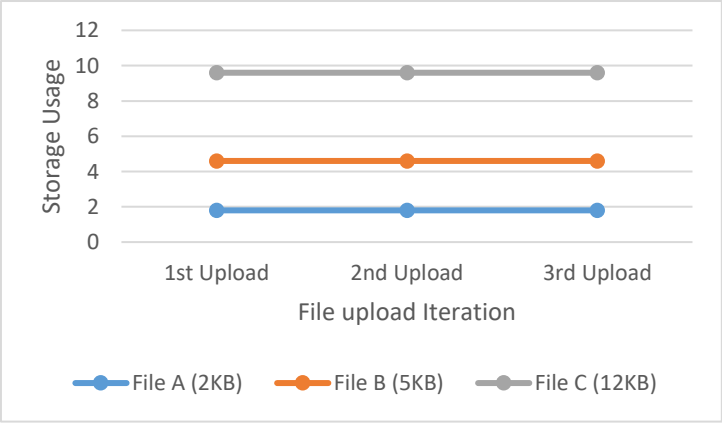

Figure 18 : storage usage when same file upload multiple times

Since the prototype solution breaks each file into 2KB data blocks, de-duplication check for 2kb data blocks. This block size is configurable and the changing block size should be a wise decision. If we increase the block size, the data duplication ratio will be reduced and storage saving becomes low. If we reduce the block size, then the deduplication ratio will be high and storage saving will be high. But the system will have additional overhead because of a number of files/blocks get increased.

Figure 18 depicts how cloud storage was used by three different file uploads. Three files are in three different sizes (2kb, 5 kb & 12 kb) and each file uploaded to the prototype system three times. According to the graph each file consumes the storage space only at the initial upload and in the 2nd and 3rd uploads haven't consumed any additional storage space. File C shows the additional storage saving because there are duplicate data blocks within the file itself. Therefore this will depict the deduplication process works as expected.

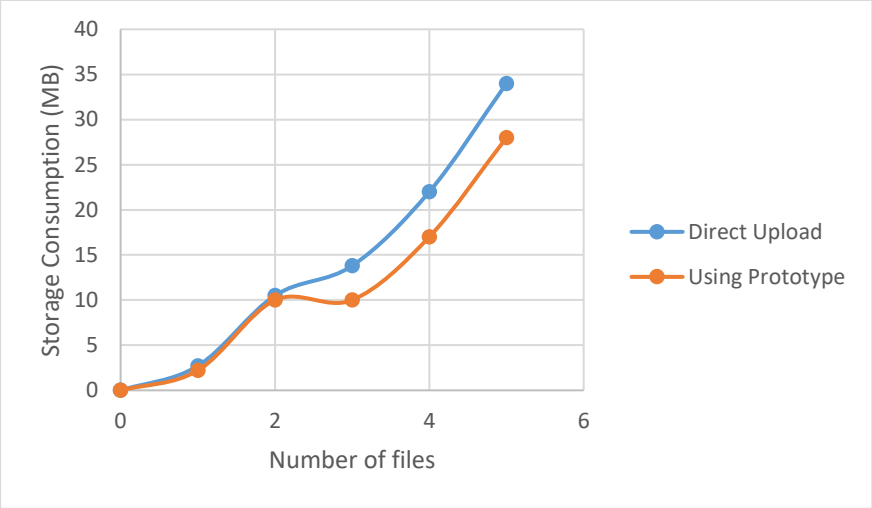

Figure 19 : Dropbox Storage in direct upload & using prototype

Figure 19 depicts how cloud storage usage when we upload multiple files. It compares how the storage use when data upload directly to the drop box and data upload via the prototype system to the dropbox. For this test scenario, we used a series of sample files which the initial content is common for all the files (such as cover page). In the graph, one line shows how storage grows when files uploaded to Dropbox

directly. The second line shows how storage grows when files uploaded to Dropbox via the proposed solution. It can clearly see that there is a considerable storage saving while using the prototype system than use the cloud storage directly.

5.3.2. File Upload Performance

When considering the corporate environment, system performance is a critical factor that always should consider. Even from the management perspective, it is required to have well-performing systems to get the effective output from workers of the company.

Following figure 20 shows how the prototype system performs under the various file upload scenarios. These stats were recorded using a single instance of the application server running on a general purpose machine. System specification of the machine that used to deploy the application is Intel Core i5-4300U CPU, 12GB RAM, and 150GB HDD. Since this system is mainly performing network data transfers, Network bandwidth is also a critical factor to decide the system performance. Current prototype implementation uses 4G LTE (20Mbps) network to communicate with public cloud storage (Dropbox).

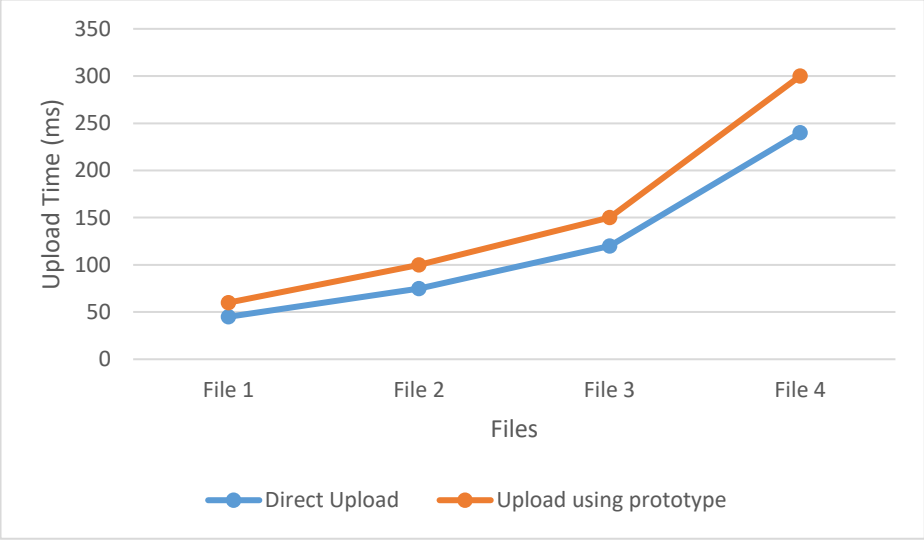

Figure 20 : File Upload Performance comparison

According to the statistics, it is identified that file uploading via prototype system is time-consuming than the direct upload. This can be caused because of additional processing performed in the prototype system before uploads such as deduplication, compression, and encryption. However, with the gain of storage saving and security benefits, the system performance will not be a significant disadvantage to the proposed solution. Further, the system performance can be get increased by improving the hardware specification and other methodologies discussed in the final evaluation.

5.3.3. Network Bandwidth Usage

The proposed solution uses the public network only to communicate with Cloud storage. When the data blocks are ready to upload to the cloud storage, those blocks have gone through several steps to optimize the number of blocks and block size. Deduplication process and the compression are handling the data block optimization. Once the deduplication and compression apply on the uploaded file, it is ensured that the size need to upload to the cloud has reduced than the actual file size.

According to the findings on the prototype application, network bandwidth usage for file upload is lesser in prototype application than the direct upload to Dropbox. This was measured using the tool called "Wireshark". Reduced network bandwidth usage is a huge benefit to the corporates as they can reduce the cost and able to allocate the rest of the network bandwidth for other purposes.

5.4. Discussion

The main objective of this research is to introduce a secure mechanism to store and retrieve business data in public cloud storages with data optimization facility. The motivation for the project idea is, most of the corporate policies are don't allow to store unencrypted data in the public storages and if the data is encrypted, deduplication

process already implemented in the cloud storages itself will not work. Therefore, the problem bounce between Security and space saving; If we focus on space saving, security will be compromised and if we focus on security, the system will not get benefited from the space saving.

As considering the overall architecture, proposing system is deployed in the company internal network and exposed to the public cloud storage only when data block upload & download process. Therefore, the system is only accessible to company users only. Since all the communications are done via HTTPS channel, all that data on the wire are well secured. Further, all the communications between the application server and cloud server also done via secured HTTPS. So it is not possible to perform any attack from the external parties.

In the proposing system, deduplication happens in the target server. Since this application used in the internal network, deduplication at target will not cost for network and it will not use the client machine resources as well. In the proposing solution, a duplicate is checked using the hash value of each block. As in the literature review, some researchers suggest using convergent encryption method to check for duplicates. If we use convergent encryption in our implementation, key management will be an additional overhead and data compression might not be possible. One of the disadvantages in the proposing solution is using SQL database for string hash keys. For small data sets, SQL database will be enough, but with the growth of the number of hash keys, the queries will not perform as expected.

After performing deduplication, the proposed system performs compression and encryption on each data block. Compression is an additional step taken to optimize cloud storage usage. This will give more saving and cost benefit to the company. Prior to upload each data block to the cloud storage, the prototype will encrypt each data block. This will give huge security benefit when they are stored in public storage. When the data blocks are stored in the cloud, attackers are not possible to break the encrypted data blocks. Since blocks are encrypted we don’t need to worry about even the trustworthiness of the Cloud storage providers as well.

Further, this will not support storage level deduplication and cross-user deduplication in the cloud storage level. Therefore attackers in cloud storage will not have a chance to learn or attack the data set that is in the out account. Further, cloud storage contains small data blocks and it doesn't contain the information on how to compose the file from combining data blocks.

So even if the cloud is security is compromised attackers don't have meaning by only having individual data blocks in the cloud without knowing the relationship on them. Those relationship data are stored in the SQL database inside the company network. As an overall, the author assumes the proposed solution has achieved almost all the objectives of the research and some of the identified plus & minus points shown in the table below.

Plus points	Minus points
Source-based deduplication.	Use a relational database to store Hash values of the data blocks.
Efficient deduplication using block level hash values.	
Compress data blocks for further storage saving.	
Store encrypted data blocks in cloud storage.	
All the data processing handles inside the corporate network and exposes only the encrypted blocks to the outside.	

Table 5: Plus & minus points in the Prototype Solution

CHAPTER 6

CONCLUSION

This research describes a secured & efficient solution to access public cloud storages with data deduplication and storage optimizing mechanisms. Based on the literature review various existing solutions have been identified and their pros and cons adopted when designing the solution. Most of the existing researches are based on the external solutions that always make additional performance overhead from various implementations such as convergent encryption, Searchable encryption, and key management. By analyzing all the ideas, comprehensive prototype solution has designed and implemented successfully. Since this solution works as proof of concept, various improvements can be applied in future enhancements and make it a comprehensive solution.

An overall research project is a notable success as it has achieved the aims & objectives and contributed massively to the author's knowledge and skill set. The thesis has provided a comprehensive account of how the approach was constructed from requirement elicitation to tools, technologies, and methodologies used in the design and implementation. Furthermore, issues, limitations, and challenges are also discussed in the course of this thesis. The author also specifies future work identified by the author as well as the expert evaluators at the end of the report. Therefore, the thesis will also serve as a valuable reference point for future research into secured systems to access public cloud storages.

The approach proposed by this project can be adopted in any domain. It will provide a highly scalable, reliable, efficient and secure approach to access the public cloud storage. This will bring a significant competitive advantage to consumers of such a system that allows enjoying the benefit of cloud computing while adhering to internal policies and procedures.

6.1. Future Enhancements

Due to time constraints, only the essential features of the prototype were implemented. This leaves a lot of room for future enhancements and expansions. Further, some future enhancements were identified in the project evaluation stage and they are as follows.

- Improve the system user authentication module to make it pluggable to any authentication mechanism used in a corporate environment.

- Identify and apply the proper algorithm to define optimum block size that will improve the deduplication ratio.

- Introduce more efficient storage to store Hash values of the data block, which use to check for duplicates.

- Improve the system that can connect multiple cloud storage services or multiple cloud storage accounts from the backend. Data blocks are sent to cloud storage using a random logic.

- Improve the system adding new functionalities such as file sharing, File deleting, file versioning and etc.

REFERENCES

[1] P. Anderson and L. Zhang, "Fast and Secure Laptop Backups with Encrypted De-duplication," in *LISA*, 2010.

[2] F. Gens. (2009, 29th Mar 2016). New IDC IT Cloud Services Survey: Top Benefits and Challenges. Available: http://blogs.idc.com/ie/?p=730

[3] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "ClouDedup: secure deduplication with encrypted data for cloud storage," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, 2013, pp. 363-370.

[4] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication," presented at the Proceedings of the 4th ACM international workshop on Storage security and survivability, Alexandria, Virginia, USA, 2008.

[5] S. Patidar, D. Rane, and P. Jain, "A Survey Paper on Cloud Computing," in *2012 Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, Rohtak, Haryana, 2012, pp. 394- 398

[6] P. M. Mell and T. Grance, "SP 800-145. The NIST Definition of Cloud Computing," National Institute of Standards & Technology2011.

[7] S.-F. Yang, W.-Y. Chen, and Y.-T. Wang, "ICAS: An inter-VM IDS Log Cloud Analysis System," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Beijing, 2011, pp. 285- 289.

[8] S. Zhang, S. Zhang, X. Chen, and X. Huo, "Cloud Computing Research and Development Trend," in *Second International Conference on Future Networks, 2010. ICFN '10.*, Sanya, Hainan, 2010, pp. 93- 97.

[9] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology (NIST) NIST Special Publication 800-145, September 2011 2011.

[10] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, "Cloud storage as the infrastructure of cloud computing," in *Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on*, 2010, pp. 380-383.

[11] V. Beal. (2018, 2018 Nov 18). *Cloud Storage*. Available: https://www.webopedia.com/TERM/C/cloud_storage.html

[12] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, *et al.*, "Windows Azure Storage: a highly available cloud storage service with strong consistency," presented at the Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, Cascais, Portugal, 2011.

[13] W. B. Chuan, S. Q. Ren, S. L. Keoh, and K. M. M. Aung, "Flexible Yet Secure De-duplication Service for Enterprise Data on Cloud Storage," in *Cloud Computing Research and Innovation (ICCCRI), 2015 International Conference on*, 2015, pp. 37-44.

[14] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," in *International Conference on Financial Cryptography and Data Security*, 2014, pp. 99-118.

[15] J. Li, X. Chen, M. Li, J. Li, P. P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE transactions on parallel and distributed systems*, vol. 25, pp. 1615-1625, 2014.

[16] S. Keelveedhi, M. Bellare, and T. Ristenpart, "DupLESS: server-aided encryption for deduplicated storage," in *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*, 2013, pp. 179-194.

[17] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage," *IEEE Security & Privacy*, pp. 40-47, 2010.

[18] F. Rashid, A. Miri, and I. Woungang, "A secure data deduplication framework for cloud environments," in *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*, 2012, pp. 81-87.

[19] A. Rahumed, H. C. Chen, Y. Tang, P. P. Lee, and J. C. Lui, "A secure cloud backup system with assured deletion and version control," in *Parallel Processing Workshops (ICPPW), 2011 40th International Conference on*, 2011, pp. 160-167.

[20] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *International Conference on Financial Cryptography and Data Security*, 2010, pp. 136-149.

[21] J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, 2002, pp. 617-624.

[22] T. Team. (2018, 2019 Jan 10). *Dropbox Is Doing Well, But Looks Rich In The Face Of Industry Headwinds*. Available: https://www.forbes.com/sites/greatspeculations/2018/05/21/dropbox-is-doing-well-but-looks-rich-in-the-face-of-industry-headwinds/#49a1fff736ed

[23] M. Moon. (2014, 2019 Jan 10). *Dropbox passwords posted online and millions more might follow*. Available: https://www.engadget.com/2014/10/14/dropbox-log-in-posted-online/?ncid=rss

[24] javamex. (2017, 2019 Jan 21). *How Deflater works*. Available: https://www.javamex.com/tutorials/compression/